

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO

ALEF IURY SIQUEIRA FERREIRA

Harmony Through Alignment: Unifying Multimodal Representations of Sound and Text Across Languages

Goiânia
2026



UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a [Lei 9.610/98](#), o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

☒ Dissertação ☐ Tese ☐ Outro*: _____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

Exemplos: Estudo de caso ou Revisão sistemática ou outros formatos.

2. Nome completo do autor

Alef Iury Siqueira Ferreira

3. Título do trabalho

Harmony Through Alignment: Unifying Multimodal Representations of Sound and Text Across Languages

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento ☒ SIM ☐ NÃO¹

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:

- a) consulta ao(a) autor(a) e ao(a) orientador(a);
 - b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
- O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.



Documento assinado eletronicamente por **Alef Iury Siqueira Ferreira, Discente**, em 01/09/2026, às 12:01, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Nádia Félix Felipe da Silva, Usuário Externo**, em 03/09/2026, às 16:07, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **6489181** e o código CRC **0B075ECE**.

ALEF IURY SIQUEIRA FERREIRA

Harmony Through Alignment: Unifying Multimodal Representations of Sound and Text Across Languages

Dissertação apresentada ao Programa de Pós-Graduação da Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Sistemas Inteligentes e Aplicações.

Orientadora: Profa. Nádia Félix Felipe da Silva

Co-Orientadora: Profa. Sandra Eliza Fontes de Avila

Goiânia
2026

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Ferreira, Alef Iury Siqueira

Harmony Through Alignment: Unifying Multimodal Representations of Sound and Text Across Languages [manuscrito] / Alef Iury Siqueira Ferreira. - 2026.

CCXXXIX, 239 f.:

Orientadora: Prof(a). Dra. Nadia Felix Felipe da Silva ; co-orientadora: Dra. Sandra Eliza Fontes de Ávila

Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2026.

Apêndice.

Bibliografia.

Inclui: siglas, algoritmos, lista de figuras, lista de tabelas.

1. Multimodalidade. 2. Alinhamento Emergente. 3. Representation Learning. 4. Aprendizado Auto-supervisionado. 5. Redes Neurais Profundas.
I. da Silva, Nadia Felix Felipe, orient. II. de Ávila, Sandra Eliza Fontes, co-orient.
III. Título.

CDU 5



UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº 14 da sessão de Defesa de Dissertação de **Alef Iury Siqueira Ferreira**, que confere o título de Mestre em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos trinta e um dias do mês de julho de dois mil e vinte e seis, a partir das catorze horas, via sistema de webconferência da RNP, realizou-se a sessão pública de Defesa de Dissertação intitulada “**Harmony Through Alignment: Unifying Multimodal Representations of Sound and Text Across Languages**”. Os trabalhos foram instalados pela Orientadora, Professora Doutora Nádia Félix Felipe da Silva (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professora Doutora Sandra Eliza Fontes de Avila (Unicamp), coorientadora; Professora Doutora Luciana de Oliveira Berretta (INF/UFG), membra titular interna; e Professor Doutor Arnaldo Candido Junior (UNESP), membro titular externo. A participação dos membros da banca ocorreu por meio de videoconferência. Durante a arguição os membros da banca não fizeram sugestão de alteração do título do trabalho. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pela Professora Doutora Nádia Félix Felipe da Silva, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos trinta e um dias do mês de julho de dois mil e vinte e seis.

TÍTULO SUGERIDO PELA BANCA



Documento assinado eletronicamente por **Nádia Félix Felipe da Silva, Usuário Externo**, em 31/07/2026, às 16:08, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Luciana De Oliveira Berretta, Professora do Magistério Superior**, em 31/07/2026, às 16:10, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Arnaldo Candido Junior, Usuário Externo**, em 31/07/2026, às 16:11, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Alef Iury Siqueira Ferreira, Discente**, em 31/07/2026, às 17:13, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Sandra Eliza Fontes de Avila, Usuário Externo**, em 03/08/2026, às 11:13, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site
https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0,
informando o código verificador **6219318** e o código CRC **84044333**.

Referência: Processo nº 23070.023147/2026-97

SEI nº 6219318

Dedico esta dissertação à minha família.

Agradecimentos

Agradeço, de maneira especial, à minha orientadora, Profa. Nádia Félix, por todo o suporte, pela confiança e pelas inúmeras oportunidades que me proporcionou ao longo desta trajetória. À minha coorientadora, Profa. Sandra Avila, agradeço pela orientação, pelos ensinamentos e pelas experiências que levarei comigo para sempre. Minha profunda gratidão a essas duas pessoas extraordinárias, que marcaram minha trajetória e contribuíram para transformar minha vida.

Agradeço também aos integrantes do H.IAAC Meta 5, grupo do qual tive o enorme prazer e orgulho de fazer parte. São pesquisadores e pessoas incríveis, cujas discussões, contribuições e trocas de conhecimento ajudaram a lançar luz sobre as questões abordadas neste trabalho.

Aos meus colegas, colaboradores e amigos, agradeço pelas conversas, pelo apoio e pelo compartilhamento de conhecimentos, que possibilitaram o surgimento de novas ideias e contribuíram diretamente para a realização deste trabalho.

Aos meus pais, Eliene e Listernandes, agradeço pelo amor, pelo incentivo e pelo apoio incondicional em todos os momentos. Sua luta incessante permitiu que eu chegasse até aqui, e cada um dos sacrifícios realizados ao longo de todos esses anos foi fundamental para esta conquista. Nada disso teria sido possível sem vocês.

À minha parceira, Ana Maria, agradeço pelo carinho, pela compreensão e pelo apoio ao longo de toda esta jornada. Obrigado por estar ao meu lado nos momentos felizes e também nos mais difíceis, oferecendo força, acolhimento e companheirismo.

Agradeço, ainda, ao Centro de Excelência em Inteligência Artificial (CEIA) e ao Centro de Competência EMBRAPAII em Tecnologias Imersivas (AKCIT) pela disponibilização de recursos que possibilitaram o desenvolvimento desta e tantas outras pesquisas. Agradeço também ao Recod.ai Lab pela infraestrutura, pelos recursos e pelo suporte oferecidos durante a realização deste trabalho.

Por fim, agradeço a todas as pessoas que, direta ou indiretamente, contribuíram para esta trajetória e para a concretização desta dissertação.

Both sides enter the conversation poorer than they will in its course become.

Zygmunt Bauman,
Bauman on Bauman – Pro Domo Sua.

Resumo

Ferreira, Alef Iury. **Harmony Through Alignment: Unifying Multimodal Representations of Sound and Text Across Languages**. Goiânia, 2026. 235p. Dissertação de Mestrado. Instituto de Informática, Universidade Federal de Goiás.

O alinhamento entre modalidades alcançou avanços significativos, especialmente no domínio texto-imagem. Entretanto, a criação de representações robustas e multilíngues de linguagem e áudio ainda apresenta desafios, como a escassez de conjuntos de dados texto-áudio linguisticamente diversos e o elevado custo computacional do treinamento em larga escala. Para enfrentar essas limitações, esta dissertação apresenta o CACARA (*Cross-Modal Alignment Leveraging a Text-Centric Approach for Cost-Effective Multimodal and Multilingual Learning*), um *framework* para integrar novas modalidades de forma eficiente. O CACARA emprega uma estratégia de alinhamento emergente, na qual o áudio é alinhado a um modelo multilíngue texto-imagem pré-treinado e congelado. Utilizando apenas dados em inglês, o modelo herda as capacidades multilíngues do codificador textual, permitindo recuperação *zero-shot* entre línguas e modalidades, com menor tempo de treinamento e consumo de energia. Como contribuição complementar, esta pesquisa apresenta o PALMA (*Pre-trained Audio-Language Multilingual Alignment via Mixture-of-Layers*), um *framework* para o desenvolvimento e a otimização de modelos bimodais de linguagem e áudio. O PALMA avalia codificadores de estado da arte e paradigmas modernos de treinamento multimodal. Sua principal inovação é o método *Mixture-of-Layers* (MoL), uma estratégia aprendível que combina características de múltiplas camadas do codificador para produzir representações de áudio mais ricas. As avaliações empíricas validam as abordagens propostas, que alcançam resultados competitivos. Nossos achados fornecem fortes evidências de que o alinhamento emergente é uma estratégia viável, escalável e de baixo custo para a expansão multimodal. Este trabalho contribui para o desenvolvimento de sistemas multimodais mais acessíveis, eficientes e linguisticamente inclusivos.

Palavras-chave

Pré-treinamento, Linguagem-Áudio, Multimodal, Multilíngue, Aprendizado Contrastivo, Aprendizado de Representações, Recuperação Texto-Áudio, Recuperação Áudio-Texto, Classificação Zero-Shot.

Abstract

Ferreira, Alef Iury. **Harmony Through Alignment: Unifying Multimodal Representations of Sound and Text Across Languages**. Goiânia, 2026. 235p. MSc. Dissertation. Instituto de Informática, Universidade Federal de Goiás.

The alignment of data modalities has achieved significant success, particularly in the text-image domain. However, extending these principles to create robust, multilingual representations of language and audio introduces distinct challenges that remain partially unaddressed in the literature. These include the scarcity of linguistically diverse text-audio datasets and the prohibitively high computing cost of training large-scale models. To address the high computational and data costs associated with multimodal expansion, this dissertation first introduces CACARA, a framework designed to integrate new modalities efficiently. CACARA employs an emergent alignment learning strategy. In our case, a new modality, audio, is aligned with a pre-trained and frozen multilingual image-text model. The model inherits the text encoder’s rich multilingual capabilities by training our framework using only English data. This approach enables zero-shot cross-lingual and cross-modal retrieval, reducing training time and energy consumption. As a complementary contribution, this research presents PALMA (Pre-trained Audio-Language Multilingual Alignment via Mixture-of-Layers), a framework for the systematic design and optimization of bimodal language-audio models. PALMA goes beyond conventional methods by comprehensively evaluating state-of-the-art (SOTA) encoders and advanced training paradigms. Its central innovation is the “Mixture-of-Layers” (MoL) method. This learnable aggregation strategy fuses features from multiple encoder layers to create a richer, more nuanced audio representation. Empirical evaluations validate the proposed approaches, which achieve competitive results. Our findings give strong evidence that emergent alignment is a viable, scalable, and low-cost multimodal expansion strategy. This work contributes to the development of more accessible, efficient, and linguistically inclusive multimodal systems.

Keywords

Language-Audio Pretraining, Multimodal, Multilingual, Contrastive Learning, Representation Learning, Text-to-Audio Retrieval, Audio-to-Text Retrieval, Zero-Shot Classification

Contents

List of Figures	12
List of Tables	14
List of Algorithms	18
List of Abbreviations	19
1 Introduction	22
1.1 Problem Description	23
1.2 Motivation and Challenges	23
1.3 Research Questions	25
1.4 Objectives	27
1.5 Contributions	28
1.6 How to Read This Dissertation	30
2 Concepts	33
2.1 Artificial Neural Networks	33
2.1.1 The Artificial Neuron	34
2.1.2 The Perceptron	35
2.1.3 Multilayer Perceptron	36
2.1.4 Activation Functions and Differentiability	37
2.1.5 Backpropagation and Gradient-Based Learning	39
2.2 Deep Learning	40
2.2.1 Convolutional Neural Networks	41
Convolutional Layer	42
Pooling Layer	43
2.2.2 Recurrent Neural Networks	44
2.2.3 Attention Mechanism	45
2.2.4 Transformers	47
Vision Transformers	50
2.3 Learning Machines	51
2.3.1 Supervised Learning	52
2.3.2 Unsupervised Learning	52
2.3.3 Self-Supervised Learning	53
Contrastive Learning	54
2.4 Representation Learning	56
2.5 What is a Modality?	58
2.5.1 Modality and Human Communication	58

2.5.2	Modality from a Machine's Perspective	59
2.5.3	Multimodality	60
2.6	Multimodal Representation Learning	61
2.6.1	Frameworks for Multimodal Integration	61
2.6.2	Modality Gap	63
2.6.3	Explicit and Implicit Alignment	65
2.7	Audio-Language Models	66
2.7.1	Architectural Foundations and Design Principles	67
2.7.2	Contrastive Language-Audio Pretraining	68
2.8	Acoustic Pre-Processing	69
3	Related Work	75
3.1	Contrastive Audio-Language Pretraining	75
3.2	Data Scaling and Automatic Annotation	76
3.3	Model Design and Multi-Objective Learning	77
3.4	Compositional Alignment	78
3.5	Multilingual Contrastive Audio-Language Models	79
3.6	Beyond Two Modalities	80
3.7	Positioning CACARA and PALMA in the Literature	81
4	Methodology	84
4.1	CACARA	85
4.1.1	Introduction	85
4.1.2	CACARA Model	86
4.1.3	Multimodal and Multilingual Emergent Alignment	88
4.1.4	Optimization Pipeline	89
4.2	PALMA	90
4.2.1	Introduction	90
4.2.2	Experimental Framework and Model Design	91
4.3	Evaluation	93
5	Results	96
5.1	Results at a Glance: A TL;DR Overview	96
5.2	CACARA	101
5.2.1	Experiments and Results	101
5.2.2	Audio Encoder Consolidation and Hyperparameter Tuning	102
5.2.3	Data Augmentation	105
5.2.4	Multimodal Evaluation	105
	Audio-Language Retrieval and Classification	107
	Emergent Audio-Image Alignment	108
5.2.5	Multilingual Evaluation	112
	Cross-Lingual Gap	114
5.2.6	Efficiency and Scaling	117
5.2.7	Expanded Resources	118
5.2.8	Qualitative Analysis	120
5.2.9	Final Remarks	122
5.3	PALMA	124
5.3.1	Experiments and Results	124

5.3.2	Hyperparameter Tuning	125
5.3.3	Text Encoder Selection	128
5.3.4	Locked Audio Pretraining and Sigmoid Loss	131
5.3.5	Mixture-of-Layers (MoL)	134
	Main Mixture-of-Layers Results	136
	Layer-Weight Importance Exploration	140
	Dual-Encoder MoL and Projection-Only Ablations	144
	Exploratory Data Augmentation Analysis	145
5.3.6	Training Data Combination	147
5.3.7	Batch Size Exploration	150
5.3.8	Multilingual Supervision	154
	Multilingual Training Strategies	154
	Effect on Multilingual Retrieval	154
	Effect on Zero-Shot Classification	156
	Comparison with Multilingual Baselines	157
	English Performance Gains from Multilingual Supervision	161
	Summary	164
5.3.9	Final Remarks	165
6	Conclusion	167
6.1	Answering Research Questions	169
6.1.1	RQ1: How can multilingual audio-text alignment be achieved under English-centric data constraints?	169
6.1.2	RQ2: Can a new modality be efficiently integrated into an existing multilingual multimodal representation space?	170
6.1.3	RQ3: How do encoder choice and training strategy affect the performance and efficiency of multilingual audio-language representation learning?	171
6.1.4	RQ4: What is the impact of cross-layer aggregation on multilingual audio-text representations?	173
6.1.5	Summary	174
6.2	Limitations	175
6.3	Future Work	176
6.4	Generative AI Use Disclosure	177
6.5	Published and Under-Review Papers	177
6.5.1	Related to the Dissertation	177
6.5.2	Collaborations	177
	Bibliography	181
A	Resources Description	203
A.1	Data Augmentation	203
A.2	Datasets	204
A.3	Audio Encoders	207
A.4	Text Encoders	209
B	Evaluation Protocol	212
B.1	Multilingual Zero-Shot Classification Templates	212
B.2	Retrieval Evaluation Protocol	212

C	Hyperparameters and Computing Resources	216
C.1	CACARA	216
C.2	PALMA	216
D	CACARA - Expanded Resources Result	219
E	PALMA - Multilingual Results	221
E.1	Retrieval	221
E.2	Classification	229
F	Mixture-of-Layers Implementation Details	233

List of Figures

1.1	Overview of the main contributions.	30
2.1	Artificial neuron representation.	34
2.2	Schematic representation of a fully connected multilayer perceptron.	37
2.3	Examples of activation functions and their first derivatives.	38
2.4	Example of a 2D convolution for a 2×2 kernel and stride of 1.	42
2.5	Example of a max pooling operation.	43
2.6	Transformers Encoder-Decoder structure.	49
2.7	Multi-head self-attention.	49
2.8	Vision Transformers architecture.	50
2.9	Schematic illustration of representation learning.	56
2.10	Representation of the concept of heterogeneity gap.	62
2.11	Illustration of a modality gap between two arbitrary modalities.	63
2.12	Common Audio-Language Models architectures.	68
2.13	An illustration of the CLAP architecture.	69
2.14	Timeline of representative contrastive Audio-Language Models.	70
2.15	Illustration of the sampling and quantization phases for converting an analog signal into a digital signal.	71
2.16	Examples of common audio representations.	74
4.1	Flow of adding a new modality (Audio) to an existent text-image bimodal model.	87
4.2	The CACARA model's training and optimization steps are divided into four main stages.	89
4.3	Overview of the PALMA optimization pipeline.	92
5.1	Performance comparison of CACARA with multimodal baselines.	108
5.2	t-SNE visualization of CACARA embeddings on VGG-SS.	111
5.3	Multilingual audio-text retrieval performance of CACARA on AudioCaps and ClothoV2.	113
5.4	Relationship between distance to the English text-embedding centroid and multilingual audio-to-text retrieval performance on AudioCaps.	116
5.5	Computational efficiency and environmental impact comparison between the proposed CACARA model and a fully tri-modal baseline.	117
5.6	Multilingual audio-text retrieval performance of CACARA on AudioCaps.	119
5.7	Multilingual audio-text retrieval performance of CACARA on ClothoV2.	119
5.8	Examples of successful and unsuccessful text-to-audio and audio-to-text retrieval.	121

5.9	Examples of successful audio-to-image retrieval and image-to-audio retrieval.	123
5.10	Representation of the flow of Mixture of Layers (MoL).	135
5.11	Layer-weight distributions learned by the Weighted-Sum MoL configuration for the CED audio encoder.	141
5.12	Layer-weight distributions learned by the Weighted-Sum MoL configuration for the BEATs audio encoder.	142
5.13	Layer-weight distribution learned by the Weighted-Sum MoL configuration for the mGTE text encoder.	143
5.14	Effect of batch size on retrieval performance on AudioCaps.	152
5.15	Effect of batch size on retrieval performance on ClothoV2.	152
5.16	Effect of batch size on zero-shot classification performance.	153
5.17	AudioCaps multilingual retrieval performance under different multilingual training strategies.	157

List of Tables

3.1	Comparison of CACARA with representative multimodal and multilingual models.	82
3.2	Comparison of PALMA with other contrastive-based ALMs.	83
5.1	Learning rate ablation results for the audio encoder consolidation phase of CACARA on Audio-to-Text and Text-to-Audio retrieval tasks, evaluated on AudioCaps and ClothoV2.	103
5.2	Learning rate ablation results for audio encoder consolidation phase of CACARA on the zero-shot audio classification task, evaluated on ESC-50 and UrbanSound8K.	104
5.3	Ablation results for the different augmentation strategies tested in the optimization phase, for the Audio to Text and Text to Audio retrieval tasks, evaluated on AudioCaps and ClothoV2.	106
5.4	Ablation results for the different augmentation strategies tested in the optimization phase, for the zero-shot audio classification task, evaluated on ESC-50 and UrbanSound8K.	106
5.5	Multimodal results, divided into two datasets: AudioCaps and ClothoV2, and two retrieval tasks: audio-to-text and text-to-audio.	109
5.6	Multimodal results, evaluated on ESC-50 and UrbanSounds8K datasets for the classification task.	110
5.7	Audio-image retrieval results on VGG-SS for multimodal models and CACARA.	110
5.8	Results considering the Audio-to-Text and Text-to-Audio retrieval tasks on two different CACARA versions across twelve languages.	112
5.9	Classification results on two different versions of CACARA across twelve languages.	114
5.10	Computational efficiency and environmental impact comparison between the proposed CACARA model, which tunes only the audio encoder, and a fully tri-modal baseline.	117
5.11	Recall values for the Audio-to-Text and Text-to-Audio retrieval tasks using the expanded resources model across AudioCaps and ClothoV2 for the twelve evaluated languages.	120
5.12	Classification task results on the ESC-50 and UrbanSound8K datasets across the twelve evaluated languages.	122
5.13	Retrieval task results for the first PALMA ablation stage, evaluated on AudioCaps and ClothoV2 across Audio-to-Text and Text-to-Audio retrieval.	126
5.14	Classification task results for the first PALMA ablation stage, evaluated on the ESC-50 and UrbanSound8K datasets.	127

5.15	Recall@k results for Audio-to-Text and Text-to-Audio retrieval on Audio-Caps and ClothoV2 across all audio and text encoder combinations.	129
5.16	Classification accuracy for different audio–text encoder pairings on ESC-50 and UrbanSound8K.	130
5.17	Recall@k results for Audio-to-Text and Text-to-Audio retrieval on Audio-Caps and ClothoV2 across PALMA configurations.	132
5.18	Classification accuracy on ESC-50 and UrbanSound8K across PALMA configurations.	132
5.19	Recall@k results for Audio-to-Text and Text-to-Audio retrieval on Audio-Caps and ClothoV2 for PALMA configurations using the CED audio encoder.	137
5.20	Recall@k results for Audio-to-Text and Text-to-Audio retrieval on Audio-Caps and ClothoV2 for PALMA configurations using the BEATs audio encoder.	138
5.21	Classification accuracy on ESC-50 and UrbanSound8K for PALMA configurations using the CED audio encoder.	139
5.22	Classification accuracy on ESC-50 and UrbanSound8K for PALMA configurations using the BEATs audio encoder.	140
5.23	Retrieval ablation for PALMA using the CED audio encoder and mGTE text encoder, comparing simultaneous MoL aggregation in both encoders, projection-only training, and projection-only training with dual-encoder MoL.	145
5.24	Classification ablation for PALMA using the CED audio encoder and mGTE text encoder, comparing simultaneous MoL aggregation in both encoders, projection-only training, and projection-only training with dual-encoder MoL.	145
5.25	Full retrieval analysis of data augmentation strategies for PALMA using the CED audio encoder, mGTE text encoder, LaT, and Audio MoL with Self-Att+Mean aggregation.	146
5.26	Classification analysis of data augmentation strategies for PALMA using the CED audio encoder, mGTE text encoder, LaT, and Audio MoL with Self-Att+Mean aggregation.	147
5.27	Recall@k results for Audio-to-Text and Text-to-Audio retrieval on Audio-Caps and ClothoV2 for CED + mGTE + MoL (Self-Att + Mean) under different training-data combinations.	148
5.28	Classification accuracy on ESC-50 and UrbanSound8K for CED + mGTE + MoL (Self-Att + Mean) under different training-data combinations.	149
5.29	Recall@k results for Audio-to-Text and Text-to-Audio retrieval on Audio-Caps and ClothoV2 across different batch sizes.	151
5.30	Zero-shot classification accuracy on ESC-50 and UrbanSound8K across different batch sizes.	151
5.31	Retrieval performance under different multilingual training strategies.	155
5.32	Zero-shot classification performance under different multilingual training strategies.	158
5.33	Retrieval performance comparison between PALMA multilingual training strategies and multilingual audio-language baselines.	158
5.34	Zero-shot classification comparison between PALMA multilingual training strategies and multilingual audio-language baselines.	159
5.35	Language-wise retrieval performance for the Random Sampling strategy.	160

5.36	Complete ML-CLAP retrieval results across evaluation languages on AudioCaps and ClothoV2. A2T and T2A correspond to Audio-to-Text and Text-to-Audio retrieval, respectively.	161
5.37	Complete GLAP retrieval results across evaluation languages on AudioCaps and ClothoV2.	162
5.38	Complete zero-shot classification performance for PALMA Random Sampling, ML-CLAP, and GLAP across evaluation languages on ESC-50 and UrbanSound8K.	163
5.39	English retrieval performance on AudioCaps and ClothoV2.	164
5.40	English zero-shot classification performance on ESC-50 and UrbanSound8K.	165
A.1	Summary of the audio-text datasets used for retrieval-oriented training and evaluation.	207
A.2	Summary of datasets used for validation and evaluation.	207
A.3	Comparison of the pretrained audio encoders evaluated in the CACARA optimization pipeline.	209
A.4	Comparison of the multilingual text encoders.	211
B.1	Language-specific prompt templates used for multilingual zero-shot classification.	213
C.1	Shared hyperparameters used across the main CACARA training phases.	217
C.2	Phase-specific batch sizes, training lengths, and hardware used in the CACARA optimization pipeline.	217
C.3	Shared hyperparameters used across the PALMA optimization pipeline.	218
C.4	Phase-specific batch sizes, training lengths, and hardware used in the PALMA optimization pipeline.	218
D.1	Retrieval results for the expanded resources experiments.	220
E.1	Language-wise retrieval performance for the English-only strategy.	221
E.2	Language-wise retrieval performance for the Random Sampling strategy.	222
E.3	Language-wise retrieval performance for the LE-Mixture 10% strategy.	223
E.4	Language-wise retrieval performance for the LE-Mixture 20% strategy.	224
E.5	Language-wise retrieval performance for the LE-Mixture 30% strategy.	225
E.6	Language-wise retrieval performance for the LE-Mixture 40% strategy.	226
E.7	Language-wise retrieval performance for the LE-Mixture 50% strategy.	227
E.8	Language-wise retrieval performance for the LE-Mixture 100% strategy.	228
E.9	Zero-shot classification performance for English-only across evaluation languages. Values are reported as mean $\pm$ standard deviation across three runs.	229
E.10	Zero-shot classification performance for Random Sampling across evaluation languages.	229
E.11	Zero-shot classification performance for LE-Mixture 10% across evaluation languages.	230
E.12	Zero-shot classification performance for LE-Mixture 20% across evaluation languages.	230

E.13	Zero-shot classification performance for LE-Mixture 30% across evaluation languages.	231
E.14	Zero-shot classification performance for LE-Mixture 40% across evaluation languages.	231
E.15	Zero-shot classification performance for LE-Mixture 50% across evaluation languages.	232
E.16	Zero-shot classification performance for LE-Mixture 100% across evaluation languages.	232

List of Algorithms

1	Grouped Audio–Text Retrieval Evaluation	215
2	MEAN Aggregation	233
3	WEIGHTED-SUM	234
4	MoL Gating: Self-Attention Refiner + Mean Pooling	234
5	MoL Gating: Self-Attention refiner + Attentive Pooling	235

List of Abbreviations

AA	Auto-ACD
AC	AudioCaps
ALIGN	A Large-scale Image and Noisy-text Embedding
ALM	Audio-Language Model
ANN	Artificial Neural Network
ASC	AudioSetCaps
AST	Audio Spectrogram Transformer
AudioMAE	Audio Masked Autoencoder
BEATs	Bidirectional Encoder Representations from Audio Transformers
BERT	Bidirectional Encoder Representations from Transformers
BM	Bimodal Model
BPTT	Backpropagation Through Time
C	ClothoV2
CACARA	Cross-Modal Alignment Leveraging a Text-Centric Approach for Cost-Effective Multimodal and Multilingual Learning
CED	Consistent Ensemble Distillation for Audio Tagging
CLAP	Contrastive Language-Audio Pretraining
CLIP	Contrastive Language-Image Pre-training
C-MCR	Connecting Multi-modal Contrastive Representations
CNN	Convolutional Neural Network
CPC	Contrastive Predictive Coding
dB	Decibel
DFT	Discrete Fourier Transform
ESC-50	Environmental Sound Classification 50
FFT	Fast Fourier Transform
FFN	Feedforward Network
GFLOPs	Giga Floating-Point Operations
GMACs	Giga Multiply-Accumulate Operations
GPU	Graphics Processing Unit
HTS-AT	Hierarchical Token-Semantic Audio Transformer

Hz Hertz
InfoNCE Information Noise-Contrastive Estimation
kWh Kilowatt-hour
LaBSE Language-agnostic BERT Sentence Embedding
LAION Large-scale Artificial Intelligence Open Network
LAION-CLAP Large-scale Artificial Intelligence Open Network Contrastive Language-Audio Pretraining
LALM Large Audio-Language Model
LaT Locked-audio Text Tuning
LiT Locked-image Text Tuning
LLM Large Language Model
LoRA Low-Rank Adaptation
LSTM Long Short-Term Memory
MAE Masked Autoencoder
MAE-AST Masked Autoencoding Audio Spectrogram Transformer
MHA Multi-Head Attention
ML Machine Learning
ML-CLAP Multilingual Contrastive Language-Audio Pretraining
MLMM Multilingual Multimodal Pre-training
MLP Multilayer Perceptron
MM Multimodal Model
mGTE Multilingual General Text Embeddings
MoL Mixture of Layers
MS-CLAP Microsoft Contrastive Language-Audio Pretraining
NMT Neural Machine Translation
PALMA Pre-trained Audio-Language Multilingual Alignment via Mixture-of-Layers
R@Avg Average Recall
R@k Recall at k
RNN Recurrent Neural Network
RoBERTa Robustly Optimized BERT Pretraining Approach
RQ Research Question
SigLIP Sigmoid Loss for Language-Image Pre-training
SOTA State-of-the-Art
SSL Self-Supervised Learning
STFT Short-Time Fourier Transform
TL;DR Too Long; Didn't Read
VALOR Vision-Audio-Language Omni-Representation
VAST Vision-Audio-Subtitle-Text

ViT Vision Transformer

WC WavCaps

XLM-RoBERTa Cross-lingual Language Model RoBERTa

Introduction

Human perception emerges from the interaction of multiple senses, which together shape our experience and understanding of the world. Through vision, hearing, smell, taste, and touch, humans gather complementary information about their surroundings [159, 177]. This integration enables the brain to combine information from multiple senses into a single representation [52, 125]. Perception provides raw sensory data, while cognition processes and interprets this input. Together, they form the foundation of human experience [52, 159, 177]. Sensory organs gather data from the environment, which cognition then uses to enable complex mental functions like reasoning, memory, and decision-making [13]. Rather than operating in a strictly one-way sequence, the relationship between sensory input and perception forms a continuous feedback loop. Perception influences cognition, which in turn shapes how we interpret future sensory experiences [159].

Replicating this complex perceptual and cognitive architecture is one of the central challenges in developing effective automated agents [12]. Ideally, such systems should be capable of interpreting and integrating information from multiple sources, just like humans do when navigating complex environments, identifying patterns, and responding adaptively [12, 18]. A system restricted to a single form of input can capture only a partial representation of the available information, as real-world phenomena can be expressed through multiple complementary ways [12, 30, 122]. By contrast, models that integrate signals across different forms of input, or different “views”, often referred to as “modalities”, offer a path toward richer semantic representations, more flexible retrieval mechanisms, and more adaptive interaction with the real-world [12, 122, 182, 206]. Within this multimodal landscape, the interaction between language and sound is particularly important, as it underlies communication, supports artistic expression, and enriches contextual understanding of the surrounding environment [182, 206].

However, this relationship is not only multimodal, in the sense that it involves the integration of multiple input sources or communication channels, but also multilingual. Sounds are experienced everywhere, yet their interpretation and description are mediated by language, which can vary across linguistic and cultural contexts [54]. As a result, the same acoustic event may be described differently across languages and communities, and

systems restricted to English cannot adequately serve the diversity of real-world users or linguistic communities [54]. For this reason, developing multimodal models that bridge audio and text across languages is not merely a technical challenge, but also a necessary step toward building systems that are more inclusive, more broadly applicable, and better suited to multilingual real-world settings [121, 206, 213].

1.1 Problem Description

The widespread adoption of digital media has produced a vast and diverse body of acoustic and textual content, ranging from podcasts, music, and environmental recordings to transcriptions, captions, and user-generated descriptions [8, 158]. Despite this growth, developing systems that can effectively align audio and text remains a significant challenge, particularly in multilingual settings [27, 206]. Most existing approaches are developed either for a single modality or for a single language, and large-scale audio-text modeling remains strongly centered on English corpora. As a result, current methods offer limited support for learning shared semantic representations that remain coherent across both modalities and different languages.

This dissertation investigates how multilingual multimodal representations of audio and text can be learned under data constraints, with particular emphasis on English-centric training settings. More specifically, it examines whether multilingual capabilities can emerge from combining English-only audio-text supervision with a pre-trained multilingual text encoder, whether translated text can serve as a viable strategy for extending audio-text data beyond English, and how encoder choice, fine-tuning strategies, and layer aggregation influence performance in cross-modal information retrieval. The central hypothesis of this dissertation is that effective multilingual audio-text alignment can be learned efficiently even under these constraints, provided that cross-lingual structure is transferred through the text encoder or supported by appropriate data and architectural design choices.

1.2 Motivation and Challenges

There are several compelling reasons to align audio and text representations across languages. Systems with this capability could substantially advance applications such as multilingual audio captioning [98], machine translation for audio content [123], assistive technologies for people with sensory impairments [97], and cross-modal and cross-lingual information retrieval [48, 199]. At the same time, achieving this goal remains difficult due to challenges that span data availability, representation learning, model design, and training efficiency.

One of the most significant limitations is the scarcity of audio-text paired data. While valuable human-annotated resources exist, such as AudioCaps [99] (~57K samples), Clotho [45] (~5K samples), and MusicCaps [1] (~5.5K samples), they are limited in scale due to the high cost of manual annotation [99, 204]. To overcome this bottleneck, researchers have developed much larger datasets like WavCaps [127] (400K), LAION-Audio-630K [201], Auto-ACD [183] (~1.5M), and AudioSetCaps [10] (~1.9M), using automated annotation pipelines. However, even these larger datasets are dwarfed by their counterparts in the vision-language domain, which can contain billions of pairs [167, 204], highlighting a significant disparity in scale that limits model development. Furthermore, these datasets are exclusively English-centric, presenting a critical obstacle to creating genuinely multilingual models. This lack of linguistic diversity is compounded by a lack of diversity in the audio sources themselves, which are often drawn from a few overlapping base datasets like AudioSet [62] and FreeSound [58].

A second challenge concerns representation learning itself. The quality of learned joint embeddings depends not only on the choice of unimodal encoders [49, 202], but also on geometric properties of the shared latent space. In particular, multimodal models often suffer from the so-called “modality gap” [53, 113], a phenomenon observed in models such as Contrastive Language-Image Pre-Training (CLIP) [154] and Contrastive Language-Audio Pretraining (CLAP) [49]. In this setting, embeddings from different modalities occupy distant regions of the representation space, even when they are intended to be semantically aligned. Importantly, this separation is not merely a consequence of differing data distributions, but is also related to the intrinsic behavior of deep neural networks and contrastive optimization [4, 46, 113, 166, 173, 211]. As a result, the modality gap can hinder performance on downstream tasks by weakening the coherence of the learned joint space. Furthermore, naively forcing embeddings closer together may distort the semantic structure of the representation space and does not necessarily lead to better alignment [53, 72].

A third challenge lies in training objectives and computational efficiency. Contrastive learning objectives, such as InfoNCE [140], are standard for learning aligned embeddings. However, their optimization within a multilingual, multimodal setting introduces unique difficulties. One of the main issues concerns the construction of positive and negative pairs. This process is fundamental to contrastive learning, since the strength of the learned representations depends on the quality and quantity of negative examples. When too few negative samples are used, the contrastive signal may be insufficient, often leading to suboptimal performance [91]. On the other hand, configurations with large quantities of negatives are more effective but introduce high computational costs. Additionally, the staggering computational cost of training large-scale multimodal models poses a challenge. The entire life-cycle of these models, from initial training to fine-

tuning, is resource-intensive and carries a substantial environmental footprint [165]. The memory required to fine-tune these models often fully surpasses the capacity of high-end single-GPU systems, rendering this crucial research inaccessible to those without access to extensive, large-scale infrastructure. This high barrier to entry poses a significant bottleneck to innovation and inclusivity in the field [165].

A fourth challenge concerns how current architectures make use of encoded representations. Most existing approaches rely primarily on the final encoder layer to construct joint embeddings. However, deep neural networks learn hierarchical representations across depth [51, 114, 144–146, 188]. In speech encoders, for instance, lower layers (those near the input) often capture local acoustic patterns, while higher layers (those near the output) encode broader semantic information [144]. Under limited-data conditions, relying only on the final layer may discard intermediate signals that could contribute to cross-modal alignment. For this reason, cross-layer aggregation, which combines information from multiple depths of the network, emerges as a promising yet still underexplored strategy for improving multimodal representations.

Taken together, these challenges motivate the central premise of this work: multilingual multimodal alignment for audio and text should not depend exclusively on prohibitively expensive training pipelines. Instead, it is necessary to investigate whether efficient alternatives can produce robust multilingual alignment. In this work, those alternatives are examined through three main axes: the use of translation as a strategy for multilingual data extension, the use of a pre-trained multilingual text encoder as a source of cross-lingual structure, and the use of architectural and training choices that may improve the quality of the shared representation space.

1.3 Research Questions

In light of the challenges described above, this work is guided by the following research questions:

RQ1. How can multilingual audio-text alignment be achieved under English-centric data constraints?

Most large-scale audio-text paired datasets are available only in English, limiting the training of models capable of operating across languages. Nevertheless, pretrained multilingual text encoders already organize representations from multiple languages within a shared semantic space. This creates the opportunity to introduce a new modality into that space by aligning its representations only with the multilingual textual anchor. In this work, we incorporate audio as the new modality and investigate whether the resulting emergent multilingual transfer is sufficient to support audio-text retrieval and

zero-shot classification in languages not observed during the explicit alignment training. We further examine whether this transfer remains consistent across languages, datasets, retrieval directions, and downstream tasks. Because the inherited multilingual structure may not benefit all languages equally, we also evaluate translated captions as a practical source of explicit multilingual supervision when large native multilingual audio-text datasets are unavailable. Therefore, this research question is addressed by examining both the multilingual capabilities inherited through emergent alignment with a frozen, pre-established semantic representation space and the additional improvements achieved through explicit multilingual supervision using translated captions.

RQ2. Can a new modality be efficiently integrated into an existing multilingual multimodal representation space?

The development of multimodal models commonly relies on the joint alignment, or subsequent realignment, of all participating modalities. Each modality is typically mapped into a shared semantic space through a specialized encoder tower, and conventional training strategies often update all encoder towers jointly. As additional modalities are introduced, this approach increases both computational and memory requirements. An alternative is to preserve an existing pretrained multimodal representation space and optimize only the encoder associated with the newly introduced modality. In this work, we investigate whether an audio encoder can be aligned with the textual component of a frozen multilingual image-text model, using the text encoder as the alignment anchor and without retraining the original image and text encoders. Because the image and text representations are already aligned, this procedure may also establish an indirect relationship between audio and images, even in the absence of paired audio-image supervision during training. We therefore evaluate whether text-anchored alignment enables emergent interactions among audio, text, and images while preserving the multilingual structure of the original representation space. We further analyze the predictive-performance, computational, and environmental trade-offs of this sequential alignment strategy relative to multimodal approaches that require more extensive and expensive joint training.

RQ3. How do encoder choice and training strategy affect the performance and efficiency of multilingual audio-language representation learning?

The performance of an audio-language model does not depend solely on the individual quality of its audio and text encoders. Encoders pretrained with different objectives may organize their representations differently, making some combinations more compatible with cross-modal alignment than others. Likewise, training strategies that are effective in other multimodal domains may behave differently when transferred to the audio-language setting. To address this question, we systematically compare

multiple pretrained audio encoders and multilingual text encoders, examining not only their individual performance but also the representational synergy that emerges from their combination in different downstream tasks. We further investigate the effects of full fine-tuning, locked-encoder training, and alternative contrastive learning objectives. Because predictive performance must be considered alongside computational cost, the analysis also encompasses optimization settings, training-data composition, batch size, and multilingual supervision. These experiments allow us to identify both the most effective and robust configurations and the performance-efficiency trade-offs involved in designing contrastive multilingual audio-language models.

RQ4. What is the impact of cross-layer aggregation on multilingual audio-text representations?

Pretrained encoders learn hierarchical representations in which different layers capture complementary forms of information. In audio encoders, lower and intermediate layers may emphasize acoustic and temporal structure, whereas deeper layers tend to encode increasingly abstract and semantic features. Nevertheless, most audio-language models construct their audio representations exclusively from the final encoder layer, potentially overlooking useful information retained at intermediate layers. We investigate whether aggregating representations from multiple audio-encoder layers can improve cross-modal alignment relative to relying solely on the final layer. To this end, we propose and evaluate the Mixture-of-Layers (MoL) mechanism alongside alternative aggregation strategies in audio-to-text retrieval, text-to-audio retrieval, and zero-shot audio classification. We further analyze the learned contribution of each layer to determine which regions of the encoder hierarchy remain informative for audio-language alignment, providing evidence that intermediate layers retain relevant complementary information. This analysis enables us to identify the benefits of cross-layer aggregation, the conditions under which these gains emerge, and the remaining limitations of the resulting representation space.

1.4 Objectives

The primary objective of this dissertation is to design and evaluate efficient approaches for learning multilingual multimodal representations of audio and text under limited multilingual supervision. To achieve this goal, the dissertation pursues the following specific objectives:

- O1:** To investigate whether multilingual audio-text alignment can be improved under English-centric data constraints through translated captions, multilingual supervision strategies, and pretrained multilingual text encoders.

- O2:** To evaluate whether audio can be efficiently integrated into an existing multilingual multimodal representation space without full multimodal retraining, by aligning a new audio encoder with a frozen pretrained image-text model.
- O3:** To systematically analyze how encoder choice, training strategy, data composition, and other hyperparameters can affect the performance and efficiency of multilingual audio-language models.
- O4:** To develop and assess cross-layer aggregation methods for exploiting intermediate representations from pretrained encoders, evaluating their impact on audio-text retrieval and zero-shot audio classification.

1.5 Contributions

This dissertation contributes to the study of multilingual multimodal representation learning, with particular emphasis on aligning sound and text across languages. Its methodological and empirical contributions address two central constraints in this area: the limited availability of native multilingual audio-text datasets and the computational cost of training large multimodal models. Figure 1.1 organizes these contributions into four complementary research axes: multilingual supervision, efficient multimodal expansion, audio-language model optimization, and representation-level analysis. Within these axes, the figure connects the data resources and methodological components developed in this work to the main empirical findings and their broader implications. The individual contributions are described below.

1. **A problem formulation for multilingual sound-text alignment under data and computational constraints.** This work frames multilingual audio-language learning as a problem that combines three interconnected challenges: the scarcity of multilingual audio-text pairs, the predominance of English-centric resources, and the high computational cost of large-scale multimodal training. This scenario motivates the investigation of efficient methods capable of building shared representation latent spaces that remain coherent across modalities and languages.
2. **Multilingual supervision and evaluation for audio-text representation learning.** This work constructs multilingual audio-text supervision and evaluation resources by translating the captions of the datasets considered in this dissertation into twelve languages. The contribution does not lie in caption translation as a general technique, which has been adopted in other domains, but in its application to multilingual audio-language representation learning, a setting in which comparable multilingual resources remain limited. These resources enable a controlled investigation of how explicit multilingual supervision affects cross-lingual audio-text alignment and multilingual generalization. The work also analyzes different

strategies for composing multilingual training, including Random Sampling and LE-Mixture. Random sampling is not introduced here as a new general-purpose sampling method. Instead, the original contribution lies in its systematic evaluation in multilingual audio-text training and in the empirical evidence that this simple strategy can achieve results competitive with more structured language-mixture approaches. This finding indicates that effective multilingual supervision does not necessarily require complex language-scheduling mechanisms.

3. **CACARA, an efficient framework for multimodal and multilingual expansion through emergent alignment.** This work proposes CACARA, a framework that adds a new modality to a pretrained multilingual representation space. Instead of retraining all modalities jointly, CACARA aligns the newly added modality encoder with a frozen textual anchor, allowing the new modality to inherit part of the multilingual and multimodal structure already present in the pretrained space. This contribution investigates whether audio-text, audio-image, and multilingual capabilities can emerge from efficient modality integration.
4. **PALMA, a systematic optimization pipeline for multilingual audio-language models.** This work introduces PALMA as a complementary bimodal framework focused on audio-text representation learning. PALMA explores how audio encoders, multilingual text encoders, training strategies, data composition, hyperparameter choices, and multilingual supervision affect zero-shot performance on downstream tasks. This contribution provides a controlled analysis of the design choices that influence the quality and efficiency of audio-language alignment.
5. **Mixture-of-Layers, a learnable cross-layer aggregation strategy for pretrained acoustic representations.** This work proposes the Mixture-of-Layers mechanism to combine information from multiple encoder layers instead of relying only on the final representation. The analysis shows that pretrained acoustic encoders organize information hierarchically and that intermediate layers can remain informative for audio-language alignment. This contribution opens a path toward representation-aware alignment strategies that better exploit the internal structure of pretrained encoders.
6. **An analysis of performance-efficiency trade-offs in multilingual multimodal learning.** Beyond predictive performance, this work analyzes the computational implications of the proposed strategies. By comparing sequential alignment, encoder choices, training configurations, layer aggregation, and multilingual supervision strategies, this research provides empirical evidence that competitive multilingual multimodal systems can be built through efficient and modular design.

Together, these contributions support the hypothesis that multilingual alignment between sound and text can be obtained through efficient reuse of pre-trained models,

careful encoder selection, and representation-level strategies that avoid the need for full large-scale multimodal retraining.

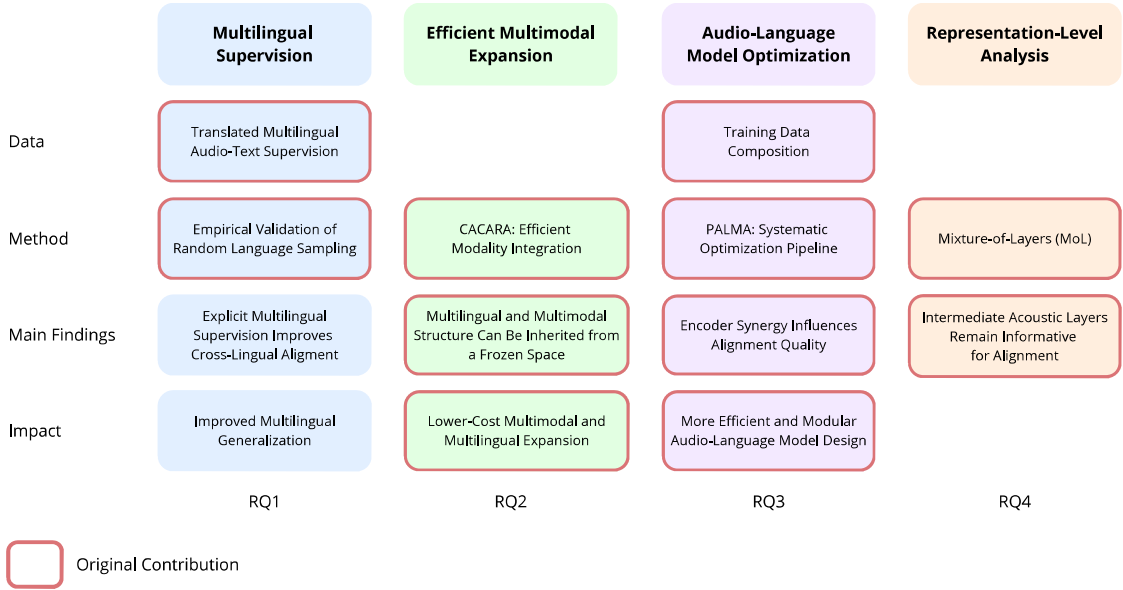


Figure 1.1: Overview of the contributions of this dissertation across four research axes: multilingual supervision, efficient multimodal expansion, audio-language model optimization, and representation-level analysis. The diagram connects data and methodological contributions to their main empirical findings and broader impacts. Red-outlined boxes identify original contributions introduced in this dissertation.

1.6 How to Read This Dissertation

This dissertation is organized around two complementary directions for learning multimodal and multilingual representations. The first direction, represented by CACARA, investigates how to add a new modality to an existing multilingual multimodal representation space via efficient, emergent alignment. The second direction, represented by PALMA, investigates how bimodal audio-language models can be systematically designed and optimized through encoder selection, training strategies, and cross-layer aggregation.

Chapter 1 introduces the motivation, problem definition, research questions, objectives, and contributions of this dissertation. This chapter establishes the general motivation behind multilingual audio-text alignment under English-centric data constraints and the efficient integration of a new modality into an existing multilingual multimodal representation space.

Chapter 2 presents the theoretical and technical foundations needed to understand the remainder of the dissertation. It covers neural networks, deep learning, representation learning, modality, multimodality, multimodal representation learning, audio-language models, contrastive learning, and acoustic pre-processing.

Chapter 3 reviews the literature most closely related to this dissertation. It discusses contrastive audio-language pretraining, dataset scaling, synthetic audio captioning, encoder design, efficient training, fine-grained alignment, multilingual audio-language models, and the transition from bimodal alignment to broader multimodal expansion. This chapter provides the background for the limitations of English-centric audio-text datasets and multilingual audio-language models. It also supports the discussion on the transition from bimodal audio-language alignment to broader multimodal systems, as well as encoder design, training strategies, and representation aggregation.

Chapter 4 describes the proposed methods. Section 4.1 presents CACARA and explains how audio is aligned with a frozen multilingual image-text model. This section provides the methodological approach for integrating a new modality into an existing multilingual multimodal space without retraining all encoders jointly. It also addresses part of the question of multilingual audio-text alignment under English-centric data constraints, as CACARA investigates whether multilingual audio-text capabilities can emerge from English-only audio-text supervision using a pre-trained multilingual text encoder. Section 4.2 presents PALMA and describes the design choices explored for bimodal audio-language learning, including encoder combinations, training strategies, and the Mixture-of-Layers mechanism. This section provides the methodological basis for analyzing the impact of encoder choice and training strategy, as well as the role of cross-layer aggregation in audio-text representation learning.

Chapter 5 presents the empirical evaluation of the proposed approaches. The CACARA results address multilingual audio-text retrieval and zero-shot classification across languages, datasets, and tasks. They also evaluate whether audio, text, and image can be connected in a shared representation space after sequential alignment, while analyzing the efficiency, scaling behavior, and qualitative retrieval examples of the proposed approach. The PALMA results compare different audio encoders, text encoders, and training strategies in audio-text retrieval and zero-shot classification. They also evaluate the impact of the Mixture-of-Layers mechanism and analyze whether combining information from multiple encoder layers improves cross-modal alignment.

Finally, Chapter 6 summarizes the main findings of the dissertation, revisits each research question, discusses the limitations of the proposed approaches, and outlines future research directions for multilingual and multimodal audio-language representation learning. In particular, it synthesizes evidence from CACARA on multilingual transfer and efficient multimodal expansion, from PALMA on bimodal audio-language model

design and cross-layer aggregation, and on the broader implications of efficient alignment strategies for building more accessible multilingual multimodal systems.

Concepts

2.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational models inspired by the biology of the brain, comprised of interconnected processing units, commonly referred to as artificial neurons, whose outputs are determined by a weighted combination of inputs followed by an activation mechanism [15, 119, 163]. More broadly, ANNs can be interpreted as learning systems in which knowledge is represented in a distributed manner through adaptive parameters rather than through explicitly hand-crafted symbolic rules [15, 119, 163]. Historically, ANNs emerged from the connectionist view of intelligence, which assumes that intelligent behavior arises from the interaction of many simple units whose connections are adjusted through learning [119]. At the same time, modern neural networks are described as flexible statistical models for pattern recognition and nonlinear function approximation rather than as faithful biological replicas of the brain [71].

The origins of this line of research, often described as “neural computing” [119], are usually traced back to the work of McCulloch and Pitts [119, 124], which introduced a formal neuron model in which the output of a unit depends on whether a weighted combination of inputs exceeds a threshold. Their proposal showed that networks using such a neuron model could theoretically compute several basic logical functions, establishing a stronger link between neuroscience and computer science [119, 124, 163]. A second important step came from Hebb’s theory of learning [82], which proposed that repeated co-activation strengthens the connection between neurons. This idea shifted the focus from a fixed network structure to the adaptive modification of connections, helping establish the principle that learning can be understood as the adjustment of synaptic weights [82, 119].

Rosenblatt’s perceptron [160, 161] combined the artificial neuron abstraction with a supervised learning rule, adjusting weights from labeled examples. The perceptron demonstrated that a machine could learn a decision boundary directly from data, generating optimism about neural approaches in the domain of pattern recognition [160, 161, 168]. However, a few years later, Minsky and Papert [129] showed that single-layer perceptrons

are fundamentally limited to linearly separable problems. As discussed by Luger [119], the exclusive-or problem became the canonical example of this limitation, as a single linear decision boundary cannot solve it. This critique did not invalidate neural approaches altogether, but it showed that shallow threshold models were insufficient for addressing more complex, nonlinearly separable problems [15, 119, 168].

Addressing these limitations required moving beyond the original perceptron formulation. Continuous activation functions, multilayer architectures, and gradient-based learning methods provided the basis for more expressive and adaptive neural models [119]. Together, these developments established the foundations of modern neural networks and deep learning [15, 71, 168].

2.1.1 The Artificial Neuron

The artificial neuron is the basic computational unit of an ANN. In its general form, it receives an input vector $\mathbf{x} = [x_1, x_2, \dots, x_m]^\top$, combines these inputs linearly through a set of learnable parameters $\mathbf{w} = [w_1, w_2, \dots, w_m]^\top$ and a bias term b , and then applies an activation function $\phi(\cdot)$ to produce the output $\mathbf{y}$ [15, 78, 163]. In general, an artificial neuron can be described as comprising three main components: the synapses, each associated with a weight, a linear combiner, which forms a weighted sum of the input signals considering the weights of the synapses, and an activation function that limits the amplitude range of the output signal, as shown in Figure 2.1.

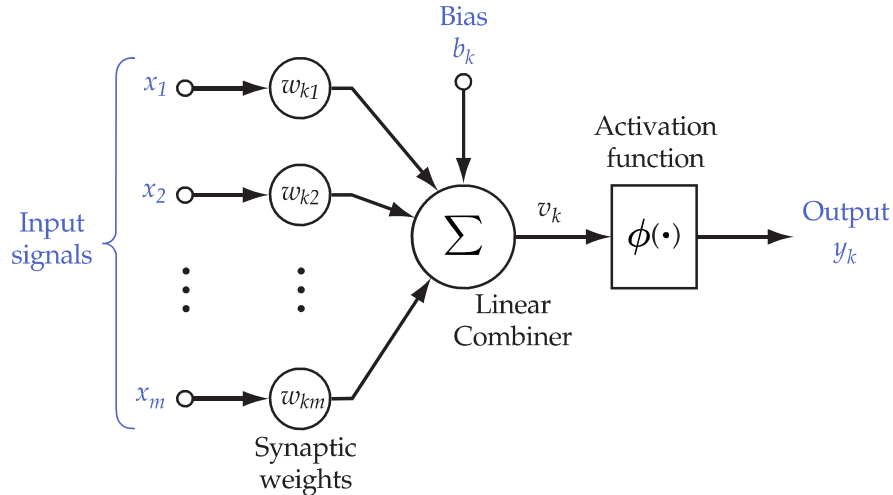


Figure 2.1: Artificial neuron representation. Adapted from Haykin et al. [78].

Formally, for a neuron k , this operation can be written as:

$$v_k = \sum_{i=1}^m w_{ki}x_i + b_k, \quad (2-1)$$

$$y_k = \phi(v_k), \quad (2-2)$$

where v_k denotes the neuron's activation potential (or pre-activation), and y_k is its output [78].

In vector form, the same operation can be written more compactly as:

$$y = \phi(\mathbf{w}^\top \mathbf{x} + b). \quad (2-3)$$

The affine transformation $\mathbf{w}^\top \mathbf{x} + b$ defines how the input dimensions are combined, with each weight controlling the magnitude and direction of the contribution of a corresponding input feature. The activation function determines how this combined signal is transformed before being propagated to subsequent units. Finally, the bias shifts the response of the neuron independently of the input, allowing the model to represent decision boundaries that do not necessarily pass through the origin [15, 78].

2.1.2 The Perceptron

The perceptron, introduced by Rosenblatt, is a single-layer feedforward ANN, that is, a network in which information propagates only from the input toward the output, without feedback connections that return the output of the network to earlier layers. It was proposed as a trainable model for binary classification. In terms of signal propagation, it remains close to the McCulloch-Pitts neuron: the input signals are combined linearly through weighted connections, and the output is passed through a hard-limiting threshold function to produce a binary response [119, 163].

In its classical bipolar form, the perceptron receives an input vector $\mathbf{x} \in \mathbb{R}^m$, computes a weighted sum of the inputs together with a bias term, and predicts one of two classes according to whether the resulting activation exceeds a threshold t :

$$\hat{y} = \begin{cases} 1, & \text{if } \mathbf{w}^\top \mathbf{x} + b \geq t, \\ -1, & \text{otherwise.} \end{cases} \quad (2-4)$$

where $\mathbf{w}$ is the weight vector, b is the bias, and t is the decision threshold.

The perceptron learning procedure is a simple form of supervised learning. After the model produces a prediction for a training example, a teacher provides the desired output, and the weights are modified only when the current prediction is incorrect. For instance, considering a training sample $(\mathbf{x}_k, y_k)$, from a training set $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where $\mathbf{x}_k$ denotes the input vector and $y_k \in \{-1, +1\}$ denotes the desired label, the update of the i -th weight can be written as:

$$\Delta w_i = \eta(y_k - \hat{y}_k)x_{ki}, \quad (2-5)$$

where Δw_i denotes the change applied to the i -th weight, $\eta > 0$ is the learning rate, which scales the correction applied to the weights, and $\hat{y}_k$ is the current perceptron output [119]. Since both y_k and $\hat{y}_k$ take values in $\{-1, +1\}$, the difference $y_k - \hat{y}_k$ can only be 0, 2, or -2 . No update is made when the example is correctly classified. Otherwise, the weight vector is shifted in a direction that reduces the classification error associated with the current example. In vector form, an equivalent update can be written as:

$$\mathbf{w} \leftarrow \mathbf{w} + \eta(y_k - \hat{y}_k)\mathbf{x}_k, \quad (2-6)$$

$$b \leftarrow b + \eta(y_k - \hat{y}_k). \quad (2-7)$$

If there exists a set of parameters that correctly classifies every example in the training set, the perceptron learning procedure will converge to such a solution. However, when the classes are not linearly separable, no single-layer perceptron can represent the desired decision boundary [119, 163].

2.1.3 Multilayer Perceptron

A multilayer perceptron (MLP) extends the single-layer perceptron by introducing one or more hidden layers between the input layer and the output layer. This extension directly addresses the linearity limitation: while a single perceptron can only represent a linear decision boundary, an MLP with nonlinear hidden units can transform the input representation before classification, allowing the network to represent nonlinear decision boundaries and solve problems that are not linearly separable [15, 78, 119]. In a feed-forward architecture, the connections form a directed acyclic graph: each unit receives signals from units in earlier layers and sends its output to units in later layers [78, 163]. The neurons in the hidden layers are not directly observed in either the input or the output of the network; their role is to intervene between the external input and the final response of the model [78]. Figure 2.2 shows a visual representation of a generic multilayer perceptron.

The formulation of the MLP resembles that of a perceptron, as shown in Equation 2-3. Let $\mathbf{h}^{(0)} = \mathbf{x}$ denote the input vector. For a network with L layers of adaptive weights, the computation at layer ℓ can be written as:

$$\mathbf{h}^{(\ell)} = \phi^{(\ell)}\left(\mathbf{W}^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}\right), \quad \ell = 1, \dots, L, \quad (2-8)$$

where $\mathbf{W}^{(\ell)}$ and $\mathbf{b}^{(\ell)}$ are the weight matrix and bias vector of layer ℓ , and $\phi^{(\ell)}(\cdot)$ is the corresponding activation function. The network output is given by the representation of the final layer: $\mathbf{y} = \mathbf{h}^{(L)}$.

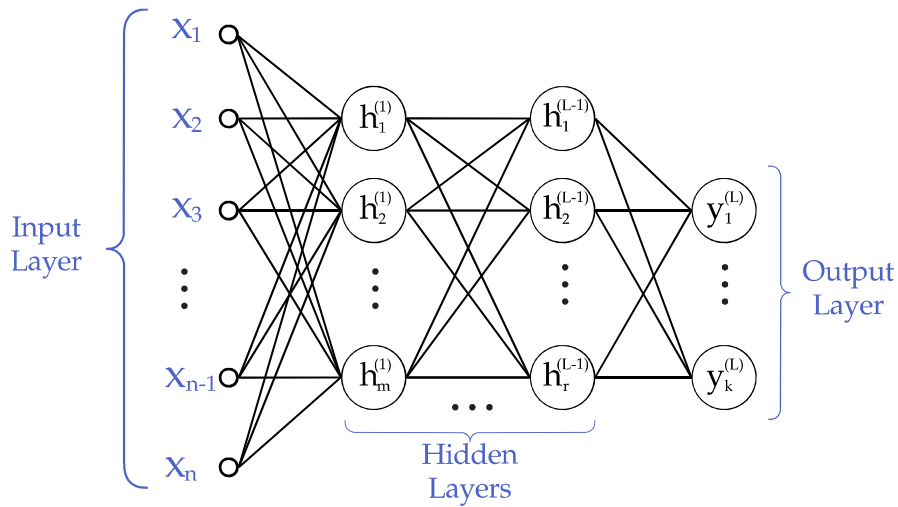


Figure 2.2: Schematic representation of a fully connected multi-layer perceptron.

This model is called a multilayer perceptron for historical reasons, since each stage of processing resembles the perceptron. The essential difference is that the hidden units use continuous nonlinear activation functions, rather than the hard threshold used in the classical perceptron [15]. This makes the network output differentiable with respect to the parameters, which is required by gradient-based training procedures [15].

If all hidden-layer activation functions are linear, the composition of the layers remains a linear transformation. In that case, the network is equivalent to a model without hidden layers, and depth does not increase representational power [15, 78]. From a theoretical standpoint, this expressive power is supported by the universal approximation theorem [34, 87], according to which a two-layer feedforward network, that is, a network with a single hidden layer and a sufficient number of hidden neurons, can approximate any continuous function on a compact domain to arbitrary accuracy [78, 152]. Although this theorem does not address the difficulty of finding appropriate parameter values during training, it provides a formal justification for the use of multilayer perceptrons as general nonlinear approximators [78, 152].

2.1.4 Activation Functions and Differentiability

The activation function is the component of the artificial neuron that is responsible for transforming and limiting the amplitude range of the activation potential value into the output. In multilayer neural networks, the activation function defines how the response of each hidden unit varies with its input and, consequently, how intermediate representations are formed throughout the network [15, 78]. Examples of activation functions are shown in Figure 2.3.

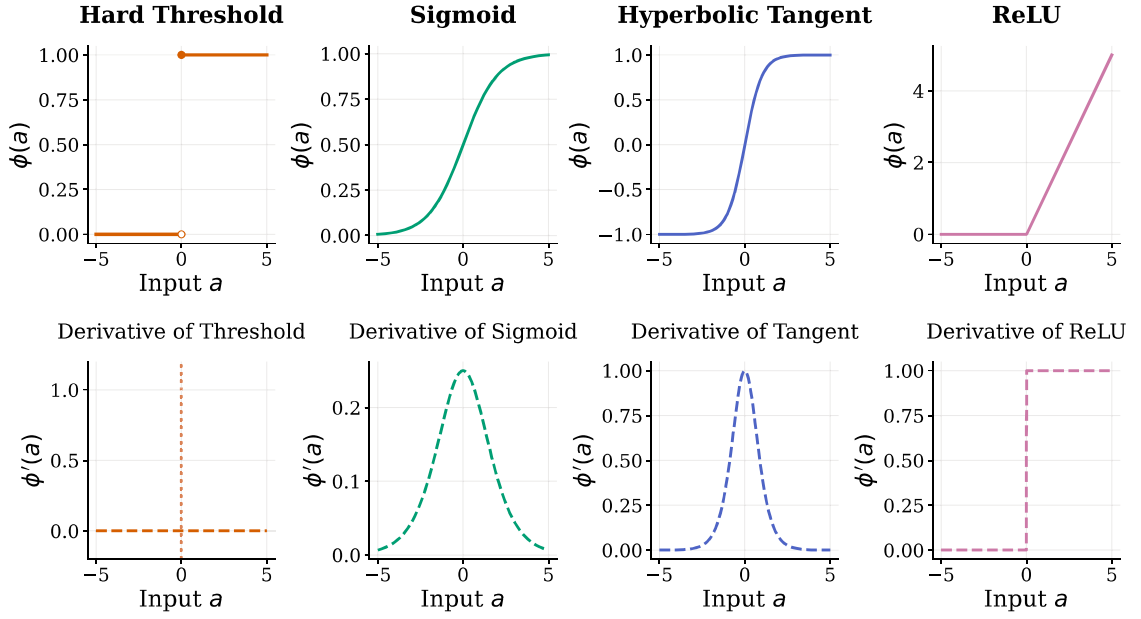


Figure 2.3: Examples of activation functions and their first derivatives.

In the classical perceptron, the activation is a hard threshold, which produces an abrupt binary decision. Such a function is suitable for defining a linear classifier, but it is problematic in multilayer settings trained by gradient-based methods, since it is non-differentiable at the threshold and constant almost everywhere else. In contrast, the multilayer perceptron was historically formulated with continuous nonlinear activation functions, precisely because the computation of local gradients in backpropagation depends on the derivative of the nonlinearity associated with each neuron [15, 78].

Two classic examples are the logistic sigmoid and the hyperbolic tangent:

$$\sigma(a) = \frac{1}{1 + e^{-a}}, \quad (2-9)$$

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}. \quad (2-10)$$

These functions are smooth and nonlinear, which makes them suitable for the classical derivation of backpropagation [15, 78]. Their derivatives can be written as:

$$\sigma'(a) = \sigma(a)(1 - \sigma(a)), \quad (2-11)$$

$$\frac{d}{da} \tanh(a) = 1 - \tanh^2(a). \quad (2-12)$$

The logistic sigmoid maps real-valued inputs to the interval $(0, 1)$, while $\tanh(\cdot)$ maps them to $(-1, 1)$. Because $\tanh(\cdot)$ is zero-centered, it often yields hidden activations with a more balanced sign structure than the sigmoid [15]. A well-known limitation

of sigmoid-like activation functions is saturation. For large positive or negative inputs, their derivatives become very small. In the logistic sigmoid function, the derivative reaches its maximum in the mid-range and tends to zero near the extremes of the output range, implying that weight updates become small when neurons operate in saturated regimes [71, 78]. This behavior can make optimization more difficult, particularly when gradient information must propagate through many layers, a phenomenon commonly referred to as the “vanishing gradient problem” [71, 136].

In modern neural networks, a widely adopted alternative is the rectified linear unit (ReLU), defined by:

$$\phi(a) = \max(0, a). \quad (2-13)$$

The ReLU returns zero for negative inputs and the identity for positive inputs. Although it is not differentiable at $a = 0$, it remains compatible with gradient-based optimization in practice because it is differentiable almost everywhere and admits a simple subgradient treatment at the origin. Moreover, ReLU-based networks have a particularly clear geometric interpretation: they implement continuous piecewise-linear functions, in which different activation patterns of the hidden units define different linear regions of the input space [152].

2.1.5 Backpropagation and Gradient-Based Learning

Training a multilayer perceptron consists of finding values for the parameters $\{\mathbf{W}^{(\ell)}, \mathbf{b}^{(\ell)}\}_{\ell=1}^L$ that minimize a loss function measuring the discrepancy between the predictions of the network and the desired outputs over a training set [15, 152]. If we denote by $\mathbf{w}$ the collection of all adaptive parameters of the network, and by $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ the training examples, one common choice of objective is the least squares loss [152]:

$$\mathcal{L}(\mathbf{w}) = \sum_{n=1}^N \mathcal{L}_n(\mathbf{w}), \quad \mathcal{L}_n(\mathbf{w}) = \frac{1}{2} \|\mathbf{y}_n - \theta(\mathbf{x}_n, \mathbf{w})\|^2, \quad (2-14)$$

where $\theta(\mathbf{x}_n, \mathbf{w})$ denotes the network output for input $\mathbf{x}_n$ [15]. The training problem is therefore a continuous, high-dimensional, and generally non-convex optimization problem, meaning that the loss surface may contain multiple local minima and different valleys, so that following only local gradient information does not guarantee convergence to a global minimum [152]. Because the total loss is written as a sum over training examples, its gradient also decomposes into a sum of per-example contributions, a property that is exploited by modern training methods [15, 152].

In multilayer networks, one of the main challenges is efficiently computing the derivatives of the loss with respect to all parameters. The role of backpropagation is to propagate the error information from the output layer to the hidden layers, allowing the

gradient of the loss to be obtained through repeated application of the chain rule [15, 119, 163]. In this sense, backpropagation provides the mechanism that makes the training of multilayer perceptrons practical.

Once these derivatives are available, they can be used by a gradient-based optimization rule. In its simplest form, gradient descent updates the parameters iteratively according to:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla \mathcal{L}(\mathbf{w}^{(\tau)}), \quad (2-15)$$

where $\eta > 0$ is the learning rate, τ denotes the iteration step, and $\nabla \mathcal{L}(\mathbf{w}^{(\tau)})$ denotes the gradient of the loss function with respect to the parameter vector at iteration τ . Thus, backpropagation and gradient descent are related but distinct: the former computes the gradients, whereas the latter uses them to update the parameters. In practice, modern neural networks are commonly trained with stochastic or mini-batch variants of gradient-based optimization, which estimate the gradient from one or a few training examples at a time [15, 152].

2.2 Deep Learning

Deep learning can be interpreted as the modern version of multilayer perceptron models, made practical by training much deeper models (with a large number of hidden layers). While multilayer perceptrons already introduced the idea of learning through compositions of linear transformations and nonlinear activations, deep learning emphasizes the use of many successive layers to learn intermediate representations of increasing abstraction directly from data, using the depth as a mechanism for representation learning [71, 186].

Intuitively, deeper models can represent complex input-output mappings through hierarchical compositions, in which higher-level features are built from lower-level ones. This property has proved especially important in domains such as computer vision, speech, and natural language processing, where the relevant structure of the data is often distributed across multiple levels of abstraction [71, 136, 186]. Although the basic principles remain those of shallow ANNs, the practical success of deep learning also depended on advances in large-scale datasets, computational hardware, training procedures, and architectural innovations [71, 136, 168]. The artificial neuron, the multilayer perceptron, differentiable activation functions, and backpropagation remain the conceptual basis of deep learning. Modern deep learning mainly differs by applying these ideas at a greater scale, with deeper and more diverse architectures [71, 152, 168, 186].

2.2.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) [107, 108] are a special kind of architecture designed for input signals with a grid-like topology, such as images and time series [71, 136, 152]. There are several reasons why this type of data motivated the development of specialized architectures, particularly because fully connected networks exhibit limitations when applied to spatially or temporally structured representations.

One limitation of fully connected networks concerns dimensionality. In an MLP, the basic operation at each layer is an affine transformation followed by a nonlinearity. For an image $\mathbf{x} \in \mathbb{R}^{W \times H \times C}$, where W denotes the width, H the height, and C the number of channels, even a single fully connected layer with D output units requires a weight matrix with $(W \times H \times C) \times D$ parameters. As an example, for a 224×224 RGB image, this corresponds to $224 \times 224 \times 3 = 150,528$ input dimensions. The number of weights becomes prohibitively large even for shallow networks, leading to practical difficulties in terms of memory usage, computational cost, and the amount of required training data [136, 152].

Fully connected networks also fail to exploit the structure of spatial or temporal data. In images, for example, neighboring pixels are more strongly related than distant ones, and local patterns like edges, corners, textures, and object parts are often important informative visual structures. Similarly, in one-dimensional signals like audio, nearby samples are more strongly correlated. However, a dense layer treats all input dimensions uniformly and has no built-in notion of locality or neighborhood relationships. As a result, it does not naturally reflect the fact that structured data, such as images or audio signals, are often organized through local spatial or temporal relationships [136, 152].

Additionally, meaningful patterns may appear at different positions in the input. For example, an object remains the same object even if it is slightly shifted in an image. In a fully connected model, however, the pattern associated with that object at one position is represented by a different subset of parameters than the same pattern at another position. Consequently, the network may need to learn similar detectors independently for different locations, requiring more training data than a model that can share parameters across positions [71, 152].

CNNs tackle these limitations by introducing local connectivity and parameter sharing. A convolutional layer processes local regions of the input using small, learnable filters, also called kernels, that are reused across spatial or temporal positions. This design reduces the number of trainable parameters, exploits local neighborhood structure, and allows the same detector to respond to similar patterns wherever they appear in the signal. Because the same kernel is applied across positions, convolution also induces translation equivariance, that is, when the input is shifted or translated, the corresponding feature responses captured by hidden neurons of the model shift accordingly [71, 152].

Convolutional Layer

A convolutional layer performs the main operation of a CNN. Instead of connecting every output unit to every input unit, a convolutional layer computes each output from a small local neighborhood of the input, usually called its “receptive field”, and a learnable kernel defines the operation. A kernel is a small set of coefficients that are applied to local regions of the input and reused across spatial positions. In the case of multi-channel inputs, the kernel contains coefficients not only for each spatial or temporal segment, but also for each input channel and each output channel [186].

Formally, for an input tensor $\mathbf{x}$ with C_{in} channels, the pre-activation at spatial position (i, j) and output channel c_2 can be written as:

$$s(i, j, c_2) = \sum_{c_1=1}^{C_{\text{in}}} \sum_{(k,l) \in \mathcal{N}} w(k, l, c_1, c_2) x(s_h i + k, s_w j + l, c_1) + b(c_2). \quad (2-16)$$

where $\mathcal{N}$ denotes the set of spatial offsets covered by the kernel, or in other words, the local window covered by the kernel, $w(k, l, c_1, c_2)$ are the kernel coefficients connecting input channel c_1 to output channel c_2 , and $b(c_2)$ is the bias of the output channel [184]. The terms s_h and s_w denote the vertical and horizontal strides, respectively. They determine how far the kernel is shifted in the input as it moves from one output position to the next. The corresponding activation is then obtained by applying a nonlinear function:

$$h(i, j, c_2) = \phi(s(i, j, c_2)). \quad (2-17)$$

An example of a 2D convolution can be seen in Figure 2.4, where a 2×2 kernel is used. Although this operation is commonly called convolution in the deep learning literature, most practical implementations correspond more precisely to cross-correlation, since the kernel is not flipped. In the context of learning, however, this distinction is usually not essential, because the filter coefficients are adapted from data and both formulations yield equivalent representational capacity [71, 152, 184, 186].

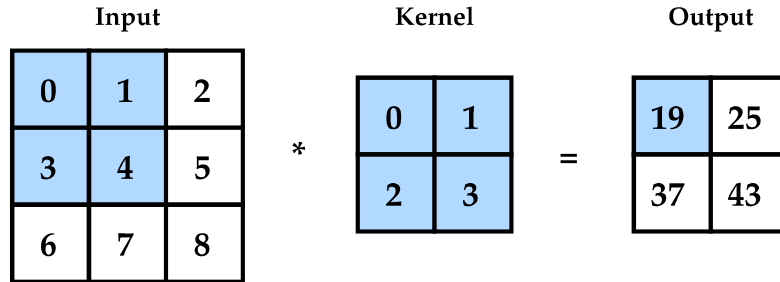


Figure 2.4: Example of a 2D convolution for a 2×2 kernel and stride of 1. Adapted from [136].

Pooling Layer

A pooling layer acts like a local compressor. It receives the spatial representations produced by a convolutional layer and reduces their resolution by replacing groups of neighboring values with a single representative value [136, 184, 186]. This operation is useful because after a convolutional layer has detected certain local patterns, it is often not necessary to preserve the exact position of every response. Instead, it may be sufficient to know that a strong response occurred within a small region [184, 186].

Let $\mathbf{h} \in \mathbb{R}^{H \times W \times C}$ denote the representation produced by a convolutional layer, where H and W are the spatial dimensions and C is the number of channels. A pooling operation is usually applied independently to each channel. For a local window of size $K_h \times K_w$ and strides s_h and s_w , average pooling computes the mean value inside each local region:

$$p_{\text{avg}}(i, j, c) = \frac{1}{K_h K_w} \sum_{u=0}^{K_h-1} \sum_{v=0}^{K_w-1} h(is_h + u, js_w + v, c). \quad (2-18)$$

In this case, the output value summarizes the general intensity of the responses inside the region. Another common choice is max pooling, which keeps only the largest value inside each local region:

$$p_{\text{max}}(i, j, c) = \max_{\substack{0 \leq u < K_h \\ 0 \leq v < K_w}} h(is_h + u, js_w + v, c). \quad (2-19)$$

Max pooling preserves the strongest evidence that a certain pattern was detected somewhere inside the pooling window, while discarding its precise location [136, 184]. For this reason, pooling can make the representation less sensitive to small translations or local variations in the input, yielding a partial form of local translation invariance [136, 152, 184, 186]. Figure 2.5 illustrates this process for max pooling.

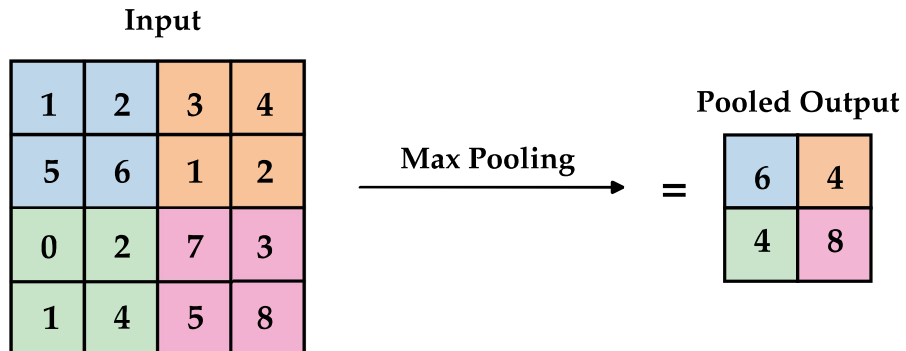


Figure 2.5: Example of a max pooling operation. Each output value stores the maximum activation observed inside the corresponding local region of the input representation. Adapted from [136].

2.2.2 Recurrent Neural Networks

Recurrent neural networks (RNNs) [162] are a family of neural architectures designed to model sequential data. Unlike feedforward networks, in which the connections form a directed acyclic graph, the computation performed by RNNs has cycles, that is, the outputs of one layer can feed back into the computation pipeline [71, 186]. While more specialized architectures like CNNs exploit local structures by applying the same filters across spatial or temporal positions, RNNs introduce the mechanism of “working memory”, keeping an internal state that is updated as the sequence is processed. This state allows information from previous elements to influence the processing of later ones [71, 186].

The parameter sharing across time is the main property of RNNs. In a similar way to how CNNs reuse the same filters across different positions, RNNs reuse the same transition function at every step of the sequence. This makes it possible to process sequences of different lengths and to generalize across positions in the sequence [71].

Formally, let $\mathbf{x}_1 = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T)$ denote an input sequence of length T . At each time step t , the network receives the current input vector $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$, and uses them to compute a new hidden state $\mathbf{h}_t$. A simple recurrent layer can be written as:

$$\begin{aligned}\mathbf{a}_t &= \mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h, \\ \mathbf{h}_t &= \phi(\mathbf{a}_t)\end{aligned}\tag{2-20}$$

where $\mathbf{a}_t$ is the pre-activation vector, $\mathbf{W}_{xh}$ is the input-to-hidden weight matrix, $\mathbf{W}_{hh}$ is the hidden-to-hidden recurrent weight matrix, $\mathbf{b}_h$ is a bias vector, and $\phi(\cdot)$ is a nonlinear activation function, commonly $\tanh(\cdot)$ [71, 136]. The initial hidden state $\mathbf{h}_0$ is usually defined as a vector of zeros or treated as a learnable parameter.

The hidden state can then be used to produce an output. When the model predicts one output at each time step, this operation can be written as:

$$\begin{aligned}\mathbf{o}_t &= \mathbf{W}_{hy}\mathbf{h}_t + \mathbf{b}_y, \\ \hat{\mathbf{y}}_t &= \psi(\mathbf{o}_t),\end{aligned}\tag{2-21}$$

where $\mathbf{o}_t$ denotes the output pre-activation, $\mathbf{W}_{hy}$ is the hidden-to-output weight matrix, $\mathbf{b}_y$ is the output bias, and $\psi(\cdot)$ is an output activation function [71, 136].

The recurrence can be expressed more generally as:

$$\mathbf{h}_t = f_{\theta}(\mathbf{h}_{t-1}, \mathbf{x}_t),\tag{2-22}$$

where f_{θ} is the shared transition function parameterized by θ . After repeated application, the hidden state $\mathbf{h}_t$ becomes a representation that depends on the sequence prefix $\mathbf{x}_1 =$

$(\mathbf{x}_1, \dots, \mathbf{x}_t)$. Therefore, the hidden state works like a summary of the information observed up until that point in time. Since this summary has fixed dimensionality, it is generally lossy, preserving mainly the aspects of the past sequence that are useful for the task being optimized [71].

Although the recurrent computation graph contains cycles, training is made possible by unrolling the network over a finite number of time steps. The unrolled representation is a feedforward computational graph in which each step has its own hidden state, but all steps share the same parameters [71, 186]. If the total loss is written as a sum of losses across time,

$$\mathcal{L} = \sum_{t=1}^T \mathcal{L}_t, \quad (2-23)$$

then the gradients with respect to the shared parameters are obtained by backpropagating through the unrolled graph. This procedure is called backpropagation through time (BPTT) [71, 184, 186]. Because the same parameters are used at each step, the final update accumulates gradient contributions from multiple positions in the sequence.

Despite their expressive power, simple RNNs are difficult to train on long sequences. During BPTT, gradients must propagate through many recurrent transitions. As in very deep feedforward networks, repeated multiplication by weight matrices and derivatives can cause gradients to become very small or very large, leading to the vanishing or exploding gradient problems [71, 186].

To alleviate these problems, gated recurrent architectures introduce additional mechanisms that control how information is stored, updated, and forgotten in the hidden state. Long short-term memory networks (LSTMs) [85] and gated recurrent units (GRUs) [25, 28] are two common examples. These models use gates to regulate the flow of information across time, making it easier to preserve relevant signals over longer sequences [71, 184]. Although more recent attention-based architectures have become dominant in many sequence modeling tasks, RNNs remain an important conceptual step in the development of deep learning models for sequential data.

2.2.3 Attention Mechanism

Attention is a mechanism that allows a model to focus on, or weight, the most relevant parts of an input representation when producing an output, processing global information more efficiently. For instance, when answering a question about a specific scene, it is expected that one person focus on the regions most relevant to the question at hand, rather than assuming that all regions are equally important. In neural networks, this intuition is implemented by allowing one representation to assign different weights to other representations, so that the output is built mainly from the elements considered most relevant to the current operation [186].

Historically, attention became more prominent in sequence-to-sequence models for Neural Machine Translation (NMT) [9, 120]. However, the idea had appeared earlier in neural sequence models in handwriting generation, where it was introduced as an attention mechanism constrained to move forward through the input sequence [73]. In encoder-decoder RNNs for NMT, the encoder reads the source sentence and compresses it into a fixed-length context vector, usually derived from the final hidden state. The decoder then uses this vector to generate the output sequence. This design creates a bottleneck, especially for long sequences, because all relevant information from the input must be stored in a single vector. Attention was introduced to reduce this limitation by allowing the decoder to access the encoder hidden states directly at each decoding step [9, 71, 120, 136].

Intuitively, attention can be thought of as a differentiable lookup operation. Given a query vector $\mathbf{q}$, a set of key vectors $\mathbf{k}_1, \dots, \mathbf{k}_m$, and a corresponding set of value vectors $\mathbf{v}_1, \dots, \mathbf{v}_m$, the model first measures how “aligned” the query is with each key. These alignments scores are then normalized into attention weights, and the output is computed as a weighted combination of the values. Formally, this can be written as:

$$\text{Attn}(\mathbf{q}, \{\mathbf{k}_i\}_{i=1}^m, \{\mathbf{v}_i\}_{i=1}^m) = \sum_{i=1}^m \alpha_i \mathbf{v}_i, \quad (2-24)$$

where each α_i is a non-negative scalar weight and $\sum_{i=1}^m \alpha_i = 1$. A common way to compute these weights is to apply a softmax function to an attention score $a(\mathbf{q}, \mathbf{k}_i)$:

$$\alpha_i = \frac{\exp(a(\mathbf{q}, \mathbf{k}_i))}{\sum_{j=1}^m \exp(a(\mathbf{q}, \mathbf{k}_j))}. \quad (2-25)$$

In this formulation, the query specifies what information is being requested, the keys determine which elements are relevant to the query, and the values contain the information that will be combined to produce the output [136, 186]. This type of operation is often described as a soft dictionary lookup, since the model does not retrieve a single entry from memory, but rather a weighted average of multiple entries [136].

In the original sequence-to-sequence setting, the query is usually derived from the decoder state, while the keys and values are derived from the encoder states. At decoding step t , the model computes a context vector $\mathbf{c}_t$ by attending to all encoder states:

$$\mathbf{c}_t = \sum_{i=1}^T \alpha_{ti} \mathbf{h}_i^e, \quad (2-26)$$

where $\mathbf{h}_i^e$ denotes the encoder hidden state at input position i , and α_{ti} indicates how much the decoder attends to that position when producing the next output token. In NMT, these weights work as a learned alignment between source and target words [9, 120, 136].

Attention mechanisms are commonly divided into soft and hard attention. In

soft attention, as mentioned before, the output is computed as a weighted average over all candidate values. This makes the operation differentiable, allowing the model to be trained with standard backpropagation [9, 203]. In hard attention, the model selects only one or a few input locations instead of averaging over all of them. This can yield a sparser, more selective mechanism, but the discrete selection process is not directly differentiable and usually requires stochastic estimation or reinforcement learning methods during training [136].

Self-attention is a variant of the attention mechanism and has become one of the main components of modern neural network architectures. In self-attention, the queries, keys, and values are all computed from the same set of input representations. As a result, each element of the input can gather information from all other elements, allowing the model to capture long-range dependencies without relying on recurrent processing. This is particularly useful in sequences, since information from distant positions can interact directly through the attention weights.

In matrix form, the scaled dot-product attention used in modern networks architectures is commonly written as:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right) \mathbf{V}, \quad (2-27)$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are matrices containing the queries, keys, and values, and d_k is the dimensionality of the key vectors [136, 192]. The product $\mathbf{Q}\mathbf{K}^\top$ computes the similarity between each query and each key, producing scores that decides how much to attend to each representation. The scaling factor $\sqrt{d_k}$ helps control the magnitude of these dot products before the softmax operation, avoiding excessively peaked attention distributions when the dimensionality is large [186].

This formulation provides the basis for the Transformers [192] architecture. Instead of using recurrence to propagate information step by step through a sequence, Transformers use layers with self-attention to allow input representations to exchange information directly. These layers are usually combined with position-wise feedforward networks, residual connections, normalization layers, and multi-head attention.

2.2.4 Transformers

The Transformers [192] architecture is based on self-attention and was originally introduced for sequence-to-sequence modeling in NMT, where the recurrent computation was replaced with attention-based operations. Instead of reading the input sequence step by step, as in RNNs, a Transformer processes the sequence in parallel as a set of input representations that can interact directly through self-attention layers [96, 136, 192].

This design addresses two limitations of recurrent models. First, in an RNN, information must be propagated through a chain of hidden states, which can make long-range dependencies difficult to preserve. Second, the recurrent computation is inherently sequential, since the hidden state at time step t depends on the hidden state at time step $t - 1$. Transformers reduce these limitations by allowing each input representation to attend directly to other representations in the sequence. As a result, the computation for different positions can be performed in parallel during training, making it more suitable for large-scale training [96, 136, 192].

The original Transformer architecture (Figure 2.6) follows an encoder-decoder structure. The encoder receives the input sequence and produces contextual representations of its elements. The decoder then generates the output sequence, using masked self-attention¹ over the previously generated outputs and an additional attention mechanism over the encoder representations [136, 192]. In many applications, however, only one part of this structure is used. Encoder-only models are common in representation learning [37, 105, 220], while decoder-only models are commonly used for autoregressive generation [155, 156, 209].

A transformer encoder is built by stacking several encoder blocks. Each block contains a multi-head self-attention (MHA) layer followed by a feedforward network (FFN) applied independently to each position. Residual connections [81] and normalization layers [6] are usually added around these components to improve optimization and stabilize training [136, 184, 192]. A simplified encoder block can be written as:

$$\begin{aligned}\mathbf{Z} &= \text{LayerNorm}(\mathbf{X} + \text{MHA}(\mathbf{X})), \\ \mathbf{H} &= \text{LayerNorm}(\mathbf{Z} + \text{FFN}(\mathbf{Z}))\end{aligned}\tag{2-28}$$

where $\mathbf{X}$ denotes the input representations of the block, $\text{MHA}(\cdot)$ denotes multi-head self-attention, $\text{FFN}(\cdot)$ denotes the position-wise feedforward network, and $\mathbf{H}$ is the output of the block. This formulation emphasizes that the transformer alternates between two operations: mixing information across positions through self-attention and transforming each position independently through a feedforward network.

Multi-head attention (Figure 2.7) extends the single self-attention operation by applying several attention mechanisms in parallel. Each head has its own learned projections for queries, keys, and values, allowing the model to represent different notions of similarity between the input elements [96, 152, 192]. The outputs of the heads are concatenated and projected back to the model dimension. This allows the same layer to capture different types of relationships, such as syntactic, semantic, or contextual relations

¹Masked self-attention applies a causal mask that allows each position to attend only to itself and preceding positions, preventing access to future tokens during autoregressive generation.

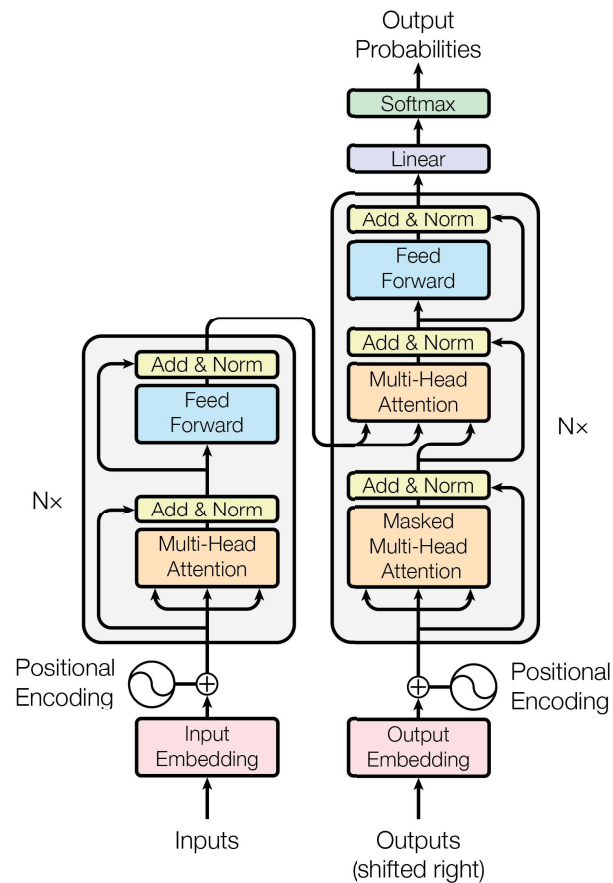


Figure 2.6: Transformers Encoder-Decoder structure. Extracted from Vaswani et al. [192].

in language.

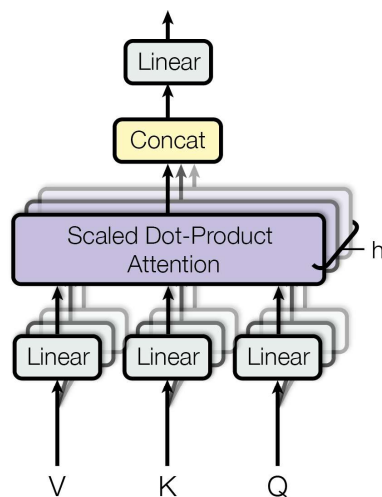


Figure 2.7: Multi-head self-attention. Extracted from Vaswani et al. [192].

Since self-attention compares representations by their content, it does not naturally encode the order of the input. If the same representations are permuted, the self-

attention operation is permuted in the same way. For this reason, transformers add positional information to the input representations. This can be done with fixed sinusoidal positional encodings or learned positional embeddings. These positional signals allow the model to distinguish between elements that have similar content but occur in different positions [136, 152, 192].

Vision Transformers

The Vision Transformers (ViTs) [43] (Figure 2.8) architecture is an adaptation of the Transformer encoder to image classification and is currently the basis for state-of-the-art (SOTA) audio classification models. In ViT, an image is converted into a sequence of patch representations, which are then processed by a standard transformer encoder. This differs from CNNs, which process images through local convolutional filters and gradually increase the receptive field with depth [43, 184, 186].

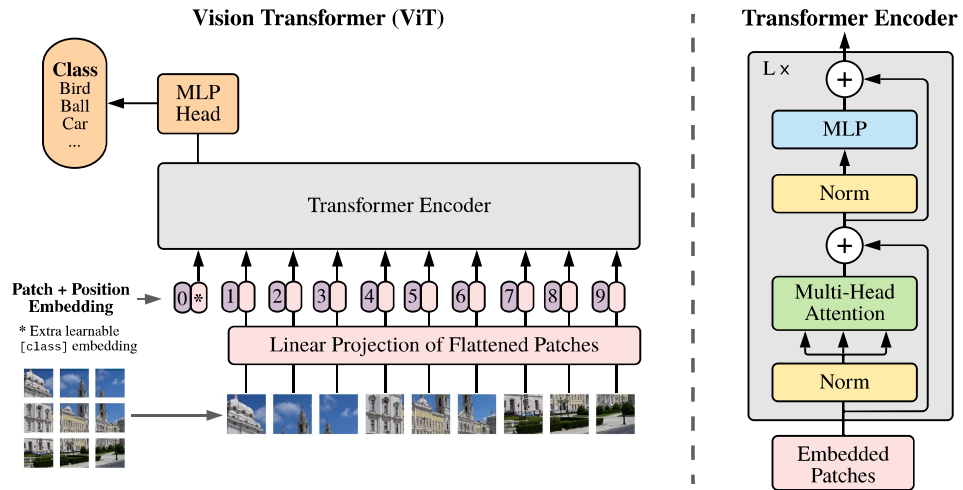


Figure 2.8: Extracted from Dosovitskiy et al. [43].

Formally, let an image be divided into non-overlapping patches of size $P \times P$. Each patch is flattened into a vector and mapped to a fixed-dimensional embedding through a learned linear projection. For an image of size $H \times W$, this produces $N = \frac{HW}{P^2}$ patch representations [43, 184]. Positional embeddings are then added to these representations, allowing the model to use information about the spatial arrangement of the patches.

In the original ViT formulation, an additional learnable classification representation is prepended to the sequence of patch representations. After the encoder processes the full sequence, the final state associated with this classification representation is used to predict the image class [43, 136].

The motivation for using Transformers in vision different from that in recurrent sequence modeling. CNNs already process images in parallel, so Transformers are not mainly introduced to avoid sequential computation. Instead, the motivation concerns the

locality bias of convolution. A convolutional layer assumes that nearby pixels are the most relevant, and information from distant regions can only interact after several layers, unless the architecture includes larger kernels, pooling, or non-local mechanisms. In contrast, self-attention allows each patch representation to interact with every other patch representation from the first layer [184, 186].

This global interaction is useful because image understanding often depends on relationships between distant regions. Self-attention provides a direct mechanism for representing these relationships. At the same time, this flexibility comes with a computational cost, since the attention matrix grows quadratically with the number of input elements [136, 184].

Compared with CNNs, ViTs rely on weaker built-in assumptions (inductive biases) about spatial processing, such as locality and translation invariance. This can have an advantage when very large datasets are available, since the model has more freedom to learn the relevant relationships from the data. However, in lower-data regimes, the inductive biases of CNNs can still be useful, since they encode assumptions that are well matched to natural images [43, 184].

2.3 Learning Machines

In cognitive science, “learning” can be broadly defined as the process in which an organism progressively forms internal models of the external world. Through experience, fragments of new data that are captured by the senses are transformed into structured representations that can be reused in new contexts [36]. Dehaene [36] presents seven complementary perspectives on learning, including the formation of internal world models, the adjustment of parameters, the reduction of error, the exploration of possible solutions, the optimization of rewards, and the restriction of the search space through prior assumptions. Although these definitions originate from cognitive science and neuroscience, they are closely related to the principles underlying modern machine learning.

In Machine learning (ML), a widely used definition by Mitchell [131] states that a computer program learns from experience E with respect to a class of tasks T and a performance measure P if its performance at tasks T , as measured by P , improves with experience E . This definition makes explicit the experience available to the system, the task that the system is expected to perform, and the criterion used to evaluate its performance [71, 136]. In this sense, learning can be seen as the adaptation of a system through interaction with data or an environment, so that it can perform better on future instances of a given problem or set of problems. From this perspective, a learning algorithm uses past experience to produce or adjust another algorithm; that is, it is a meta-algorithm that outputs algorithms [186].

The main learning paradigms in ML that apply these concepts are further discussed next.

2.3.1 Supervised Learning

Supervised learning is one of the most common paradigms in ML. In this setting, the learning algorithm receives examples composed of inputs and corresponding target outputs, and the objective is to learn a mapping that predicts the target associated with a new input [71, 136]. Formally, as defined in 2.1.2, given a training set $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where $\mathbf{x}_n \in \mathcal{X}$ denotes an input example and $y_n \in \mathcal{Y}$ denotes its label, or target, the goal is to learn a function $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ parameterized by θ . The parameters are adjusted so that the model's predictions approximate the desired outputs in the training data.

This paradigm is called “supervised” because the learning process is guided by an external signal that indicates the desired answer for each example. In classification problems, the target y corresponds to a discrete class label. In regression, the target is a continuous value [71, 136]. In both cases, learning consists of choosing a hypothesis from a space of possible functions and optimizing it according to a performance criterion or loss function [186].

A common formalization of supervised learning is empirical risk minimization [71]. Under this view, the model is trained by minimizing the average loss over the training set:

$$\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{n=1}^N \mathcal{L}(f_\theta(\mathbf{x}_n), y_n), \quad (2-29)$$

where $\mathcal{L}$ measures the discrepancy between the model prediction and the target output. For example, regression tasks often use squared error losses, while classification tasks commonly use cross-entropy losses [71, 136]. The central challenge is not only to reduce the error on the training examples, but to learn patterns that generalize to unseen data.

In supervised learning, each input in the training set is assumed to be associated with a corresponding label. However, this dependence on labeled data can become a major limitation, especially in domains where annotation is expensive, ambiguous, or unavailable at scale [71, 106, 136]. This limitation motivates learning paradigms that can exploit data without requiring explicit targets.

2.3.2 Unsupervised Learning

Unsupervised learning considers settings in which the model has access to input data but not to explicit target labels. Given a dataset $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$, the objective is to discover useful structure in the data itself [71, 136, 186]. This structure may take

different forms, such as clusters of similar examples, low-dimensional representations, latent factors, or probability distributions that describe how the data is generated.

From a probabilistic perspective, unsupervised learning can be described as modeling of the input distribution $p(\mathbf{x})$, in contrast to supervised learning, which often models the conditional distribution $p(y | \mathbf{x})$. In this setting, the model must capture regularities in the observed data without being told which categories or outputs are correct [71, 136].

Unsupervised learning forces the model to “explain” the high-dimensional inputs, rather than only the low-dimensional outputs, because the learning algorithm must discover regularities in the structure of the data [136]. If the representation captures useful structure, it should improve sample efficiency and allow good performance with fewer annotations [186]. This makes unsupervised learning important in domains where large quantities of data are available, but reliable labels are scarce.

2.3.3 Self-Supervised Learning

Self-Supervised Learning (SSL) is a paradigm in which unlabeled data is used to construct supervised prediction tasks. Instead of relying on externally provided labels, the model is trained on proxy tasks whose targets are automatically derived from the data’s structure [136]. In this sense, SSL occupies an intermediate position between supervised and unsupervised learning in which the data does not contain annotated labels, but the training objective is still formulated as a supervised prediction problem.

In a typical SSL setting, part of the input is used to predict another part of the same input. For example, a model may be trained to predict masked words or sub-words from their surrounding context, reconstruct missing regions of an image, colorize a grayscale image, predict future frames, recover corrupted signals, or identify whether two transformed examples originate from the same underlying sample [136]. More generally, the original input can be divided into an observed component and a target component, and the model is trained to predict the target from the observed information.

Self-supervised methods are often grouped into two broad families: generative and contrastive approaches [152]. In generative SSL, part of the input is hidden, corrupted, or removed, and the model is trained to reconstruct or predict the missing content. Examples include masked language modeling, and image inpainting tasks [37, 147, 152]. In contrastive self-supervised learning, the model is trained by comparing examples, bringing related representations closer together and separating unrelated ones [24, 152]. For instance, two augmented views of the same image, two related audio segments, or two neighboring sentences can be treated as positive pairs, while unrelated examples are treated as negative pairs [24, 140, 152].

The main motivation behind SSL is that solving a proxy task can force the model to learn representations that are useful beyond the task used during pre-training [136]. Predicting missing elements requires the model to capture dependencies among observed elements, while reconstructing corrupted inputs requires it to model regularities in the data distribution [71, 136]. Similarly, contrastive objectives encourage the model to organize the representation space according to meaningful similarities between examples [24, 92, 106, 140]. Once learned, these representations can be transferred to downstream tasks with limited labeled data, commonly through fine-tuning or linear evaluation [136, 152].

Self-supervised learning has become especially important in modern representation learning because it enables large-scale pre-training from unlabeled corpora [71, 136]. This is particularly relevant for language, audio, vision, and multimodal data, where raw data is abundant but high-quality annotation is expensive [71]. In multimodal learning, self-supervised objectives can also exploit natural correspondences between modalities, such as the relationship between an audio signal and its textual description, an image and its caption, or speech and its transcription [49, 154]. As a result, SSL provides a foundation for learning general representations that can later support classification, retrieval, generation, and cross-modal alignment [49, 140, 154].

Contrastive Learning

Contrastive learning is a family of SSL methods in which the proxy task is defined through comparisons between examples. The model is trained to organize a latent space where related examples are mapped close to one another and unrelated examples are mapped farther apart [24, 92, 106]. Unlike supervised learning, where the training signal is provided by external class labels, contrastive learning derives supervision from relationships automatically constructed from the data.

Because contrastive learning does not rely on predefined labels, it requires a mechanism to define which examples should be considered similar or dissimilar [92, 106]. This is commonly done by constructing anchors, positive examples, and negative examples. Given an anchor sample $\mathbf{x}$, a positive example $\mathbf{x}^+ \sim p^+(\cdot | \mathbf{x})$ is selected so that it shares relevant information with the anchor, whereas negative examples $\mathbf{x}^- \sim p^-(\cdot | \mathbf{x})$ are assumed to correspond to different concepts [106, 152]. Positive pairs are often obtained by applying different transformations to the same input, such as cropping, masking, noise injection, or temporal sampling. They may also arise from naturally related views of the same underlying event, such as neighboring segments in a sequence or paired observations from different modalities. Negative examples are usually obtained from other samples in the mini-batch, from a memory bank, or from a queue of previously encoded representations [24, 92]. These negatives provide the contrastive signal that encourages the model to distinguish the anchor from unrelated or less related examples.

This formulation is closely related to the learning of invariances. When two transformed views of the same sample are treated as a positive pair, the model is encouraged to ignore the variations that distinguish those views while preserving the information they share [106]. For instance, in visual representation learning, two augmented versions of the same image may be used as a positive pair, making the encoder less sensitive to changes in color, cropping, or scale [24]. More generally, the construction of positive pairs determines which variations should be treated as irrelevant, while the choice of negative pairs determines which examples should remain distinguishable [106].

Formally, let $f_\theta(\cdot)$ be an encoder that maps an input $\mathbf{x}$ to a representation $\mathbf{h} = f_\theta(\mathbf{x})$, and let $g_\phi(\cdot)$ be an optional projection head that maps this representation to a contrastive embedding $\mathbf{z} = g_\phi(\mathbf{h})$. A contrastive objective encourages the similarity between an anchor embedding $\mathbf{z}_i$ and a positive embedding $\mathbf{z}_i^+$ to be higher than the similarity between $\mathbf{z}_i$ and a set of negative embeddings $\{\mathbf{z}_j^-\}_{j=1}^N$. A widely used objective is the InfoNCE loss [140], that can be defined as:

$$\mathcal{L}_i = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_i^+)/\tau)}{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_i^+)/\tau) + \sum_{j=1}^N \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j^-)/\tau)}, \quad (2-30)$$

where $\text{sim}(\cdot, \cdot)$ is a similarity function, commonly cosine similarity or a dot product between normalized embeddings, and τ is a temperature parameter. This loss can be seen as a classification problem in which the model must identify the positive example among a set of negative candidates [92, 106].

Several influential SSL methods are based on this principle. Contrastive Predictive Coding (CPC) [140] learns representations by predicting future latent observations using a contrastive objective. SimCLR [24] demonstrates that strong data augmentations, a nonlinear projection head, large mini-batches, and longer training can produce effective visual representations without labels. MoCo [80] addresses the need for many negative examples by using a momentum encoder and a queue-based dictionary, allowing the model to compare each sample against a large and consistent set of negatives.

The contrastive loss functions encourage two complementary properties: alignment and uniformity [196]. Alignment refers to the tendency of the model to map positive pairs to nearby points in the embedding space. Uniformity comes from the negative term and refers to the tendency of the overall representation distribution to spread across the space, often on the unit hypersphere when embeddings are normalized. This perspective might explain why simply pulling positive pairs together is not sufficient to optimally spread the representations in the latent space, as the absence of a mechanism for separating different examples could lead the model to learn a collapsed representation in which all inputs are mapped to the same representation vector [186, 196].

Contrastive learning is relevant to multimodal representation learning, as that

many multimodal datasets naturally provide positive pairs across modalities. In vision-language representation learning, an image and its caption can be treated as a positive pair, while mismatched images and captions act as negatives [93, 154]. Similarly, in audio-text, an audio clip and its textual description can be aligned in a shared embedding space [49]. This makes contrastive learning particularly suitable for cross-modal retrieval and zero-shot transfer, since samples from different modalities can be compared directly through their encoded representations.

2.4 Representation Learning

The learning paradigms discussed previously differ in the type of supervision available to the model and in the objective used during training. However, they share a common goal of transforming raw data into a representation that makes future computation, prediction, or decision-making easier. This process is known as “representation learning”. Rather than relying only on manually designed features, representation learning aims to learn transformations of the data that expose useful structure for downstream tasks [14, 71].

Formally, given an input space $\mathcal{X}$ and a representation space $\mathcal{Z}$, the goal is to learn an encoding function $f_\theta : \mathcal{X} \rightarrow \mathcal{Z}$ that maps an input sample $\mathbf{x} \in \mathcal{X}$ to a latent representation $\mathbf{z} = f_\theta(\mathbf{x})$. This mapping is illustrated in Figure 2.9. The representation space $\mathcal{Z}$ is expected to be more useful, more organized, or more abstract than the original input space [186]. In many cases, $\mathbf{z}$ is lower-dimensional than $\mathbf{x}$, but dimensionality reduction is not the only goal. A representation may also be useful because it makes relevant factors easier to separate, removes nuisance variation, supports prediction, or enables easier knowledge transfer to other tasks [14, 71, 186].

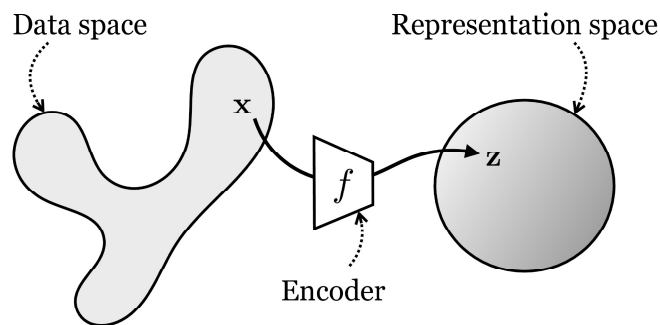


Figure 2.9: Schematic illustration of representation learning, in which an input sample $\mathbf{x}$ from a complex data space is mapped into a representation $\mathbf{z}$ in a latent space. Extracted from Torralba et al. [186].

A useful representation is often considered one that makes subsequent learning or problem solving easier [71, 186]. In deep neural networks, this process occurs progressively, as each layer transforms the input into a new representation and higher layers tend to encode more abstract information [14, 71]. Bengio et al. [14] discuss several desirable properties of representations, including smoothness, multiple explanatory factors, hierarchical organization, shared factors across tasks, temporal and spatial coherence, sparsity, and simple dependencies among factors. These properties reflect the idea that raw data comprises underlying factors of variation, and that a good representation should make these factors easier to identify, separate, and reuse.

Torralba et al. [186] emphasize compression and prediction as central principles of representation learning. According to the authors, a representation should often discard redundant or task-irrelevant information while preserving what is useful for the task at hand. Thus, it should support invariance to nuisance factors, such as noise or other variations that do not affect the semantic content of the input. However, this compression involves a trade-off, since removing too much information may harm tasks that depend on the discarded details [14, 71]. A good representation should also provide a useful substrate for prediction, whether the target is a class label, a missing value, a future observation, or a relationship between examples [186]. This connects representation learning directly to supervised, unsupervised, and self-supervised learning, where labels, input structure, or proxy tasks shape the representations learned by the model [71, 186].

Different families of representation learning methods impose different objectives and constraints on the learned space. Autoencoders learn representations by encoding the input into a latent representation and then reconstructing the original data from it. Contrastive methods learn representations by comparing related and unrelated examples. Clustering methods organize examples into discrete groups. Predictive and imputation-based methods train models to infer missing, future, or otherwise unobserved parts of the data. Pretext-task methods define auxiliary tasks whose solution requires the model to capture meaningful structure in the input [186]. These approaches differ in their mechanisms, but they share the goal of producing representations that are more useful than the raw input for later learning.

Representation learning is a concept directly connected to multimodal learning, given that models need to learn not only useful representations within each modality but also shared spaces in which semantically related information can be compared across modalities [12, 14, 71]. For example, an audio clip and its textual description have very different raw representations, but they can be mapped to nearby areas in a shared latent space when they refer to the same event or concept, which is the basis of many cross-modal retrieval and alignment systems [49, 154]. However, according to Torralba et al., there is no single representation that is optimal for all possible tasks, since a

representation that discards some factors of variation may be useful for one objective but harmful for another [14, 186]. Thus, the quality of the representation determines whether representations can be effectively aligned, whether knowledge can transfer across tasks, and whether models trained under limited supervision can generalize to new scenarios [14, 49, 154].

2.5 What is a Modality?

The definition of “modality” is relevant to understanding multimodality and multimodal representation learning, but it is not always clear what the term refers to. In the ML literature, the term is often associated with sensation, such as vision and touch [12, 141], or used to distinguish between different mechanisms of information encoding [75, 141]. These interpretations are related, but might not be sufficient definitions on their own. Instead, a modality can be seen as a form of information representation that organizes meaning according to particular structures, units, and constraints [30, 141].

2.5.1 Modality and Human Communication

A useful starting point might be human communication, where the link between language and the concept of modality has long been debated. Language is often treated as an abstract symbolic system whose meaning can be separated from its physical expression [30]. Under this view, speech, writing, gesture, sign, and images may appear as different representations of the same linguistic content. Cohn and Schilperoord [30] argue that this view depends on two related assumptions about language: unimodal and amodal.

The unimodal view assumes that language is primarily grounded in one privileged expressive channel. In most linguistic traditions, this channel has been speech, often treated as the prototypical or most natural form of language. Language is then understood primarily as a vocal-auditory phenomenon, while other forms of expression, such as writing, gesture, sign, and images, are treated as secondary, derivative, or auxiliary [30, 67, 86, 109]. Under this view, writing, for instance, may be interpreted as a visual transcription of speech, gestures as additional paralinguistic elements, and images as non-linguistic supports for verbal meaning.

In the amodal view, language is considered a deeper abstract system that can appear in different forms. For example, the same linguistic content may be spoken, written, or signed [26, 30, 77]. What matters most in this view is the underlying linguistic structure, such as syntax, semantics, and symbolic meaning. The physical form used to express that structure is treated as secondary. This view helps explain how the same

message can be expressed across different channels, since the true nature of language would persist regardless of its channel of expression.

These views share the assumption that the channel of expression, or communication, is less important than the linguistic content behind it. This interpretation, however, can hide the fact that each channel can organize and express meaning in different ways. Cohn and Schilperoord [30] challenge this assumption by arguing that the expressive form itself helps shape how meaning is produced and understood.

Cohn and Schilperoord [30] argue for a multimodal understanding of language and communication. In this view, human communication does not depend on only a single privileged channel, nor it is treated solely as an abstract system that exists separately from the ways it is expressed. Instead, communication involves multiple expressive systems. Examples such as speech, sign language, writing, and visual representations are not merely interchangeable outputs of the same abstract content. Each modality has its own structural properties, perceptual constraints, and cognitive organization, which influence how meanings are encoded, combined, and interpreted.

This perspective might change the nuance of what is considered a modality. For instance, speech can be represented as an acoustic signal over time. Sign languages use bodily articulation, spatial organization, simultaneity, and movement. Writing organizes linguistic units as graphic marks. Images may encode meaning by spatial, visual, and sequential structure [30]. These representations may overlap semantically, but they are not reducible to one another. The same event can be represented and experienced in different ways, but each modality preserves and emphasizes distinct aspects of it.

2.5.2 Modality from a Machine's Perspective

This cognitive perspective helps motivate a broader definition for ML. If communication is shaped by the forms that express it, then computational models should not treat modalities as neutral containers of the same abstract information. The modality of an input affects what information is available, how it is encoded, which inductive biases are useful, and what kind of representation the model must learn [12, 75, 141].

In ML, this broader definition motivates a task-relative definition of modality. Parcalabescu et al. [141] argue that both human-centered and machine-centered definitions are insufficient on their own. Human-centered definitions classify modalities by perception, which can be ambiguous in computational settings: written text and images are both visually perceived, yet they are usually treated as different modalities in ML. Machine-centered definitions instead focus on how information is encoded before being processed by a model [75, 141]. This view is useful for representation learning because different encodings often require different inductive biases [12, 14]. However, it

can also make modality too dependent on its technical implementation. For instance, two image formats, such as PNG and JPEG, or two audio formats, such as WAV and FLAC, should not necessarily count as different modalities if they provide the same task-relevant information.

For this reason, Parcalabescu et al. [141] define multimodality in relation to the task being solved. Under this view, an input functions as a modality when it provides a distinct task-relevant form of information representation, characterized by its own structure, units, and encoding process. Thus, modality is not an absolute property of data in isolation. It depends on how the data is represented, what information it carries, and whether that information is relevant for the task.

From the cognitive perspective, modalities shape how meaning is expressed and interpreted. From the machine learning perspective, modalities determine the form of the input, the type of encoder required, and the kind of alignment needed for downstream tasks [12, 14]. Multimodality and multimodal representation learning stands at the intersection of these perspectives, since it aims to build shared latent spaces where different forms of information can be compared, integrated, and transferred across tasks [48, 93, 154].

2.5.3 Multimodality

Following the task-relative definition of modality, multimodality can be understood as the condition in which two or more distinct forms of information representation are involved in the same task. A task becomes multimodal when these inputs provide different kinds of task-relevant information that need to be compared, aligned, integrated, translated, or jointly interpreted [12, 141]. Thus, multimodality depends not only on the number of data sources but also on the relationships among the information each modality provides.

This relationship may involve shared and modality-specific information. Different modalities can refer to the same object, event, or situation, but they do not necessarily encode the same aspects of it. Recent work describes these relations in terms of redundancy, uniqueness, and synergy, since information may be shared across modalities, specific to one modality, or useful only when modalities are considered together [46]. This view reinforces that multimodality is not only a problem of combining heterogeneous data. It also involves understanding what information is shared, what information is modality-specific, and what type of interaction is relevant for the task.

For ML, these relations need to be represented in a form that models can use. This is why multimodality is closely connected to representation learning. A multimodal model must learn representations that preserve useful information from each modality

while making their relations accessible for different tasks. At the same time, recent work shows that shared spaces may still contain modality-specific information, which can affect how well representations from different modalities align [4]. This motivates the study of multimodal representation learning as the problem of building representations that capture both cross-modal relations and modality-specific structure.

2.6 Multimodal Representation Learning

Many real-world learning problems involve data from multiple modalities, such as text, images, video, and audio [75]. As discussed previously, each modality provides a particular form of information representation, with its own structure, units, and constraints. Because of this, different modalities may provide complementary or partially overlapping information about the same object, event, or situation [46, 227]. The goal of multimodal representation learning is to learn representations that make this information useful for downstream tasks, either by comparing, aligning, integrating, or transferring information across modalities [75, 227].

A core challenge in this domain is the “heterogeneity gap”, which arises because feature vectors from different modalities originally reside in disparate, different subspaces [75]. This gap means that even if an audio and a text description refer to the same semantic concept, their raw vector representations will be completely different, hindering any joint processing [75]. As seen in Figure 2.10, the primary objective of multimodal representation learning is to narrow this heterogeneity gap. This is typically achieved by projecting the data from heterogeneous modalities into a common, shared semantic subspace where the representations of semantically similar inputs from different modalities become close to one another [75].

2.6.1 Frameworks for Multimodal Integration

To bridge the heterogeneity gap, various deep learning frameworks have been developed. These can be broadly categorized into three main architectural approaches: joint representation, coordinated representation, and encoder-decoder models [75].

Joint representation aims to project representations from unimodal encoders into a shared semantic subspace where the features are then fused into a single vector. The fusion can be achieved through simple concatenation or, more commonly, by adding or multiplying the transformed modality-specific vectors in a distinct hidden layer. This approach is advantageous for tasks that require a unified representation and can be easily extended to fuse multiple modalities. However, it typically does not allow inference of separate representations for each modality.

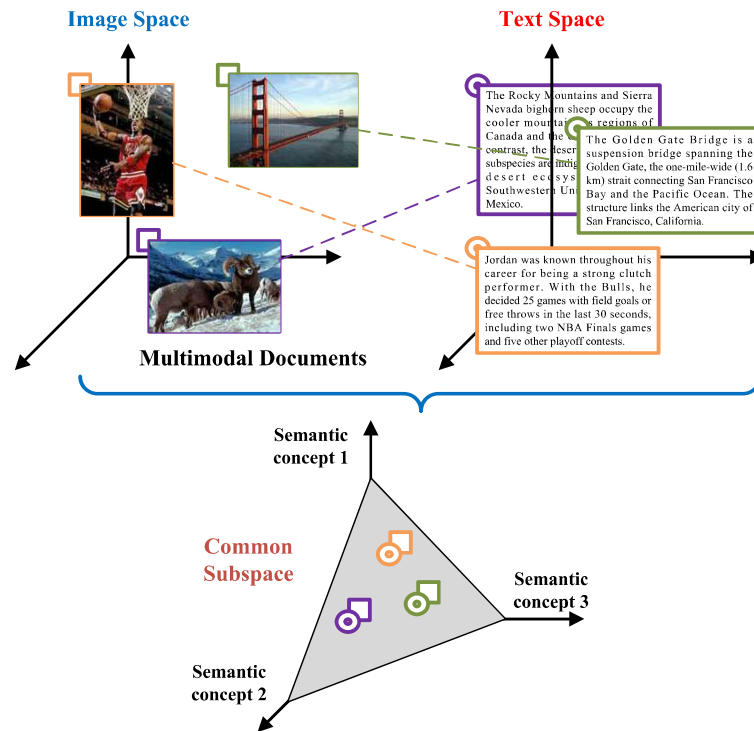


Figure 2.10: Representation of a common subspace learning that aims to project the heterogeneous data of different modalities into a common subspace. Extracted from Guo et al. [75].

Instead of fusing features into a single vector, coordinated representation aims to separate representations for each modality within a coordinated subspace. The representations are not directly merged but constrained to specific cross-modal properties. These constraints typically involve either maximizing the correlation between the representations or maximizing the similarity of paired samples while minimizing it for unpaired ones. A key advantage of this approach is its ability to preserve unique, modality-specific characteristics, as different modalities are encoded in separate networks. One good example of a model that uses coordinated representation is Contrastive Language-Image Pre-Training (CLIP) [154], which is the architectural approach most similar to the models proposed in this research project.

Finally, an encoder-decoder framework is commonly used for multimodal generative and translation tasks, such as generating a text caption from an image or audio from text. An encoder maps the source modality into a latent vector, which is an intermediate representation, and then a decoder uses this latent vector to generate a new sample in the target modality.

2.6.2 Modality Gap

Despite the goal of creating a shared semantic space, contrastive multimodal models often preserve a geometric separation between modalities. This phenomenon is known as the “modality gap”. It describes the tendency of embeddings from different modalities to occupy distinct regions of the learned representation space, even when the inputs are paired and semantically related [113]. In models such as CLIP, image and text embeddings are projected to the same hypersphere, but they may still form modality-specific regions rather than a fully mixed semantic space [53, 113, 173]. A representation of the modality gap between two arbitrary modalities can be seen in Figure 2.11.

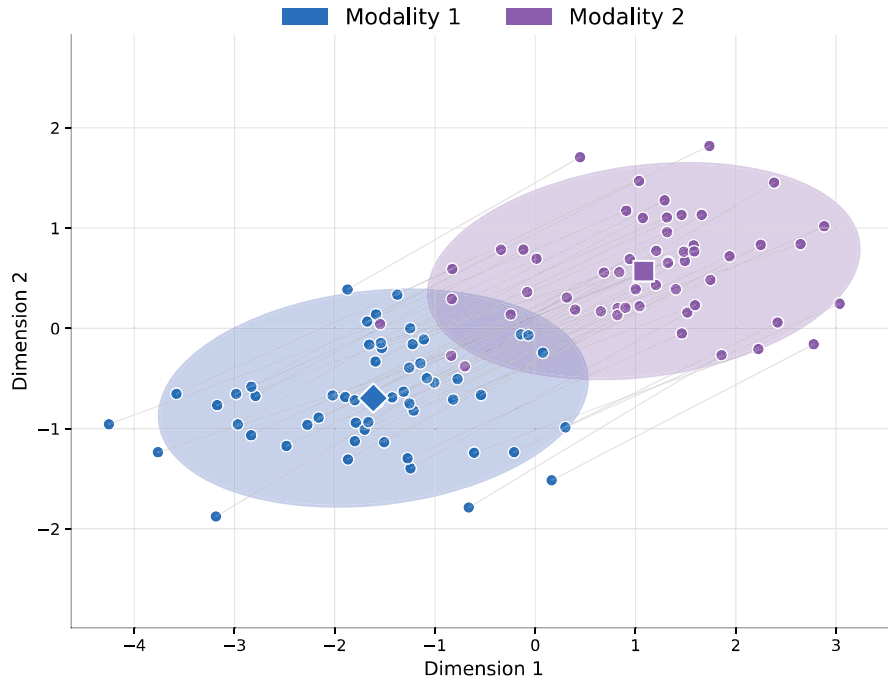


Figure 2.11: *Illustration of a modality gap between two arbitrary modalities.*

The modality gap is related to the heterogeneity gap, but it is not the same phenomenon. The heterogeneity gap refers to the original difference between modalities before alignment, since different modalities have different structures and are encoded in different feature spaces [75]. The modality gap appears after multimodal training, inside the learned shared space. In this case, the model has already learned useful cross-modal relations, but its embeddings still preserve a global separation based on modality. Thus, the modality gap is not only a property of the raw data, but also a property of the geometry learned by the model itself.

Liang et al. [113] provided the first systematic explanation of this phenomenon. They argue that the gap arises from the combination of model initialization and contrastive optimization. At initialization, deep networks tend to map their outputs to narrow regions

of the hypersphere, a behavior described as the “cone effect”. Since multimodal models usually rely on separate encoders, each encoder may start in a different cone. This creates a separation between modalities before training. During optimization, the contrastive objective pulls paired samples closer while preserving some of this separation, especially under the influence of the temperature parameter [113].

Later works refined this interpretation. Shi et al. [173] connect the modality gap to local minima of the CLIP loss (InfoNCE), suggesting that the learned space may settle into patterns where paired examples are sufficiently aligned for the loss, while the two modalities remain separated. Yaras et al. [211] analyze the learning dynamics more directly and show that the gap may increase in the early stages of training due to mismatched pairs. They also argue that the learned temperature can stabilize the gap, making it difficult to close even when training continues. This shifts the interpretation of the modality gap from a static artifact of initialization to a dynamic effect of contrastive training.

Another line of work explains the modality gap in terms of information. Schrodi et al. [166] argue that the gap is strongly connected to information imbalance between modalities. For instance, an image may contain many details that are not described in its caption. Since the caption is incomplete relative to the image, perfect alignment is difficult. The model must balance alignment with uniformity, and the gap can emerge as a by-product of this trade-off. Almud’evan et al. [4] provide a related information-theoretic view, arguing that contrastive losses can preserve modality-specific information in the representation space. Even when information shared across modalities is learned, modality-specific information may remain, contributing to representational misalignment.

This discussion also connects the modality gap with a broader question in multimodal representation learning. It is not always clear what should be aligned. Dufumier et al. [46] argue that standard cross-modal contrastive learning mainly captures redundant information between modalities, while multimodal tasks may also require unique and synergistic information. From this perspective, closing the modality gap is not simply a matter of removing all modality-specific structure. Some modality-specific information may be useful, while other information may harm cross-modal comparison. The problem is therefore not only to reduce the distance between modalities, but to control what kind of information is shared, preserved, or separated.

Several attempts have been proposed to mitigate or control the modality gap. Early strategies considered post-hoc translations in the embedding space, such as shifting one modality toward the other [113]. However, later work argues that simple translations can distort the relative distances between unpaired samples and damage the semantic structure of the space [53]. Other approaches modify the training process itself. Align-CLIP reduces the gap by sharing parameters between the vision and language encoders

and by using a semantically regularized objective over unimodal embeddings [53]. Yaras et al. [211] propose temperature control and modality swapping as training strategies. Almud’evan et al. [4] use an information bottleneck regularization to reduce modality-specific information in the shared space.

The effects of closing the gap are still debated. Some works report better retrieval or cross-modal alignment when the gap is reduced [53, 211]. Other works suggest that the impact depends on the type of downstream task. Grassucci et al. [72] argue that reducing the gap has limited or inconsistent effects on instance-wise tasks such as retrieval, but it can strongly improve group-wise tasks such as clustering, where the absolute organization of the latent space matters more. Mistretta et al. [130] also show that the gap is related to intra-modal misalignment, since CLIP-style training aligns image-text pairs but does not necessarily make image-image or text-text similarities reliable. These results suggest that the modality gap should not be treated only as an error to remove, but as a geometric property whose relevance depends on the task, the modalities, and the type of semantic structure expected from the representation space.

2.6.3 Explicit and Implicit Alignment

One of the main questions in multimodal representation learning is how to align different modalities into a shared latent space. In conventional multimodal representation learning, alignment is often explicit, meaning that the model is directly encouraged to make representations comparable across modalities through its training objective or architecture [154]. One of the most common ways to enforce explicit alignment is through contrastive learning, since paired samples from different modalities are treated as positive pairs and mapped closer in a shared embedding space, while mismatched samples are separated, as discussed in Section 2.3.3. In Audio-Language Models (ALMs), this is the same mechanism used by CLAP-style systems, where audio clips and textual descriptions are explicitly aligned so that cross-modal retrieval and zero-shot classification can be performed by comparing the similarity between latent representations [48, 49].

However, cross-modal alignment does not strictly require direct pairwise observations between the modalities being aligned. Some frameworks use an anchor modality to connect several modalities into the same space. ImageBind [65], for instance, aligns multiple modalities to images and shows that this can produce emergent alignment between modalities that were not directly paired during training. In this case, images act as a binding modality, allowing other modalities to become comparable through their relation to visual data. LanguageBind [224] follows a related idea, but uses language as the central binding modality. It directly maps different modalities to a language-centered representation space.

A more recent line of work studies implicit, or emergent, alignment, where representations become similar even without an explicit cross-modal alignment objective. The Platonic Representation Hypothesis [90] argues that, as models scale and are trained on broader data and tasks, independently trained models may converge toward similar representations of the underlying world. Under this view, different modalities provide different observations of a shared reality. Even when models are trained separately, their representations may become aligned because they capture related semantic structures. Thus, alignment can emerge not only from explicit objectives but also from the overlap between the information encoded by different modalities.

At the same time, recent studies show that emergent alignment should not be treated as universal. Tjandrasuwita et al. [185] argue that the emergence of alignment depends on the characteristics of the modalities and the task. Alignment is more likely to emerge when modalities contain redundant information and when they are less heterogeneous. When modalities provide more unique information, stronger alignment may not improve performance and may even remove useful modality-specific structure.

Other works further refine the interpretation of emergent alignment. Gröger et al. [74] argue that some evidence for global representational convergence can be affected by measurement confounders such as model width and depth. After calibration, they find stronger evidence for convergence in local neighborhood structure than in global geometry. Koepke et al. [101] also show that cross-modal alignment can weaken when evaluation moves from small, one-to-one paired datasets to larger and more realistic many-to-many settings. These findings suggest that different modalities may share coarse semantic organization while still arranging fine-grained information differently.

In summary, explicit alignment refers to mechanisms that directly enforce or encourage cross-modal comparability, such as contrastive objectives, architectural constraints, or anchor modalities. Implicit alignment refers to representational similarity that appears without a direct cross-modal alignment objective. This distinction separates alignment that is intentionally imposed during training from alignment that emerges from the learned structure of the representations.

2.7 Audio-Language Models

Audio-Language Models (ALMs) are multimodal models trained to connect audio signals with natural language. They are designed to process, understand, and reason about audio-centric content using paired audio-text data [182]. These models leverage natural language supervision as an additional cue to learn semantically rich acoustic representations that demonstrate remarkable flexibility across diverse downstream applications [49, 182].

The emergence of ALMs stems from the basic drawbacks of traditional audio classification techniques. While traditional approaches demonstrate proficiency in mapping audio signals to predetermined categories, they exhibit significant constraints when confronting the inherent complexity and variability characteristic of real-world acoustic environments. Natural language supervision offers a better alternative, as it can capture nuanced temporal details, describe multiple overlapping sound events, and convey rich contextual information that fixed tags cannot. For instance, natural language supervision allows for more detailed descriptions, such as “a person’s speech is cut short by a firm knock on the door, while someone in the background is heard coughing”, as opposed to limiting an audio sample to simple categorical labels like “door knock”, “coughing”, or “human speech” [182].

Another important property of ALMs is their ability to support open-vocabulary and zero-shot tasks. Since the model learns relations between audio and language, it can compare an audio clip with textual descriptions that were not part of a fixed training set. This makes ALMs useful for tasks such as zero-shot audio classification, audio-text retrieval, audio captioning, audio question answering, source separation, and audio generation [49, 182]. More recently, Large Audio-Language Models (LALMs) extend this idea by integrating large language models as reasoning or generation components, allowing audio information to be used in instruction-following and conversational settings [182].

2.7.1 Architectural Foundations and Design Principles

The architecture of an ALM depends on how audio and language are encoded, aligned, and used for downstream tasks. Su et al. [182] organize ALM architectures into four main families: Two Towers, Two Heads, One Head, and Cooperated Systems. Among these, the Two-Tower architecture is the most common design for contrastive audio-language representation learning. It uses one encoder for audio and another encoder for text, followed by projection layers that map both modalities into a shared embedding space. This design is conceptually related to vision-language models such as CLIP [154] and ALIGN [93], and is especially useful for retrieval and zero-shot classification, since audio and text embeddings can be computed independently and compared by similarity.

The audio encoder transforms waveforms, spectrograms, or learned acoustic features into representations that preserve relevant sound information. CNN-based models, such as CNN14 from the PANNs (Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition) family [103], are often used to capture local spectral patterns and acoustic structures. Transformer-based models, such as the Audio Spectrogram Transformer (AST) [68] and HTSAT [21], are useful for modeling longer temporal dependencies and global relations in audio signals. Self-supervised audio models, such as

SSAST [70], AudioMAE [89], and BEATs [22], can also be used as audio encoders models since they learn acoustic representations from pretext tasks such as masked reconstruction.

The text encoder maps natural language descriptions, prompts, captions, or class names into semantic representations. Earlier ALMs relied on encoder-only bidirectional language models such as BERT [37] or RoBERTa [117], which provide contextual representations for audio captions and textual labels. More recent models use larger or causal language models as reasoning or generation components. In these systems, language is used to express tasks, ask questions, describe user intentions, and coordinate audio-centered capabilities [155, 182].

Other ALM architectures support stronger interaction between audio and language. Two-Head models add a language model or decoder on top of modality-specific encoders, making them better suited to captioning, question answering, and instruction-following tasks. One-Head models process modalities using a more unified encoder-decoder structure, enabling earlier interaction between audio and text. Cooperated Systems use a Large Language Model (LLM) as a planning module to select or combine specialized audio models based on the task [182]. Figure 2.12 summarizes these architectures.

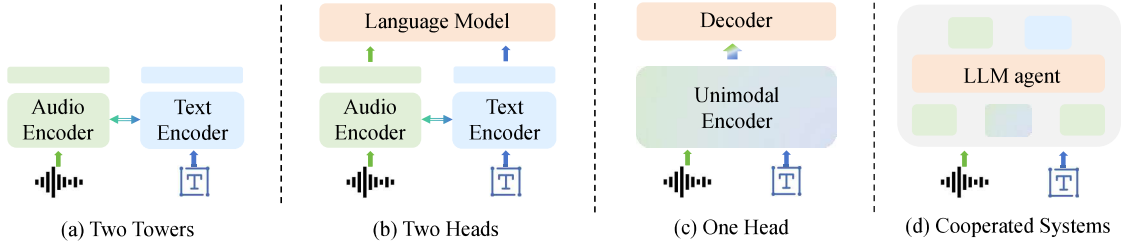


Figure 2.12: Common ALM architectures: (a) Two Towers, (b) Two Heads, (c) One Head, and (d) Cooperated Systems. Extracted from Su et al. [182].

2.7.2 Contrastive Language-Audio Pretraining

Contrastive Language-Audio Pretraining (CLAP) is one of the main foundations of modern ALMs. It adapts the contrastive training paradigm to paired audio-text data. Given an audio clip and its corresponding textual description, a CLAP-style model learns to place both representations close in a shared embedding space, while separating mismatched audio-text pairs [48, 49]. This framework is illustrated in Figure 2.13.

The importance of CLAP lies in showing how natural language can be used as supervision for audio representation learning. Instead of learning only from a fixed set of labels, the model can learn from textual descriptions that express sound events, acoustic

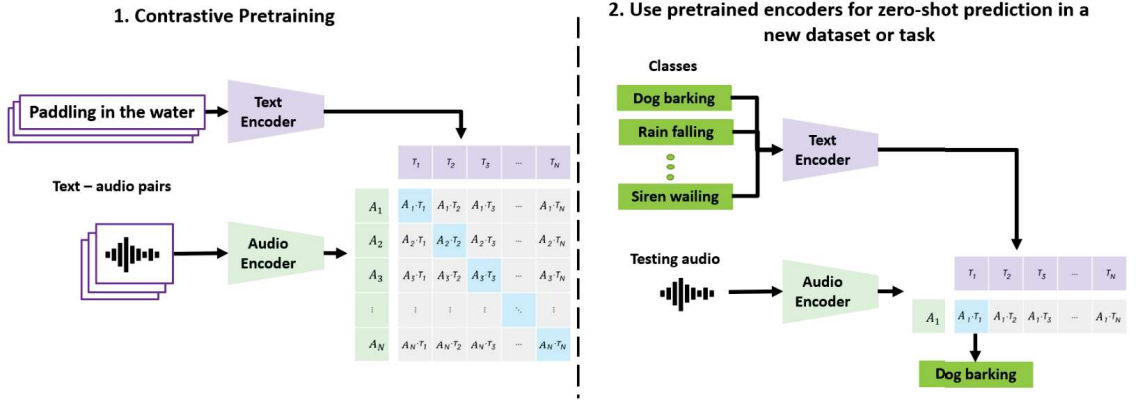


Figure 2.13: An illustration of the CLAP architecture. The diagram shows the contrastive training process on the left and the zero-shot inference process on the right. Adapted from Elizalde et al. [48].

context, and temporal relations. This makes the learned representation useful for open-vocabulary settings. In zero-shot audio classification, class names or prompts are encoded by the text encoder and compared with the audio embedding. In audio-text retrieval, the same shared space allows a text query to retrieve relevant audio clips, or an audio clip to retrieve related captions [48, 49, 182].

Since the original CLAP formulation, several works have extended contrastive audio-language pretraining by scaling datasets, improving encoders, and designing objectives for more fine-grained audio-text alignment [182]. Figure 2.14 presents a timeline of representative audio-text contrastive models from 2023 to 2026, covering both English-centered and multilingual approaches.

2.8 Acoustic Pre-Processing

Sound is a physical phenomenon produced by variations in air pressure that propagate through a medium and can be perceived by a sensory system, such as the human ear or a microphone [110]. In acoustics, the focus is the study of sound itself, while audio is commonly associated with the technical capture, representation, reproduction, storage, and processing of sound [110]. When a microphone is used to capture sound in the air, the pressure variations caused by the sound wave are converted into an electrical signal whose voltage varies continuously over time. This electrical signal is analogous to the acoustic pressure variation and is commonly called “analog audio signal”. In this form, the signal can be represented as a continuous-time function $x(t)$, where t denotes time [110]. Since computers operate on finite discrete numerical representations, this continuous analog signal must be converted into a digital representation before it can be stored or processed [153, 226].

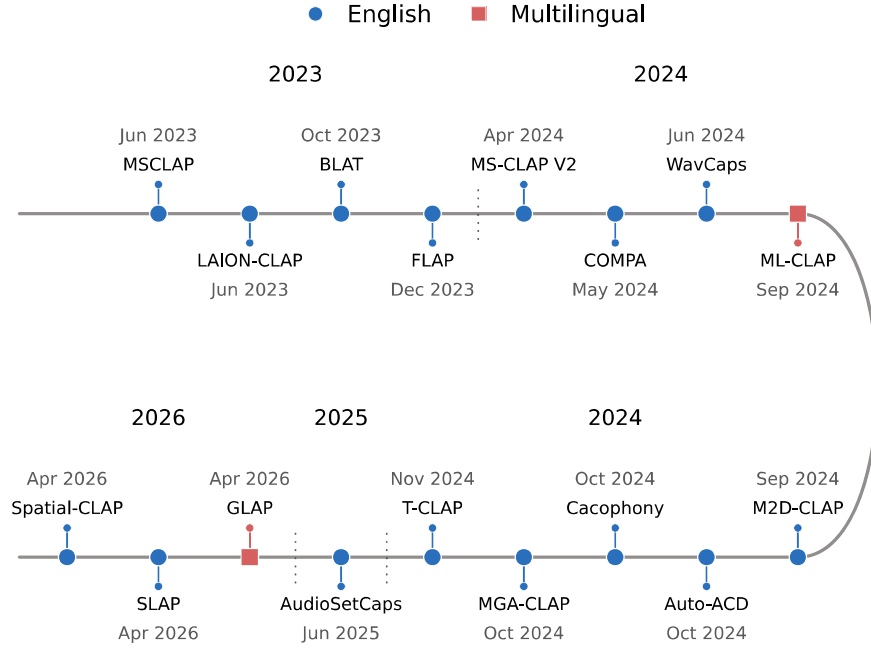


Figure 2.14: *Timeline of representative contrastive Audio-Language Models: MSCLAP [48], LAION-CLAP [201], BLAT [204], FLAP [212], MS-CLAP V2 [50], COMPA [64], WavCaps [127], ML-CLAP [207], M2D-CLAP [138], Auto-ACD [183], Cacophony [225], MGA-CLAP [112], T-CLAP [214], AudioSetCaps [11], GLAP [40], SLAP [126], Spatial-CLAP [169].*

The conversion from analog to digital involves two main operations: sampling and quantization, as illustrated in Figure 2.15. Sampling converts the continuous-time signal $x(t)$ into a discrete-time sequence by measuring its amplitude at regularly spaced time instants. If the sampling period is T_s , then the sampling frequency is $f_s = 1/T_s$. According to the Nyquist-Shannon sampling theorem [139], a band-limited signal with maximum frequency f_B can be sampled without loss of information if the sampling rate satisfies $f_s > 2f_B$, that is, the sampling rate must be bigger than double the highest frequency present in the signal [171, 226]. Frequencies above $f_s/2$ cannot be represented correctly after sampling and may appear as lower-frequency components, a phenomenon known as aliasing [110, 226]. For this reason, audio acquisition and resampling pipelines usually include low-pass filtering before sampling or downsampling [226].

Quantization converts the continuous amplitude values of the sampled signal into a finite set of numerical levels. The bit depth determines the number of available amplitude levels. For example, 16-bit audio represents each sample using 2^{16} possible values. This step makes digital storage and computation possible, but it also introduces quantization error, as can be seen in Figure 2.15 (d), which surfaces as a form of noise when compared

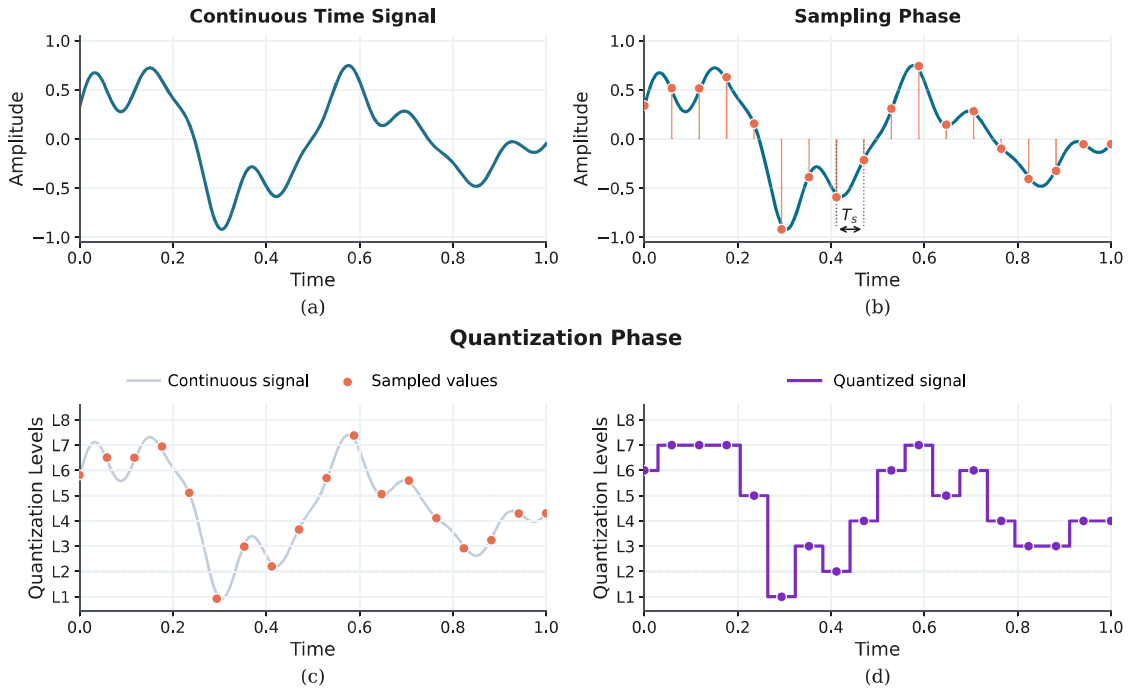


Figure 2.15: Illustration of the sampling and quantization phases for converting an analog signal into a digital signal. Panel (a) shows the original continuous-time signal, while panel (b) shows the sampling phase, in which the signal is measured at discrete time intervals defined by the sampling period T_s . Panel (c) shows the sampled values across the available 3-bit quantization levels, corresponding to $2^3 = 8$ possible levels. Panel (d) shows the resulting quantized signal, where each sampled value is mapped to the nearest discrete quantization level.

to the original continuous signal [153, 226]. Together, sampling rate and bit depth define important properties of the digital waveform, including its temporal resolution, frequency range, dynamic range, and storage size.

A digital audio signal is commonly represented as a waveform, which describes the variation of signal amplitude over time and is therefore a time-domain representation [110]. This representation preserves the temporal evolution of the signal and is useful for observing duration, silence, transients, periodicity, and amplitude variations [59, 172]. However, many acoustic features are more directly characterized in terms of frequency, like the perception of pitch, bass, treble, and warmth of a sound [59, 110]. For this reason, audio processing often also represents signals in the frequency domain, where a signal is described as a combination of sinusoidal components with different frequencies, amplitudes, and phases. Fourier analysis provides the mathematical basis for this representation. For a finite discrete-time signal $x(n)$ of length N , the Discrete Fourier Transform (DFT)

is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad k = 0, 1, \dots, N-1, \quad (2-31)$$

where (k) is the discrete frequency-bin index, corresponding to the normalized angular frequency $(2\pi k/N)$.

The DFT decomposes the signal into frequency bins and indicates how much energy is present at each frequency component [153]. In practice, the DFT is commonly computed using the Fast Fourier Transform (FFT), a class of efficient algorithms that makes spectral analysis computationally practical for long signals [32, 153]. While the frequency-domain representation is useful for identifying the spectral content of a signal, it does not directly show when each frequency component occurs. This is a limitation for audio signals since real-world sounds are usually non-stationary and change over time.

To represent how frequency content evolves, audio is commonly transformed into a time-frequency representation. One of the most widely used approaches is the Short-Time Fourier Transform (STFT). The STFT divides the waveform into short, often overlapping frames, applies a window function to each frame, and computes a Fourier transform for each windowed segment. This formulation can be written as:

$$X(m, k) = \sum_{n=0}^{N-1} x(n + mH)w(n)e^{-j2\pi kn/N}, \quad (2-32)$$

where m is the frame index, k is the frequency-bin index, $w(n)$ is the analysis window, N is the window length, and H is the hop size between consecutive frames.

A spectrogram is a 2D visual representation derived from the STFT, usually defined from the magnitude or power of the complex STFT coefficients:

$$S(m, k) = |X(m, k)|^2. \quad (2-33)$$

In a spectrogram, one axis represents time, another axis represents frequency, and the value at each time-frequency bin represents the energy or magnitude of the signal at that location. The parameters used to compute a spectrogram define a trade-off between time and frequency resolution. A longer analysis window gives better frequency resolution, since each Fourier transform observes a longer segment of the waveform. However, it also reduces temporal resolution, since fast events become less precisely localized in time. A shorter window improves temporal localization but provides coarser frequency resolution.

A common variant of the spectrogram is the Mel-Spectrogram. The Mel-Spectrogram applies a bank of triangular filters to the spectrogram's frequency axis, with filter spacing based on the Mel scale. The Mel scale is a perceptual frequency scale de-

signed to approximate how humans perceive pitch differences, with higher resolution at lower frequencies and lower resolution at higher frequencies [178]. A common conversion from frequency in Hertz to the mel scale is:

$$\text{mel}(f) = 2595 \log_{10} \left(1 + \frac{f}{700} \right). \quad (2-34)$$

Given a spectrogram $S(m, k)$ and a mel filter bank $H_r(k)$, the Mel-Spectrogram can be computed as:

$$M(m, r) = \sum_k H_r(k) S(m, k), \quad (2-35)$$

where r indexes the Mel bands. In many systems, a logarithm is then applied to obtain a Log-Mel Spectrogram:

$$\hat{M}(m, r) = \log(M(m, r) + \epsilon), \quad (2-36)$$

where ϵ is a small constant used for numerical stability. The logarithmic compression reduces the dynamic range of the representation and is consistent with the fact that perceived loudness is not linear with respect to physical amplitude [110].

Mel-spectrograms are widely used in speech, music, and environmental sound processing because they provide a compact time-frequency representation that organizes acoustic energy according to a perceptually motivated frequency scale [110, 172]. In deep learning pipelines, log-mel spectrograms are commonly represented as two-dimensional time-frequency maps and processed similarly to image-like inputs, often with a single input channel [68, 83, 103, 150]. Figure 2.16 illustrates three common representations of the same audio signal: the waveform in the time domain, the spectrogram in the time-frequency domain, and the mel-spectrogram using a perceptually motivated frequency scale.

In summary, audio pre-processing defines how an audio signal is transformed from a raw waveform into a representation that computational models can efficiently use. Pre-processing choices determine which temporal and frequency information is preserved, compressed, or discarded. Therefore, these steps are part of the representation design that shapes how audio information is made available for learning.

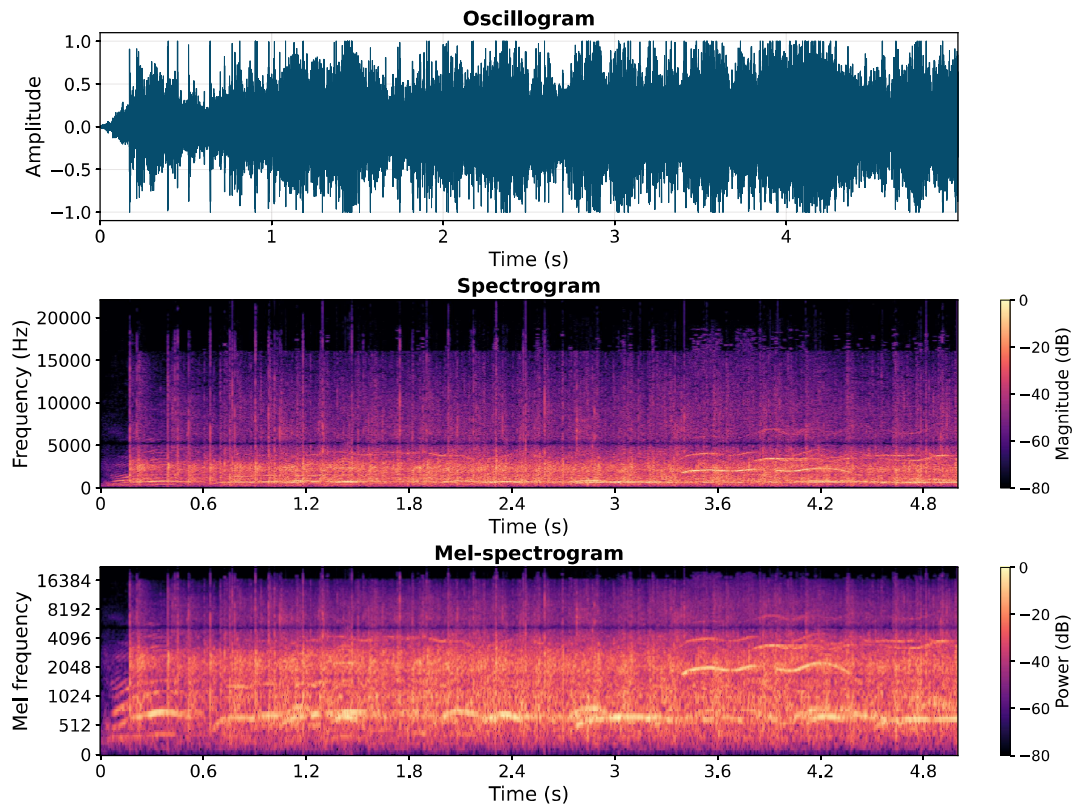


Figure 2.16: Examples of common audio representations computed from the same input signal. Panel (a) shows the oscillogram, or waveform, which represents amplitude variation over time. Panel (b) shows the spectrogram, which represents the distribution of spectral energy across time and frequency. Panel (c) shows the mel-spectrogram, where the frequency axis is mapped to the mel scale to provide a perceptually motivated time-frequency representation.

Related Work

This chapter reviews the main research areas that support the contributions of this dissertation. The proposed frameworks address two complementary directions in multimodal representation learning. CACARA focuses on efficient multimodal expansion, in which a new modality is added to an already-aligned multilingual image-text model at reduced training cost. PALMA focuses on bimodal audio-language alignment, with emphasis on multilingual retrieval, encoder selection, training strategies, and the use of representations from multiple encoder layers. Therefore, this chapter first discusses contrastive ALMs and their recent extensions, then reviews multilingual audio-language learning, and finally situates the work in relation to broader multimodal models that connect vision, language, audio, and other modalities.

3.1 Contrastive Audio-Language Pretraining

The Contrastive Language-Audio Pretraining (CLAP) [48] model set a new standard for learning audio representations from natural language supervision. Inspired by the success of its vision-language counterparts, CLIP [154] and ALIGN [93], CLAP employs a dual-encoder architecture that projects audio and its corresponding textual descriptions into a shared multimodal space. This alignment is achieved through a contrastive learning objective, that learns the similarity between paired audio-text samples. The CLAP model was trained with a dataset of approximately 128k audio-text pairs drawn from existing annotated resources like AudioCaps [99], Clotho [45], FSD50k [57], and MACS [133]. Despite the limited data scale compared to vision-language models, CLAP showed zero-shot capabilities that achieved state-of-the-art (SOTA) performance on downstream tasks in diverse domains, such as sound event categorization and audio retrieval, without any task-specific fine-tuning. Its main contribution was to show that natural language can be a flexible way to learn general-purpose audio representations.

The initial success of CLAP highlighted the scarcity of large-scale, high-quality audio-text datasets, which became a bottleneck. Addressing this challenge became a

central focus of subsequent research, leading to automated pipelines for dataset creation and augmentation.

Subsequent work expanded the CLAP paradigm by increasing training data, improving encoders, and refining the training objective. LAION-CLAP [201] introduced LAION-Audio-630K, a large-scale collection of audio-text pairs, and investigated different audio and text encoders for contrastive pretraining. It also proposed keyword-to-caption augmentation, in which short tags from datasets such as AudioSet [62] are converted into full-sentence descriptions using a language model, addressing one of the main limitations of audio-text datasets, as many contain only labels or tags rather than natural-language captions. LAION-CLAP also uses feature fusion to handle variable-length audio inputs better, making the model more suitable for retrieval and zero-shot audio classification.

A later extension, MS-CLAP V2 [50], increased the training scale to millions of audio-text pairs and modified both sides of the architecture. Instead of using an audio encoder trained only for sound event classification, it adopted an encoder trained across multiple audio tasks. On the language side, it used an autoregressive decoder-only model instead of the more common encoder-only language models. This work shows that both data scale and encoder design strongly influence the quality of audio-language representations. Additionally, the authors report that scaling the number of audio-text pairs does not uniformly improve all tasks, since the diversity, domain distribution, and quality of captions affect downstream generalization.

3.2 Data Scaling and Automatic Annotation

A recurring limitation in audio-language learning is the scarcity of high-quality paired audio-text data. While image-text learning benefits from very large web-scale datasets, audio-language datasets are smaller and harder to curate, given the high cost of annotation and the lack of audio-text pairs available on the web [204, 221, 222]. Audio clips often come with weak labels, tags, titles, or noisy metadata instead of descriptions that accurately refer to the acoustic content. For this reason, several works have focused on automatic caption generation and data cleaning.

BLAT [204] addresses this problem by generating captions directly from audio. The method uses AudioSet tags to guide an audio captioning model, producing synthetic audio-text pairs for pretraining. They did not use visual cues present in AudioSet videos because visual pivots can introduce modality mismatch where an image or video frame may contain objects that are not audible, while relevant sounds may not be visible. By generating captions from audio under tag guidance, BLAT reduces dependence on visual cues and produces data more directly aligned with the audio modality.

WavCaps [127] proposes another strategy based on LLMs. It collects weakly labeled audio clips and raw descriptions from web sources, then applies a three-stage pipeline to filter noisy text and transform unstructured descriptions into cleaner captions, producing around 400k audio-caption pairs.

Auto-ACD [183] extends automated data construction by combining multiple pretrained models. Its pipeline extracts information using audio tagging, image captioning, and object detection models, then uses an LLM to aggregate it into audio descriptions. This produces a dataset with around 1.5 million audio-caption pairs. While the inclusion of visual information can provide environmental context, it can also introduce a mismatch between visible and audible content.

AudioSetCaps [11] further advances automated caption generation by integrating ALMs, LLM-based caption generation, and CLAP-based refinement. The use of prompt chaining allows the pipeline to extract more detailed audio information before generating captions, while the refinement stage is designed to reduce hallucinated or weakly grounded descriptions, scaling audio-caption data to millions of pairs.

3.3 Model Design and Multi-Objective Learning

Beyond data scaling, recent work has investigated how architecture and training strategies affect audio-language representations. Early CLAP-style models commonly used CNN-based or Transformer-based audio encoders and encoder-only text models. Later works show that these choices can significantly affect zero-shot classification, retrieval, and transfer learning performance.

FLAP [212] focuses on efficient language-audio pretraining by combining contrastive learning, masking, and reconstruction. Instead of processing all spectrogram tokens, FLAP randomly drops audio tokens during training, reducing computation while still aligning the remaining audio representation with text. It also reconstructs masked audio tokens, adding an auxiliary objective that encourages the model to preserve information about the acoustic input. This design connects contrastive audio-language learning with masked autoencoding [79], showing that audio-language representations can benefit from both cross-modal and self-supervised objectives.

Cacophony [225] also combines contrastive alignment with self-supervised pretraining. It first trains the audio encoder on audio-only data using a masked autoencoder objective, then uses this encoder for contrastive audio-text training with an auxiliary captioning objective. This two-stage design addresses the limited scale of clean audio-text datasets by exploiting larger audio-only corpora before alignment with language. The use of an auxiliary captioning objective also provides a stronger training signal than global contrastive alignment alone.

M2D-CLAP [138] combines Masked Modeling Duo with CLAP to learn representations that are useful for both zero-shot inference and transfer learning. Language-aligned representations are useful for open-vocabulary tasks, but not all audio problems can be naturally expressed as text classification. Regression tasks, domain-specific audio analysis, and some low-level acoustic tasks may still benefit from conventional audio representations. M2D-CLAP therefore explores a representation that can serve both as an audio-language, and as a general-purpose audio representation.

The vision-language literature also provides relevant training strategies. Locked-image Tuning (LiT) [217] showed that a strong pretrained image encoder can be frozen while the text side is trained to align with it. This reduces training cost and separates representation learning from cross-modal alignment. SigLIP [216] proposed a sigmoid loss as an alternative to the standard softmax contrastive objective, reducing the dependence on global batch normalization and simplifying large-scale distributed training. Although these methods were developed in the image-text domain, they are relevant to audio-language learning because they address two central issues also present in ALMs: how to reuse pretrained encoders efficiently and how to train contrastive models under computational constraints.

3.4 Compositional Alignment

Although CLAP-style models are effective for retrieval and zero-shot classification, standard contrastive learning usually operates on global audio and text embeddings. This can limit the model’s ability to capture fine-grained relations between sound events, temporal order, attributes, and spatial information, representing concepts like bag-of-words [102]. Several recent works address these limitations by modifying the data, the loss function, or the representation structure.

COMPA [64] studies compositional reasoning in ALMs. It introduces benchmarks to evaluate whether models can distinguish audio-caption pairs that contain the same events but differ in event order or in attribute binding. The work shows that existing ALMs may perform close to chance in these settings, suggesting that global contrastive alignment can encourage a bag-of-events representation rather than a representation that captures relations among events. COMPA-CLAP addresses this with composition-aware hard negatives and a modular contrastive loss.

T-CLAP [214] focuses specifically on temporal information. It uses LLMs and audio mixing strategies to generate temporal-contrastive captions, then applies a temporal-focused contrastive loss to improve sensitivity to event order. This addresses a common limitation of CLAP-style models, in which captions such as “a dog barking followed by a person speaking” and “a person speaking followed by a dog barking” may

contain the same words but describe different acoustic sequences. T-CLAP shows that temporal structure requires explicit attention in both training data and objective design.

MGA-CLAP [112] addresses the mismatch between coarse global embeddings and local audio-text correspondences. It introduces a multi-grained alignment strategy that uses a shared codebook to connect local audio frames and textual units. The model also uses locality-aware processing and hard-negative guidance to improve alignment at different levels of granularity.

Spatial-CLAP [169] expands the scope of audio-language alignment to spatial audio. Standard CLAP models are usually designed for monaural or single-source conditions, which limits their ability to encode where events occur. Spatial-CLAP introduces a content-aware spatial encoder and spatial contrastive learning to connect sound content to spatial location, especially in multi-source settings, demonstrating that audio-language representations can be extended beyond event semantics to include spatial properties of the acoustic scene.

3.5 Multilingual Contrastive Audio-Language Models

Most ALMs are trained and evaluated primarily in English. This limits their applicability in multilingual settings, especially for retrieval systems where users may search for audio using different languages.

ML-CLAP [207] is one of the first works to explicitly address multilingual audio-text retrieval. It uses automatic translation to extend English captions into seven additional languages and adopts SONAR [47] as a multilingual text encoder. The work also introduces language enhancement and consistent ensemble distillation to improve multilingual and variable-length retrieval. A key result is that a relatively small amount of translated multilingual data can improve retrieval performance, including in English, suggesting that multilingual supervision achieved through translation can enrich the semantic structure of the shared embedding space.

GLAP [40] expands this direction by targeting multilingual and multi-domain audio-language representation learning. While many CLAP models focus mainly on environmental sounds, GLAP also considers speech-related content and evaluates multilingual capabilities across tasks such as keyword spotting. GLAP highlights the need for ALMs that are not only multilingual in the text encoder but also more general across audio domains.

3.6 Beyond Two Modalities

The second group of related works concerns models that connect more than two modalities.

ImageBind [65] extends CLIP by creating a unified embedding space for six modalities: images, text, audio, depth, thermal, and IMU data. Using images as an anchor, it aligns other modalities via natural pairings, removing the need for exhaustive cross-modal supervision and presenting the concept of emergent alignment. While ImageBind’s image-anchored alignment strategy is effective, it also introduces limitations, as some modalities may have weaker natural associations with images. As highlighted by [224], using a shared backbone for all modalities (except text) instead of modality-specific may underutilize the unique representational capacities of each data type, potentially hindering performance when adding new modalities.

LanguageBind [224] uses language as the central modality by leveraging a frozen language encoder. Training efficiency is improved through Low-Rank Adaptation (LoRA) [88] and contrastive learning. It surpasses ImageBind in classifying infrared, depth, and audio data. However, although LanguageBind adopts text as the anchor modality (similar to our approach), it relies on Vision Transformers (ViT) [42] as a shared backbone across modalities. This design, echoing a key limitation of ImageBind, reduces flexibility and may limit the use of specialized pre-trained encoders tailored to modality-specific features.

Other multimodal systems use stronger joint training across modalities. VALOR [115] studies vision-audio-language pretraining with objectives such as multi-modal grouping alignment and multimodal grouping captioning. These objectives support retrieval, captioning, and question-answering tasks by encouraging interaction among visual, audio, and textual streams. VAST [23] extends this direction by incorporating vision, audio, subtitles, and text for video understanding. By jointly optimizing several modalities, these models can learn rich cross-modal interactions. However, joint training also increases computational costs and depends on datasets where multiple modalities are simultaneously available.

By incorporating subtitles and auxiliary modalities, Vision-Audio-Subtitle-Text omni-modality foundation model (VAST) [23] addresses the limitations of earlier models that neglected these complementary streams, emphasizing the value of diverse inputs for video understanding. VAST departs from anchor-based methods (e.g., ImageBind and LanguageBind) by jointly optimizing all modalities (video, audio, text), enabling deeper cross-modal interaction. While effective, this joint approach increases computational demands and hinders scalability, especially for low-resource modalities.

Multilingual Multimodal Pre-training (MLMM) [218] tackles the dominance of

English in models by integrating large-scale multilingual image-text data. It blends pre-training and generalization strategies, employing four objectives: Image-Text Matching for coarse alignment, Masked Language Modeling for fine-grained understanding, and both Masked Region Feature Regression and Classification for vision-language alignment. MLMM shows strong cross-lingual transfer. It also explores resource-efficient generalization via multilingual knowledge distillation and acquisition, achieving state-of-the-art results.

Connecting Multi-modal Contrastive Representations (C-MCR) [197] aligns modalities via overlapping anchors using inter- and intra-MCR. However, adding new modalities requires multiple fully trained bimodal models, new encoders, and a full alignment procedure, increasing complexity and computational cost.

3.7 Positioning CACARA and PALMA in the Literature

The related literature reveals two complementary directions in multimodal representation learning. The first direction concerns bimodal audio-language alignment, where contrastive ALMs have progressed through larger datasets, synthetic captioning pipelines, stronger encoders, and more specialized objectives for temporal, spatial, compositional, and fine-grained alignment. As summarized in Table 3.2, most existing works reuse pre-trained encoders and focus on improving audio-text alignment, but only a small number explicitly address multilingual support. Moreover, previous approaches have generally not provided a systematic comparison of different audio and multilingual text encoder combinations, nor have they explicitly explored the aggregation of internal representations. In addition, recent and successful efficiency- and performance-oriented strategies from related areas, particularly vision-language modeling, remain underexplored in multilingual audio-text learning. PALMA is positioned within this gap by investigating multilingual audio-text retrieval, examining synergies between different encoder combinations, analyzing the impact of efficiency-oriented methods adapted from vision-language models, and introducing a multi-layer aggregation mechanism to capture complementary information from different levels of the audio encoder.

The second direction concerns multimodal expansion, in which models connect more than two modalities through joint pretraining, anchor-based alignment, or representation-space connections. As shown in Table 3.1, models such as ImageBind, LanguageBind, VALOR, VAST, MLMM, and C-MCR demonstrate different strategies for aligning modalities and transferring information across representation spaces. However, many of these approaches require large-scale multimodal supervision, joint optimization across multiple modalities, or additional training whenever a new modality is introduced. CACARA is positioned in this context as an efficient modality-expansion strategy. Instead

Model	Multilingual support	Free joint learning	Pretrained encoders	Reduced training cost	Scalable to new modalities
CLIP [154]	✗	✓	✗	✗	✗
CAPIVARA [165]	✗	✓	✓	✓	✗
ImageBind [65]	✗	✓	✗	✗	✗
LanguageBind [224]	✗	✓	✗	✗	✗
VALOR [115]	✗	✗	✗	✗	✗
VAST [23]	✗	✗	✗	✗	✗
MLMM [218]	✓	✗	✓	✗	✗
C-MCR [197]	✗	✓	✓	✗	✗
CACARA	✓	✓	✓	✓	✓

Table 3.1: Comparison of CACARA with representative multimodal and multilingual models. ‘Multilingual support’ refers to the model’s ability to operate in multiple languages. ‘Free joint learning’ indicates models that do not require fully aligned data across three or more modalities. ‘Pretrained encoders’ denotes reuse of pre-trained encoders without full retraining. ‘Reduced training cost’ refers to models designed to reduce resource usage. ‘Scalable to new modalities’ identifies models that allow adding new modalities without re-training the full architecture.

of training a complete tri-modal model from scratch, it adds audio to a frozen multilingual image-text model and investigates whether the new audio modality can inherit multilingual capabilities from the existing text-aligned representation space.

Together, PALMA and CACARA address different aspects of the same broader problem. PALMA explores the design of multilingual bimodal audio-language representations, while CACARA studies how to expand an existing multilingual multimodal space with a new modality at reduced training cost. Both contributions respond to limitations identified in the literature and summarized in Tables 3.1 and 3.2. Current multimodal systems often depend on English-centered data, large-scale supervision, and expensive training procedures. This dissertation contributes to a more efficient and linguistically inclusive approach to multimodal representation learning by reusing pretrained encoders, reducing alignment costs, and supporting cross-lingual generalization.

Work	Multilingual support	Encoder comparison	Pretrained encoders	Reduced training cost	Multi-layer aggregation
CLAP (MS-CLAP) [48]	✗	✗	✓	✗	✗
LAION-CLAP [201]	✗	✓	✓	✗	✗
BLAT [204]	✗	✗	✓	✓	✗
FLAP [212]	✗	✗	✓	✓	✗
MS-CLAP V2 [50]	✗	✓	✓	✗	✗
COMPA [64]	✗	✗	✓	✗	✗
WavCaps [127]	✗	✗	✓	✗	✗
ML-CLAP [207]	✓	✗	✓	✗	✗
M2D-CLAP [138]	✗	✗	✓	✗	✗
Auto-ACD [183]	✗	✗	✓	✗	✗
Cacophony [225]	✗	✗	✓	✗	✗
MGA-CLAP [112]	✗	✗	✓	✗	✗
T-CLAP [214]	✗	✗	✓	✗	✗
AudioSetCaps [11]	✗	✗	✓	✗	✗
GLAP [40]	✓	✗	✓	✗	✗
SLAP [126]	✗	✗	✓	✗	✗
Spatial-CLAP [169]	✗	✗	✓	✗	✗
PALMA	✓	✓	✓	✓	✓

Table 3.2: Comparison of PALMA with other contrastive-based ALMs. ‘Multilingual support’ indicates whether the work explicitly supports audio-language retrieval or representation learning beyond English. ‘Encoder comparison’ indicates whether the work systematically evaluates different audio and/or text encoders. ‘Pretrained encoders’ denotes the use of pretrained unimodal or multimodal encoders. ‘Reduced training cost’ refers to strategies that avoid fully training all components or reduce the computational cost of alignment. ‘Multi-layer aggregation’ indicates whether the method explicitly combines representations from multiple encoder layers.

Methodology

This chapter presents the methodology used to develop and evaluate the proposed frameworks. As discussed in the previous chapters, this dissertation investigates two complementary directions in multimodal representation learning: the systematic design of multilingual audio-language models and the efficient expansion of multilingual multimodal spaces.

The first direction is addressed through PALMA, a framework focused on multilingual audio-language alignment. It investigates how different audio and text encoders, training strategies, and multilingual supervision mechanisms affect the construction of a shared audio-text embedding space. In addition, PALMA introduces the Mixture-of-Layers (MoL) mechanism, which aggregates representations from multiple layers of the audio encoder rather than relying solely on its final-layer output. This design allows the model to exploit complementary information encoded at different levels of abstraction, improving the representation used for cross-modal retrieval and zero-shot transfer.

The second direction is addressed through CACARA, a framework designed to expand a frozen multilingual image-text model with audio. Rather than training an audio-language model in isolation, CACARA aligns audio with a previously trained multilingual multimodal space. This alignment is performed using English audio-text data, while the underlying text encoder already contains multilingual capabilities. As a result, CACARA studies whether audio can be integrated into an existing multilingual image-text representation space in a cost-efficient way, enabling multilingual audio retrieval and broader cross-modal interaction without requiring full retraining of the original model.

Therefore, PALMA and CACARA are not treated as independent solutions to unrelated problems. PALMA studies the internal design of multilingual audio-language alignment, while CACARA studies how audio-language alignment can be used to expand an existing multilingual multimodal representation space. They provide two perspectives on the same broader objective: building audio-centered multimodal representations that reduce training cost, reuse pretrained encoders, and support cross-lingual generalization.

The remainder of this chapter describes the methodological foundations, architectural choices, training procedures, and evaluation protocols used in both frameworks.

4.1 CACARA

In this section, we present the details for the model CACARA (/ka.'ka.ra/), a name derived from the Tupi word for “to get close”, which reflects the model’s ability to bring different modalities and languages together. CACARA is a multimodal and multilingual architecture trained through emergent alignment learning, enabling the seamless integration of new modalities into an existing bimodal/multimodal model without requiring full retraining. Likewise, our approach extends the model’s linguistic capabilities while preserving previously learned knowledge. Multimodal and multilingual properties emerge through alignment learning, leveraging prior training to enhance and synchronize multiple modalities and languages.

4.1.1 Introduction

Early breakthroughs in computer vision, driven by convolutional neural networks [104], were followed by advances in natural language processing, culminating in Transformer networks [192]. Beyond images and text, deep learning has achieved state-of-the-art performance in audio [190], sensor data [194], and tabular data [5], excelling in classification, retrieval, and generation tasks.

Real-world applications, however, often involve complex interactions between multiple data types. For instance, video understanding encompasses the joint processing of visual and auditory information [66]. These models integrate complementary information from different modalities, as exemplified by Contrastive Language-Image Pre-training (CLIP), which learns a joint representation space for images and text, enabling cross-modal retrieval and zero-shot classification. The benefits of multimodal learning extend beyond simple fusion, uncovering latent relationships and contextual cues that are not apparent in individual modalities [12]. However, training such models is challenging due to the need for synchronized data and the high cost of annotated datasets. One promising approach is implicit learning, in which the model learns cross-modal relationships without strict temporal alignment by leveraging inherent correlations and statistical dependencies between modalities [2].

Multilingualism adds another layer of complexity. Supporting multiple languages not only expands accessibility but also enriches models with diverse linguistic structures [31]. Yet, low-resource languages remain underrepresented due to data scarcity and limited computational resources [94]. Current research disproportionately favors high-resource languages such as English, neglecting the needs of underrepresented linguistic communities.

The intersection of multimodality and multilingualism presents both opportunities and challenges. A key concern is the computational cost of training and deploying

large-scale models, restricting access to well-resourced institutions [179, 180]. Therefore, there is a pressing need for innovative training methodologies and model architectures that can effectively leverage multimodal and multilingual data while minimizing computational overhead.

CACARA introduces a multimodal and multilingual model that addresses these challenges through two key strategies: emergent alignment learning and a modified Locked-image Text Tuning (LiT) [217] protocol. These strategies reduce training costs while preserving high performance.

We propose a new modality integration strategy that avoids retraining all encoders. Specifically, we fine-tune the audio encoder using English data while keeping the text encoder frozen to leverage its cross-lingual representations. This approach enables multilingual audio-text alignment without the computational cost of multilingual audio pre-training. To our knowledge, no prior work has explored this alignment-based strategy.

The core contribution of CACARA is to demonstrate that aligning an audio encoder with a frozen multilingual text encoder can produce two forms of emergent alignment. First, it enables multimodal alignment between audio and images, even though these modalities are never explicitly paired during training. Second, it enables emergent multilingual capabilities, since the newly aligned audio modality can inherit the cross-lingual structure already present in the pretrained text encoder. By leveraging this frozen multilingual representation space, CACARA supports audio-text alignment in more than 100 languages without requiring multilingual audio data, multilingual audio-text pairs, or retraining of the original model. This result provides a scalable and cost-effective strategy for extending multimodal systems to multilingual settings, and represents a significant outcome of the proposed alignment procedure. Further details on the training stage and how emergent alignment is leveraged are illustrated in Figure 4.1.

4.1.2 CACARA Model

The CACARA model integrates multimodal and multilingual learning through three encoders: image, text, and audio (Figure 4.1). The image encoder is based on a Vision Transformer (ViT) [42], while the text encoder utilizes XLM-RoBERTa (base version) [31]. The audio encoder, built upon BEATs [22], is incorporated via emergent alignment learning. The image and text encoders are initialized with pre-trained OpenCLIP [91] encoders. We keep the image and text models frozen to preserve the high-quality image-text alignment from OpenCLIP pre-training, optimizing only the audio encoder’s alignment within the shared feature space of the text representations, as illustrated in Figure 4.1.

Training follows the contrastive learning formulation introduced in Section 2.3.3,

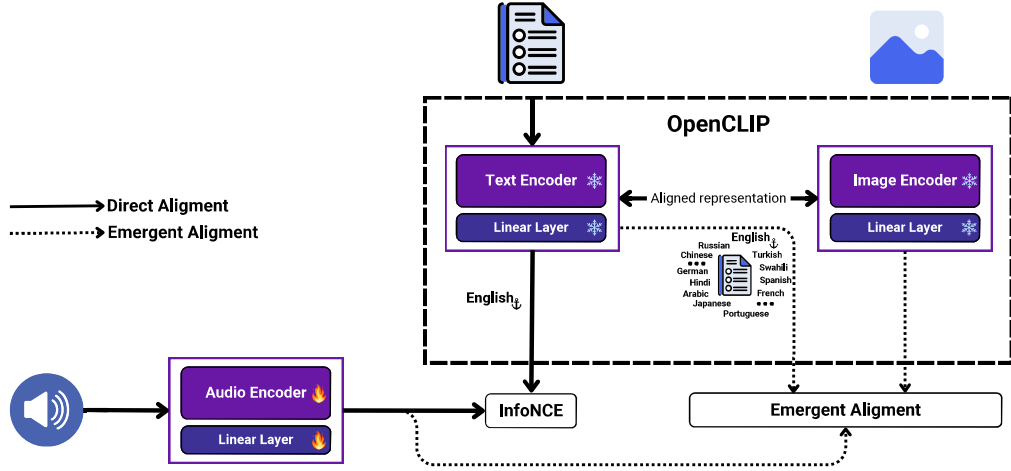


Figure 4.1: Flow of adding a new modality (Audio) to the text-image bimodal model. Linguistic expansion and alignment of these languages for the new modality. During audio model training, alignment is performed only with the anchor encoder (textual), and at this stage, the model does not process any image information. In addition to the 12 languages highlighted for translation and evaluation, the model supports 88 additional languages.

using the InfoNCE loss [189], as in CLIP [154] and CLAP [49]. Given a mini-batch of N paired samples $\{(a_i, t_i)\}_{i=1}^N$, where a_i denotes an audio sample and t_i denotes its corresponding textual description, the audio encoder $f_a(\cdot)$ and the text encoder $f_t(\cdot)$ produce embeddings that are projected into a shared representation space. The pair (a_i, t_i) is treated as a positive pair, while all other combinations (a_i, t_j) with $j \neq i$ are treated as negatives within the same batch.

Let $\mathbf{z}_i^a$ and $\mathbf{z}_i^t$ be the normalized audio and text embeddings for the i -th pair. The similarity between an audio embedding and a text embedding is computed as:

$$s_{ij} = \frac{\text{sim}(\mathbf{z}_i^a, \mathbf{z}_j^t)}{\tau}, \quad (4-1)$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity and τ is a temperature parameter. The audio-to-text contrastive loss is defined as:

$$\mathcal{L}_{a \rightarrow t} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(s_{ii})}{\sum_{j=1}^N \exp(s_{ij})}. \quad (4-2)$$

Similarly, the text-to-audio loss is computed by reversing the retrieval direction:

$$\mathcal{L}_{t \rightarrow a} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(s_{ii})}{\sum_{j=1}^N \exp(s_{ji})}. \quad (4-3)$$

The final training objective is the symmetric contrastive loss:

$$\mathcal{L}_{\text{contrastive}} = \frac{1}{2} (\mathcal{L}_{a \rightarrow t} + \mathcal{L}_{t \rightarrow a}). \quad (4-4)$$

This objective encourages each audio embedding to be close to its paired text embedding and distant from the other text embeddings in the same mini-batch. At the same time, it encourages each text embedding to retrieve its corresponding audio sample. Therefore, the same shared space can be used for both audio-to-text and text-to-audio retrieval.

Beyond that, a key aspect of CACARA’s training procedure is that, unlike LiT (Locked Image Text Tuning), which adapts a text encoder for downstream tasks while keeping only the image encoder fixed, CACARA maintains the two pre-trained encoders image-text fixed and aligns the new audio encoder to this existing shared latent space.

4.1.3 Multimodal and Multilingual Emergent Alignment

CACARA training leverages the emergent learning capacity of pre-trained models. Since the image and text encoders are pre-aligned in a shared feature space due to their pre-training, training the newly incorporated audio model against the text encoder implicitly aligns it with the shared space of all three. This emergent alignment enables the new audio model to describe untrained image modality features without training.

The text modality is the primary anchor for emergent learning within this architecture. As depicted in Figure 4.1, solid lines indicate direct training pairings (audio-text data presented to the model), while dotted lines indicate emergent learning, the emergent relationship between audio and image.

Due to the text encoder’s role as an anchor and the freezing of pre-trained text and image encoders, it remains fixed, synchronizing text with additional modalities. This eliminates the need for explicit multilingual training, as XLM-RoBERTa’s inherent multilingual capabilities extend to additional modalities through emergent alignment. Integrating a new modality synchronizes it with the multilingual features of the textual model.

This strategy enables non-multilingual configurations to inherit linguistic capabilities from the text encoder. As illustrated in Figure 4.1, multiple languages align with other modalities via this emergent synchronization, significantly reducing training costs and time. Unlike conventional approaches that train models on each language separately, CACARA trains only in English, leveraging English’s greater data availability and model quality. This contrasts with existing literature, which often incurs multiplicative computational costs by training on multilingual datasets individually [23, 115].

Although CACARA’s text encoder supports 100 languages, we selected 12 languages for evaluation: English, Portuguese, Spanish, French, German, Chinese, Japanese,

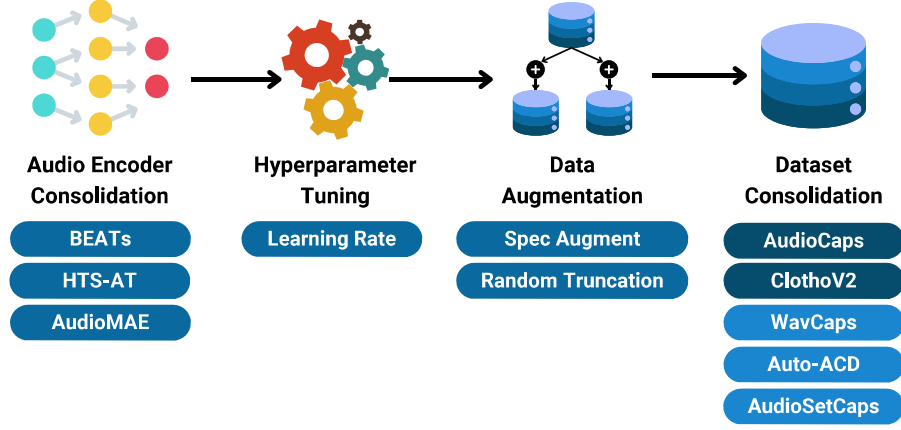


Figure 4.2: *The CACARA model’s training and optimization steps are divided into four main stages: consolidation of the newly added encoder, hyperparameter tuning, data augmentation, and consolidation of the training datasets.*

Russian, Turkish, Hindi, Arabic, and Swahili. Despite an extensive search, we did not identify any publicly available multilingual, human-annotated datasets for bimodal audio-text tasks, nor any multilingual datasets containing more than two modalities, including audio. Consequently, we adopted an evaluation methodology based on automatically translated data. Specifically, in the absence of multilingual test sets, we translated the original English test data into the target languages using Google Translate¹.

4.1.4 Optimization Pipeline

The final CACARA model was developed via a four-stage optimization pipeline (Figure 4.2). The first stage (Audio Encoder Consolidation) involved a comparative evaluation of different audio encoders to select the optimal one for the target tasks. We selected four encoders: BEATs [22], HTS-AT [21], and AudioMAE [89]. These models were chosen based on their SOTA performance in sound event detection and audio tagging, and their relatively recent introduction to the field, as established in the literature.

The second stage (Hyperparameter Tuning) focused on identifying optimal hyperparameters. The learning rate parameter was investigated and found to influence training and model performance substantially. We conducted a systematic search to determine values that ensured a fair comparison across all encoders.

The third stage (Data Augmentation) incorporated data augmentation to improve model robustness. We used two techniques: SpecAugment [142], which masks blocks of

¹<https://translate.google.com>

frequency and time channels, and Random Truncation [48], which divides training data into smaller segments.

The fourth stage (Dataset Consolidation) involved a deliberate selection of datasets, focusing on quantity, diversity, multimodal decoupling, and label quality, to evaluate combinations and identify the subset that yields optimal performance. We chose five datasets: AudioCaps [98], ClothoV2 [45], WavCaps [127], Auto-ACD [183], and AudioSetCaps [10].

4.2 PALMA

In this section, we present PALMA (Pre-trained Audio-Language Multilingual Alignment via Mixture-of-Layers), a bimodal audio-language framework designed for multilingual audio-text representation learning. PALMA (/ˈpaw.mə/), Portuguese for “clapping”, reflects the model’s connection to audio-language alignment and its continuity with CLAP-style [49] contrastive learning. While CACARA investigates how audio can be integrated into an existing multilingual multimodal space, PALMA focuses on the internal design of the audio-language alignment process itself. The framework studies how encoder selection, multilingual supervision, efficient training strategies, and multi-layer representation aggregation affect the quality of audio-text alignment.

4.2.1 Introduction

Audio-language representation learning has become a central direction in multimodal learning due to its ability to connect acoustic signals with natural language descriptions [182]. Contrastive Audio-Language Models, such as CLAP [49], learn a shared embedding space where audio clips and their corresponding textual descriptions are mapped close to one another. This formulation supports tasks such as audio-text retrieval and zero-shot audio classification, since audio and text can be compared directly in the learned representation space [49, 50].

Despite this progress, several limitations remain. Most audio-language models are trained and evaluated primarily in English, which restricts their use in multilingual retrieval scenarios [40, 207]. In addition, existing works often rely on a single configuration of audio and text encoders, making it difficult to understand how different pre-trained components affect alignment quality [202]. Another common limitation is using only the final layer of the audio encoder as the basis for the audio representation. Since intermediate layers may encode complementary acoustic, temporal, and semantic information [51, 114, 145, 146, 188], relying only on the final layer may discard useful information for cross-modal retrieval.

PALMA is proposed to address these limitations by systematically investigating the design of multilingual audio-language models. The framework evaluates different combinations of audio and multilingual text encoders, studies training strategies inspired by vision-language models, and explicitly incorporates multilingual supervision. In addition, PALMA introduces the MoL mechanism, a learnable aggregation strategy that combines representations from multiple layers of the audio encoder. This mechanism aims to create a richer audio representation by integrating information from different levels of abstraction.

The development of PALMA is also connected to the findings and motivation behind CACARA. While CACARA shows that a new audio modality can inherit multilingual capabilities from a frozen multilingual text encoder, PALMA studies the audio-language alignment process in a more controlled bimodal setting. In this sense, PALMA provides a systematic analysis of the architectural and training choices that influence audio-text alignment, while CACARA explores how such alignment can support the expansion of multilingual multimodal spaces. Both frameworks contribute to the broader goal of building efficient, transferable, and linguistically inclusive multimodal representations.

The remainder of this section describes the architecture of PALMA, the evaluated encoder configurations, the adopted training strategies, the MoL mechanism, and the evaluation protocol for measuring multilingual audio-text performance.

4.2.2 Experimental Framework and Model Design

PALMA employs a systematic multi-phase (Figure 4.3) experimental design to identify the most effective combination of audio encoders, text encoders, and training strategies for cross-modal learning. This approach ensures that each component is optimized individually before integration into the complete multimodal framework, thereby maximizing the overall system performance.

The initial phase (Audio Encoder Consolidation) focuses on identifying the optimal audio encoder architecture by comprehensively evaluating SOTA audio representation models. Three prominent audio encoders were evaluated: Bidirectional Encoder Representations from Audio Transformers (BEATs) [22], Hierarchical Token-Semantic Audio Transformer (HTS-AT) [21], and Consistent Ensemble Distillation for Audio Tagging (CED) [39]. Each encoder was evaluated across different learning rate schedules $\{1 \times 10^{-6}, 1 \times 10^{-5}, \text{ and } 1 \times 10^{-4}\}$ and weight decay regularization $\{1 \times 10^{-6}, 1 \times 10^{-4}, \text{ and } 1 \times 10^{-2}\}$ in second and next phase (Hyperparameter Tuning). The most performant and stable audio encoder identified in this phase serves as the backbone for subsequent experiments.

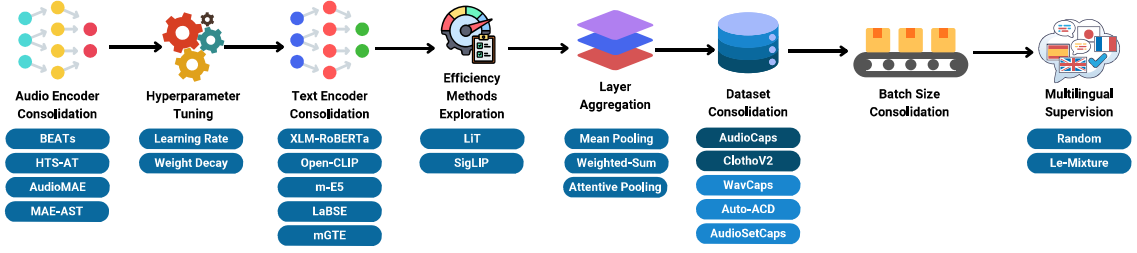


Figure 4.3: Overview of the PALMA optimization pipeline. The development process is divided into eight main stages: audio encoder consolidation, hyperparameter tuning, text encoder consolidation, efficiency method exploration, layer aggregation, dataset consolidation, batch size consolidation, and multilingual supervision.

The next phase of the methodology (Text Encoder Consolidation) investigates the impact of the different text encoders on the overall performance of the bimodal model. This involves a comparative study of different multilingual text encoders, including XLM-RoBERTa – Base [31], XLM-RoBERTa – Base from OpenCLIP², E5 Multilingual [195] Base, LaBSE [56] and mGTE [219] models, to assess their effectiveness in creating a shared semantic space with the audio modality. This phase aims to quantify the trade-offs between model size, computational cost, and performance in multilingual contexts. The evaluation is conducted through a series of experiments, measuring the model’s ability to perform zero-shot cross-lingual for retrieval and classification.

The next phase (Efficiency Methods Exploration) focused on advanced training strategies, specifically investigating the effectiveness of Locked-image Text (LiT) tuning and SigLIP (Sigmoid Loss in Language Image Pre-training) loss functions within the audio-language domain. This phase explored whether techniques proven effective in vision-language models could be successfully adapted for audio-text alignment.

The next phase (Layer Aggregation) introduces and evaluates a key architectural innovation proposed within PALMA: the MoL strategy. This method is designed to create a more comprehensive and nuanced audio representation by challenging the conventional approach of relying solely on the final layer’s output from the encoder. The MoL strategy is founded on the principle that intermediate layers capture a rich spectrum of information, from low-level signal characteristics to high-level semantic abstractions. It implements a learnable, weighted aggregation of feature maps from multiple layers of the audio encoder. This is achieved via a gating network that dynamically calculates the importance of each layer’s contribution for a given input. These weighted features are then fused into a single representation, which is passed through a projection module to align with the text modality. This phase assesses the impact of the MoL strategy on model performance,

²<https://huggingface.co/laion/CLIP-ViT-B-32-mlm-roberta-base-laion5B-s13B-b90k>

aiming to validate its ability to capture a richer, more context-aware understanding of audio. It can be easily applied alongside the LiT strategy for more efficient training.

The next phase (Dataset Consolidation) involved a deliberate selection of datasets, as in CACARA. We chose five datasets: AudioCaps [98], ClothoV2 [45], WavCaps [127], Auto-ACD [183], and AudioSetCaps [10].

The next phase, Batch Size Consolidation, investigated the effect of batch size on model performance. To this end, each configuration in this phase was evaluated using a range of batch sizes: {24, 48, 64, 96, 128, 192, 256, 384, 512}.

The final phase, Multilingual Supervision, focused on training PALMA with expanded resources and multilingual translated data. This phase also benchmarked PALMA against established multilingual baselines, aiming to demonstrate its SOTA capabilities in multilingual audio-text retrieval and classification.

4.3 Evaluation

Both PALMA and CACARA are evaluated through two main types of tasks: cross-modal retrieval and zero-shot audio classification. The retrieval experiments assess whether the model can align audio and text representations in a shared embedding space, while the zero-shot classification experiments evaluate whether the learned audio representations can be compared directly with textual class descriptions without task-specific fine-tuning.

For audio-text retrieval, we use AudioCaps [99] and ClothoV2 [45], two standard benchmarks for evaluating audio-language models. In this setting, the model receives an input from one modality as a query and must retrieve the corresponding item from another modality. Since CACARA aligns audio, text, and image representations in a shared embedding space, retrieval performance is measured by comparing embeddings through cosine similarity.

Let $\mathcal{D} = \{(a_i, t_i)\}_{i=1}^N$ be a dataset with N paired audio-text examples, where a_i is an audio sample and t_i is its corresponding textual description. The audio encoder maps a_i to an embedding $\mathbf{z}_i^a$, while the text encoder maps t_i to an embedding $\mathbf{z}_i^t$. The similarity between an audio sample a_i and a text caption t_j is computed as:

$$s_{ij} = \text{sim}(\mathbf{z}_i^a, \mathbf{z}_j^t), \quad (4-5)$$

where $\text{sim}(\cdot, \cdot)$ denotes the cosine similarity.

Two retrieval directions are considered. In audio-to-text retrieval (A→T), the audio sample is used as the query and the objective is to retrieve its corresponding text caption. For each audio query a_i , all candidate captions $\{t_j\}_{j=1}^N$ are ranked according to

their similarity scores s_{ij} . The rank of the correct caption is denoted by $\text{rank}_{a \rightarrow t}(i)$. In text-to-audio retrieval ($T \rightarrow A$), the textual description is used as the query and the objective is to retrieve the corresponding audio sample. In this case, all candidate audio samples are ranked for each text query t_i , and the rank of the correct audio is denoted by $\text{rank}_{t \rightarrow a}(i)$.

The main metric used for retrieval evaluation is Recall at K ($R@K$), as is conventional in the multimodal retrieval literature. For audio-to-text retrieval, $R@K$ is defined as:

$$R@K_{a \rightarrow t} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}[\text{rank}_{a \rightarrow t}(i) \leq K], \quad (4-6)$$

where $\mathbb{1}[\cdot]$ is the indicator function, which returns 1 when the condition is true and 0 otherwise. Therefore, $R@K$ measures the percentage of queries for which the correct target appears among the top K retrieved candidates.

Similarly, for text-to-audio retrieval, $R@K$ is defined as:

$$R@K_{t \rightarrow a} = \frac{1}{N} \sum_{i=1}^N \mathbb{1}[\text{rank}_{t \rightarrow a}(i) \leq K]. \quad (4-7)$$

We report $R@1$, $R@5$, and $R@10$. $R@1$ measures how often the correct item is ranked first. $R@5$ measures how often the correct item appears among the five most similar candidates, while $R@10$ measures how often it appears among the ten most similar candidates. To summarize performance across different retrieval strictness levels, we also report the average recall:

$$R@Avg = \frac{R@1 + R@5 + R@10}{3}. \quad (4-8)$$

The same definition is applied separately to each retrieval direction, producing $R@Avg_{a \rightarrow t}$ for audio-to-text retrieval and $R@Avg_{t \rightarrow a}$ for text-to-audio retrieval.

For zero-shot audio classification, we evaluate CACARA on ESC-50 [151] and UrbanSound8K [164]. These datasets are used to assess whether the learned audio-text space can support classification without training a task-specific classifier. ESC-50 is organized into 5 folds, which are used for 5-fold cross-validation, while UrbanSound8K is organized into 10 predefined folds, which are used for 10-fold cross-validation. In each case, performance is reported by evaluating the model across the corresponding folds and averaging the results. In this protocol, each class label is converted into a textual prompt and encoded by the text encoder. Given an audio sample, its audio embedding is compared with all textual class embeddings, and the predicted class is the one with the highest similarity score:

$$\hat{y}_i = \arg \max_{c \in C} \text{sim}(\mathbf{z}_i^a, \mathbf{z}_c^t), \quad (4-9)$$

where $\mathcal{C}$ is the set of candidate classes and $\mathbf{z}_c^t$ is the text embedding associated with the prompt of class c .

Classification performance is measured with accuracy. Given a set of N examples with ground-truth labels y_i and predicted labels $\hat{y}_i$, accuracy is defined as:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^N \mathbb{1} [\hat{y}_i = y_i]. \quad (4-10)$$

Accuracy measures the proportion of examples for which the predicted class matches the ground-truth class. In retrieval tasks with a single correct target for each query, R@1 can be interpreted as a top-1 retrieval accuracy, since it measures whether the correct item is retrieved in the first position.

The use of both ESC-50 and UrbanSound8K also allows a cross-dataset analysis of zero-shot behavior. ESC-50 contains a broad set of environmental sound categories, while UrbanSound8K focuses on urban acoustic events.

In the multilingual evaluation, the same retrieval and classification protocols are applied after translating the textual side into different languages. For retrieval, the captions from AudioCaps and ClothoV2 are translated into the 11 target languages mentioned previously, beyond English (Portuguese, Spanish, French, German, Chinese, Japanese, Russian, Turkish, Hindi, Arabic, and Swahili). For zero-shot classification, the class prompts from ESC-50 and UrbanSound8K are translated and encoded by the frozen multilingual text encoder. This evaluation setup allows us to assess whether the audio modality inherits multilingual alignment through the text representation space learned by the frozen multilingual encoder. More details about the resources used in evaluation as well as in training can be seen in Appendix A and more details on the evaluation protocol in Appendix B.

Results

This chapter presents and discusses the experimental results obtained for the two frameworks proposed in this dissertation: CACARA and PALMA. The results are organized according to the main objectives of each model. First, we analyze CACARA in terms of its ability to support emergent multimodal and multilingual alignment, evaluating whether an audio encoder trained only on English audio-text pairs can operate in a broader multilingual, cross-modal retrieval setting. Then, we evaluate PALMA as a bimodal multilingual audio-language model, focusing on the impact of encoder selection, training strategies, and the proposed MoL mechanism.

5.1 Results at a Glance: A TL;DR Overview

Before presenting the detailed experimental results and discussions of CACARA and PALMA, this section offers a concise, high-level synthesis of the principal findings of this dissertation. In the spirit of a “TL;DR” (Too Long; Didn’t Read) overview, it is intended to provide the reader with an interpretive map of the results, highlighting the main conclusions before the individual experiments, ablations, and comparisons are examined in detail. This overview also clarifies how the different experimental stages contribute to the broader investigation of efficient multilingual and multimodal audio-text representation learning. CACARA and PALMA address complementary dimensions of this problem: CACARA investigates whether audio as a modality can be efficiently incorporated into an existing frozen multilingual vision-language representation space, whereas PALMA examines how the internal design and optimization of bimodal audio-language models influence their performance, efficiency, and multilingual generalization.

Emergent alignment enables efficient multimodal and multilingual expansion. The CACARA results demonstrate that a new audio modality can be integrated into a pre-trained multilingual image-text representation space without jointly retraining the original image and text encoders. By aligning the audio encoder only with the frozen textual anchor, the model establishes direct audio-text alignment while also inheriting relationships

already encoded in the shared representation space. This procedure enables two forms of emergent transfer: multilingual audio-text capabilities in languages not explicitly observed during training and emergent audio-image alignment learned without paired audio-image supervision. These findings show that pretrained multimodal spaces can serve as foundations for lower-cost modality expansion. However, the strength of the resulting transfer varies across languages, datasets, tasks, and modality combinations, indicating that emergent alignment does not reproduce every pre-existing relationship with equal fidelity.

The geometry of the multilingual text space constrains emergent transfer. The multilingual evaluation of CACARA reveals that successful cross-lingual transfer depends not only on whether the textual anchor is multilingual, but also on how individual languages are organized within its representation space. This behavior motivates the concept of the Cross-Lingual Gap, which describes the degradation in alignment between English, the language used during explicit audio-text training, and languages introduced only through the frozen textual space. The centroid analysis shows that languages represented closer to English in the text embedding space generally retain stronger audio-text retrieval performance, whereas more distant languages exhibit greater degradation. Although this relationship should not be interpreted as establishing causality, it provides evidence that the geometry and linguistic coverage of the pretrained text space influence how effectively alignment learned in English transfers to other languages. Emergent multilinguality is therefore not a binary capability, but a graded phenomenon conditioned also by the structure of the original multilingual representation space.

Explicit multilingual supervision improves cross-lingual robustness. Although CACARA demonstrates that multilingual audio-text capabilities can emerge from English-only alignment, it also shows that relying solely on a pretrained multilingual text encoder and implicit cross-lingual transfer does not ensure equally robust alignment across languages after audio-text training. The PALMA results therefore demonstrate that explicit multilingual supervision using translated captions can substantially improve multilingual retrieval and zero-shot classification, reducing the performance degradation observed when training is restricted to English captions. Among the investigated multilingual training strategies, balanced and comparatively simpler sampling procedures achieve strong results without requiring more complex alternatives. These findings indicate that emergent multilingual transfer and translated supervision are complementary: the former provides a low-cost mechanism for initial cross-lingual expansion, whereas the latter strengthens and stabilizes the learned multilingual representation space.

Audio-encoder selection is fundamental to successful modality integration. The CACARA encoder-consolidation experiments demonstrate that the choice of the newly introduced audio backbone directly influences how effectively it can be integrated into the frozen image-text representation space. The SOTA acoustic encoders evaluated exhibit distinct performance patterns despite being aligned with the same frozen textual anchor and trained under comparable conditions. This finding suggests that the existence of a pretrained multimodal space alone may not be sufficient to ensure successful modality expansion. Rather, the representational quality of the newly introduced encoder, the objectives and data used during its pretraining, and the compatibility of its learned representation structure with the anchor space may all play important roles in determining the effectiveness of emergent alignment.

Encoder compatibility is more important than isolated encoder strength. The PALMA experiments extend this conclusion by demonstrating that audio-language performance depends not only on the individual capabilities of the audio and text encoders, but also on the compatibility between their representation spaces. Encoders that perform strongly in their original unimodal domains do not necessarily form the strongest audio-text configuration when combined. Instead, retrieval and zero-shot classification performance emerge from the representational synergy between both encoder towers and from how they respond to the adopted optimization strategy. Taken together, the CACARA and PALMA results reinforce the need to select encoders according to their behavior within the intended shared space rather than exclusively according to isolated benchmark performance.

The effect of data augmentation depends on the model and training configuration. Data augmentation yields different outcomes in CACARA and PALMA. In CACARA, the evaluated augmentation methods individually improve retrieval and zero-shot classification performance, while their combination produces the most consistent gains across tasks. These results suggest that perturbing the acoustic input can strengthen the robustness of the audio encoder as it is adapted to a frozen multimodal representation space. In PALMA, however, the exploratory augmentation analysis does not identify a single strategy that consistently surpasses the non-augmented baseline across datasets, retrieval directions, and downstream tasks. Although some augmentation methods provide modest improvements in specific retrieval settings, the configuration without augmentation remains the strongest for zero-shot classification. These findings indicate that the effectiveness of data augmentation is not universal in multimodal supervision, but instead may depend on the model architecture, adaptation strategy, existing regularization mechanisms, and target task.

Training efficiency involves configuration-dependent trade-offs. The PALMA comparison of training strategies shows that no single adaptation procedure is universally optimal across all tasks and encoder combinations. Full fine-tuning generally provides greater flexibility for cross-modal adaptation, whereas locked-encoder strategies preserve pretrained knowledge and reduce the number of trainable parameters. These efficiency gains may be accompanied by changes in retrieval and zero-shot classification performance. Similarly, alternative contrastive objectives provide configuration-dependent benefits rather than consistent improvements across all experimental settings. Learning rate, weight decay, training-data composition, batch size, and multilingual sampling further influence the final performance, showing that efficient audio-language model design requires the joint optimization of architectural, training, and data-related decisions.

CACARA substantially reduces the cost of multimodal expansion. The computational comparison between CACARA and a fully trainable tri-modal baseline confirms that sequential alignment provides practical efficiency benefits beyond merely reducing the number of updated encoder towers. By freezing the original image and text encoders and optimizing only the newly introduced audio branch, CACARA lowers the overall computational cost of training, energy consumption, and associated carbon emissions. These results demonstrate that reusing pretrained multimodal representation spaces can make modality expansion more computationally accessible while preserving competitive performance in retrieval and zero-shot classification. The resulting efficiency gains are therefore not limited to theoretical reductions in the number of trainable parameters, but translate into substantial savings in training time, energy use, and environmental impact.

Additional scaling provides benefits, but does not replace careful optimization. The expanded-resource CACARA experiments examine whether larger batches, longer training, and the combined use of all available datasets can further improve the learned representation space. Increasing the available resources improves the same model relative to its lower-resource counterpart and benefits several multilingual classification results. However, the expanded model does not uniformly surpass the strongest carefully optimized configurations, particularly in retrieval settings. These findings indicate that scaling remains useful, but its effectiveness may depend on the compatibility, quality, and distribution of the training data relative to the target task. Additional computation cannot fully compensate for suboptimal data composition or replace model- and task-specific optimization.

Intermediate encoder layers retain information relevant to audio-language alignment. The Mixture-of-Layers (MoL) experiments show that useful information remains

distributed across the hierarchy of pretrained audio encoders. Although the final layers remain highly important for semantic alignment, early and intermediate representations retain complementary information. Aggregating representations from multiple layers can produce modest improvements over final-layer-only configurations in retrieval and zero-shot classification, although the magnitude of these gains depends on the encoder, aggregation mechanism, and downstream task. More importantly, the learned layer contributions provide evidence that audio-language models, and multimodal models in general, may benefit from representation-aware mechanisms that exploit the internal organization of pretrained encoders rather than relying exclusively on their final outputs.

General interpretation. Taken together, the results indicate that efficient multilingual multimodal learning can be achieved through the reuse and selective adaptation of pretrained representation spaces rather than through complete large-scale retraining. CACARA demonstrates the feasibility of efficient modality expansion through emergent alignment, but also shows that its success depends on audio-encoder selection, augmentation strategy, the geometry of the frozen multilingual space, and the composition of the training data. Its computational analysis confirms substantial reductions in training time, energy use, and environmental impact, while its scaling experiments show that additional resources provide task-dependent rather than universal gains. PALMA complements these findings by showing that robust audio-language learning depends on encoder synergy, carefully selected training strategies, explicit multilingual supervision, and the effective use of hierarchical acoustic representations. At the same time, the experiments establish important boundaries for these conclusions: emergent transfer remains uneven, translated supervision does not fully replace native multilingual data, augmentation does not benefit every model equally, efficient adaptation may introduce performance trade-offs, additional scaling cannot replace careful optimization, and cross-layer aggregation does not improve every configuration uniformly.

The remainder of this chapter develops these findings in detail. Section 5.2 presents the CACARA experiments, including audio-encoder selection, data augmentation, emergent multimodal alignment, multilingual transfer, the Cross-Lingual Gap, computational efficiency, scaling behavior, and qualitative analyses. Section 5.3 then presents the PALMA optimization pipeline, including audio- and text-encoder selection, training strategies, cross-layer aggregation, exploratory data augmentation, data composition, batch-size analysis, and explicit multilingual supervision.

5.2 CACARA

This section presents the empirical results obtained with CACARA. The analysis evaluates whether the proposed alignment strategy enables the integration of audio into a frozen multilingual vision-language representation space. In particular, we examine CACARA’s performance in cross-modal retrieval and multilingual audio-text retrieval to assess both forms of emergent alignment investigated in this dissertation: multimodal alignment between audio and image and multilingual transfer from the frozen text encoder to the newly introduced audio modality.

5.2.1 Experiments and Results

The experimental analysis of CACARA is organized around four main aspects: multimodal alignment, multilingual generalization, computational efficiency, and expanded-resource performance. First, we evaluate the model’s multimodal capabilities, including the encoder consolidation stage, the effect of data augmentation, and the comparison with existing multimodal baselines. These experiments are presented in Sections 5.2.2, 5.2.3, and 5.2.4. Second, we analyze CACARA’s multilingual behavior in Section 5.2.5, focusing on how audio-text alignment learned from English data transfers to other languages through the frozen multilingual text encoder.

The third part of the analysis examines the computational efficiency of CACARA. In Section 5.2.6, we compare the resources required by the proposed alignment strategy with those required to train a fully tri-modal model, highlighting the cost advantages of reusing frozen pretrained components. Finally, Section 5.2.7 investigates the behavior of CACARA under expanded-resource conditions. These experiments evaluate how the model scales with additional computational capacity, allowing us better to understand the relationship between resource usage and performance.

For the results presented in this section, we used the following training datasets: AudioSetCaps (ASC), Auto-ACD (AA), WavCaps (WC), AudioCaps (AC), and ClothoV2 (C). For the CACARA model, various dataset combinations have been evaluated. More details about these datasets can be seen in Appendix A.2.

We also applied a data-filtering strategy in some experiments inspired by CLIPScore [84]. In its original formulation, CLIPScore is a reference-free metric for image caption evaluation that estimates the compatibility between an image and a candidate caption using a pretrained CLIP model. Given an image embedding $\mathbf{v}$ and a caption embedding $\mathbf{c}$, the score is computed from their cosine similarity as $\text{CLIPScore}(\mathbf{c}, \mathbf{v}) = w \cdot \max(\cos(\mathbf{c}, \mathbf{v}), 0)$, where w is a scaling constant. Therefore, CLIPScore measures how semantically compatible the visual content and the textual description are, without relying on reference captions, making it well-suited for automated filtering.

We adapt the same principle to the audio-text domain by using CLAP [49] instead of CLIP. For each audio-text pair, we compute a CLAP-based compatibility score between the audio embedding and the corresponding caption embedding. This score is then used as a filtering criterion to remove low-quality or weakly aligned pairs from the training data. The goal of this filtering step is to improve dataset quality by retaining samples whose audio and textual descriptions are more semantically consistent. Configurations that use this filtering strategy are denoted by $f\ 0.x$, where $0.x$ represents the minimum CLAP-based compatibility score required for an audio-text pair to be kept in the training set. The compatibility score is normalized to the range $[0, 1]$, where lower values indicate weaker semantic alignment between the audio content and its textual description, and higher values indicate stronger alignment. For example, a filtering threshold of $f = 0.1$ removes all audio-text pairs with compatibility scores below 0.1. Similarly, higher filtering thresholds apply stricter quality control by retaining only pairs with stronger audio-text correspondence.

5.2.2 Audio Encoder Consolidation and Hyperparameter Tuning

The first stage of the CACARA optimization pipeline focused on selecting the audio encoder to be integrated into the multilingual multimodal representation space. Since CACARA introduces an audio encoder into a frozen multilingual image-text model, the quality of this audio backbone directly influences the resulting level of alignment with these modalities [50, 202]. Therefore, we evaluated different pretrained audio encoders, including HTS-AT [21], AudioMAE [89], and BEATs [22]. Appendix A.3 summarizes the main characteristics of these encoders in terms of architecture, pretraining strategy, and approximate number of parameters.

HTS-AT was included due to its favorable trade-off between performance and computational efficiency, as well as its good performance when transferred to downstream audio tasks [50, 202, 215]. AudioMAE and BEATs were included as representative SSL audio encoders with strong performance in audio classification and general-purpose audio representation learning [22, 89, 215].

In addition to comparing encoder architectures, we also performed a learning rate ablation in order to tune this hyperparameter of each model. This step is important because the learning rate strongly influences optimization stability, convergence speed, and final model performance, and inadequate learning rate selection can lead to unfair or misleading comparisons between architectures [176, 200]. The learning rate set was defined as: $\{5 \times 10^{-4}, 5 \times 10^{-5}, 5 \times 10^{-6}\}$. This range was chosen based on the learning-rate scales used in the original works on these pretrained audio models and in previous contrastive ALMs [49, 50, 202, 204, 212].

Audio to Text						Text to Audio			
AudioCaps									
Model	LR	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
HTS-AT	5e-4	26.55	59.46	73.52	53.18	5.15	20.11	31.77	19.01
	5e-5	22.61	53.09	68.26	47.99	4.09	16.81	28.08	16.33
	5e-6	11.86	33.99	48.71	31.52	1.96	9.00	15.12	8.69
AudioMAE	5e-4	25.06	57.37	73.21	51.88	5.35	21.78	34.06	20.40
	5e-5	27.56	61.26	74.69	54.50	6.34	23.44	36.20	21.99
	5e-6	20.45	50.10	65.44	45.33	4.41	18.52	28.77	17.23
BEATs	5e-4	27.72	59.71	74.22	53.88	5.42	22.18	33.79	20.46
	5e-5	30.57	63.17	77.55	57.10	6.64	25.67	38.43	23.58
	5e-6	27.51	61.96	77.05	55.51	5.87	23.22	35.59	21.56
ClothoV2									
Model	LR	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
HTS-AT	5e-4	11.06	30.22	42.68	27.98	1.83	7.65	13.09	7.52
	5e-5	11.98	32.30	44.70	29.66	2.00	8.43	14.25	8.22
	5e-6	3.71	12.88	20.03	12.21	0.63	2.52	4.54	2.56
AudioMAE	5e-4	7.68	23.05	33.90	21.54	1.42	5.78	9.91	5.70
	5e-5	9.72	27.96	39.27	25.65	1.77	7.25	12.19	7.07
	5e-6	5.43	16.02	24.50	15.32	0.95	3.88	6.65	3.83
BEATs	5e-4	7.66	22.13	32.45	20.75	1.36	5.73	9.90	5.66
	5e-5	15.64	37.57	50.56	34.59	3.23	12.92	20.29	12.15
	5e-6	7.51	21.83	31.17	20.17	1.34	5.74	9.55	5.54

Table 5.1: Learning rate ablation results for the HTS-AT, AudioMAE, and BEATs audio encoders on Audio-to-Text and Text-to-Audio retrieval tasks, evaluated on AudioCaps and ClothoV2. **Bold** values indicate the best result for each model, dataset, modality, and metric across learning rates. Values in **blue** indicate the best results overall for that metric and dataset.

For these experiments, all models were trained for 100k steps on a combined training set comprising Auto-ACD, WavCaps, AudioCaps, and ClothoV2. Training was performed on a single RTX 8000 GPU with 48 GB. We used the Adam optimizer [100] with $\beta_1 = 0.9$ and $\beta_2 = 0.98$, together with a cosine learning rate decay schedule with 500 warmup steps. Weight decay was fixed to 1×10^{-6} . Additional details about the hyperparameters used in these experiments, as well as in subsequent experiments, are presented in Appendix C.

Since the evaluated encoders have different model capacities, we used the largest batch size that each backbone could support on the same hardware. This choice allows each model to be evaluated under its strongest feasible hardware-constrained configuration. In particular, smaller encoders such as HTS-AT, with approximately 30M param-

ters, have lower representational capacity than larger backbones such as AudioMAE and BEATs, which have more than double that capacity. If all models were constrained to the same batch size, HTS-AT would be penalized by its smaller parameter count and by being unable to exploit its practical advantage of supporting larger batch sizes. Since contrastive learning benefits from larger mini-batches by increasing the number of in-batch negative examples, previous works have shown that batch size can have a direct impact on performance in both audio-language and vision-language contrastive pretraining [48, 216]. Under this setting, HTS-AT was trained with a batch size of 256, BEATs with a batch size of 64, and AudioMAE with a batch size of 96.

The models were evaluated on audio-text retrieval using AudioCaps and ClothoV2, considering both retrieval directions: audio-to-text and text-to-audio. The retrieval results are presented in Table 5.1, while the classification results are reported separately in Table 5.2.

Model	LR	ESC-50	UrbanSound8K
Accuracy (%)			
HTS-AT	5e-4	88.55	69.41
	5e-5	84.40	63.31
	5e-6	55.75	56.62
AudioMAE	5e-4	82.30	65.69
	5e-5	85.60	71.32
	5e-6	62.60	62.46
BEATs	5e-4	84.00	65.45
	5e-5	91.40	73.09
	5e-6	86.10	72.16

Table 5.2: Learning rate ablation results for the HTS-AT, AudioMAE, and BEATs audio encoders on the zero-shot audio classification task, evaluated on ESC-50 and UrbanSound8K. **Bold** values indicate the best mean accuracy for each model and dataset across learning rates. Values in **blue** indicate the best overall mean accuracy.

The retrieval results indeed show that the learning rate affects performance. For most configurations, 5×10^{-5} provided the best trade-off between performance and consistency. In particular, BEATs achieved the best overall performance across both datasets and both retrieval directions at this learning rate. On AudioCaps, BEATs obtained the highest R@Avg for audio-to-text retrieval and text-to-audio retrieval, reaching 57.10 pp and 23.58 pp, respectively. The same pattern was observed on ClothoV2, where BEATs achieved 34.59 of R@Avg for audio-to-text retrieval and 12.15 R@Avg for text-to-audio retrieval. In classification BEATs also shows strong results with 91.40% of mean accuracy on ESC-50 and 73.09% on UrbanSounds. Interestingly, even with a lower capacity HTS-

AT was able to have equivalent, and in some cases best results, than the AudioMAE model, as can be seen on the ClothoV2 and ESC-50 datasets, showing that an efficient model with good representation can still be competitive with higher capacity models in this domain.

The results also indicate that learning rates that are too small can limit the alignment process, as observed with 5×10^{-6} in several configurations. In contrast, a larger learning rate such as 5×10^{-4} can be effective for some encoders (HTS-AT), but does not consistently produce the best results across datasets and models.

Although HTS-AT and AudioMAE obtained overall competitive results in some settings, BEATs showed the most consistent performance across datasets and retrieval directions. Based on these results, BEATs was selected as the main audio encoder for the following CACARA experiments.

5.2.3 Data Augmentation

We conducted a comprehensive ablation study to evaluate the effectiveness of the selected techniques, Random Truncation and SpecAugment, both individually and in combination. The results for retrieval and classification tasks are presented in Tables 5.3 and 5.4, respectively.

For the retrieval tasks detailed in Table 5.3, the combination of both augmentation methods proved most effective. On the AudioCaps dataset, the combined SpecAugment + RT strategy yielded the best performance in audio-to-text retrieval, improving the R@Avg score to 66.28. Interestingly, applying either SpecAugment or RT individually resulted in a slight performance degradation, highlighting a possible synergistic benefit of using both together. For text-to-audio retrieval on AudioCaps and for all retrieval tasks on ClothoV2, every augmentation configuration outperformed the baseline model.

The benefits of data augmentation were even more pronounced for the classification tasks, as shown in Table 5.4. On the ESC-50 dataset, combining SpecAugment and RT resulted in a substantial performance gain, increasing the classification accuracy by 6.75 pp, from 76.15% to 82.90%. A similar trend was observed on the UrbanSounds8K dataset, where the combined approach improved accuracy by 1.55 pp over the baseline.

Given these consistent and significant improvements, we adopted the combined SpecAugment + RT strategy for subsequent experiments.

5.2.4 Multimodal Evaluation

We compare CACARA with established bimodal and multimodal models in order to evaluate two complementary aspects of the proposed framework. First, we evaluate audio-language alignment through audio-to-text retrieval, text-to-audio retrieval, and

Model	Audio to Text				Text to Audio			
	AudioCaps							
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
BEATs + No-Augment	31.09	66.07	79.71	66.07	6.25	24.86	38.11	24.86
BEATs + SpecAug	31.05	65.60	79.55	65.60	6.59	25.53	38.45	25.53
BEATs + RT	30.33	65.15	80.09	65.15	6.21	25.51	38.13	25.51
BEATs + SpecAug + RT	32.01	66.28	79.80	66.28	6.32	25.40	39.46	25.40
ClothoV2								
Model	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
BEATs + No-Augment	10.12	28.98	41.63	28.98	2.51	9.68	15.90	9.68
BEATs + SpecAug	11.25	30.18	41.93	30.18	2.76	10.32	17.03	10.32
BEATs + RT	10.41	29.00	41.17	29.00	2.49	10.30	16.59	10.30
BEATs + SpecAug + RT	11.12	29.89	41.97	29.89	2.51	9.97	15.87	9.97

Table 5.3: Ablation results for the different augmentation strategies tested in the optimization phase, for the Audio to Text and Text to Audio retrieval tasks, evaluated on AudioCaps and ClothoV2. **Bold** values indicate the best result for each dataset, modality, and metric.

Augmentation	ESC-50	UrbanSound8K
Accuracy (%)		
BEATs + No-Augment	76.15	62.31
BEATs + SpecAug	79.05	62.76
BEATs + RT	80.95	60.49
BEATs + SpecAug + RT	82.90	63.86

Table 5.4: Ablation results for the different augmentation strategies tested in the optimization phase, for the zero-shot audio classification task, evaluated on ESC-50 and UrbanSound8K. **Bold** values indicate the best result for each dataset.

zero-shot audio classification. These experiments measure how well the newly introduced audio encoder is aligned with the textual representation space. Second, we evaluate emergent audio-image alignment through audio-to-image and image-to-audio retrieval. This second setting is more challenging, since CACARA is not trained with explicit audio-image pairs. Instead, audio-image alignment is expected to emerge indirectly through the shared representation space defined by the frozen image-text model.

We selected three representative bimodal models for comparison: CLAP [49], LAION-CLAP [202], and the WavCaps model [128]. These models focus on audio-text alignment and therefore provide a strong reference point for evaluating the audio-language capabilities introduced in CACARA. However, because bimodal models are restricted to two modalities, they cannot be directly compared with multimodal models in

terms of modality coverage or emergent cross-modal behavior. For this reason, the tables highlight the best results specifically among multimodal models.

For the multimodal comparison, we selected ImageBind [65], VAST [23], and LanguageBind [224], which represent strong reference models in multimodal representation learning. A direct comparison with Multilingual Multimodal Pre-training (MLMM) [218], the only identified work in our review that explicitly integrates both multimodality and multilinguality, was not feasible due to the absence of publicly available code and sufficient implementation details. Therefore, our comparison focuses on reproducible models that effectively align multiple modalities.

For this phase, all CACARA configurations were trained for 3 epochs, corresponding to approximately 110k optimization steps, while varying the training dataset combinations, using the BEATs audio encoder and augmentations as described in Sections 5.2.2 and 5.2.3. As described in Section 5.2.2, training was performed on the same hardware constraints and the same optimizer configuration as in the encoder consolidation and hyperparameter tuning stages.

Audio-Language Retrieval and Classification

The first evaluation setting focuses on audio-language alignment. In this setting, CACARA is evaluated on audio-to-text and text-to-audio retrieval using AudioCaps and ClothoV2, and on zero-shot audio classification using ESC-50 and UrbanSounds8K. These experiments assess whether the audio encoder can be successfully aligned with the frozen multilingual text encoder and whether this alignment supports both retrieval and classification tasks.

Based on the results summarized in Table 5.5, Figure 5.1 presents a normalized radar chart representation, enabling visual comparison of model performance ($R@1$, $R@5$, $R@10$, $R@Avg$) across datasets. The models compared are ImageBind, LanguageBind, VAST, and CACARA.

Table 5.5 reports the results for the cross-modal retrieval experiments. On AudioCaps, CACARA, trained on all datasets and with a filtering threshold of 0.2, achieved the highest $R@1$ among the multimodal models, reaching 33.98%. This represents an improvement of 14.23 pp over the strongest multimodal models, while remaining competitive with bimodal audio-language models that are trained specifically for audio-text alignment.

On ClothoV2, the best CACARA configuration was obtained when training with WavCaps, achieving an $R@1$ of 17.26%. In this dataset and in the classification datasets, CACARA obtains performance close to LanguageBind. One possible explanation is that LanguageBind also relies on an emergent alignment strategy in which language acts as the central anchor for connecting modalities. However, LanguageBind uses an English-

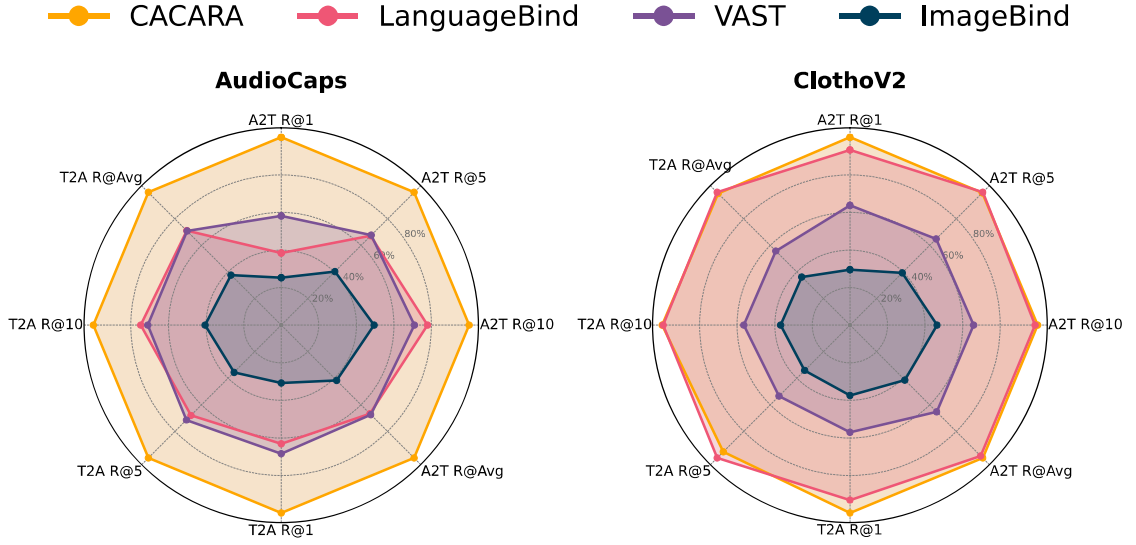


Figure 5.1: Performance comparison of ImageBind, VAST, LanguageBind, and CACARA on Audio-to-Text and Text-to-Audio retrieval using AudioCaps and ClothoV2 datasets. CACARA variants: $CACARA_{ASC/AA/WC/AC/C/f\ 0.2}$ (AudioCaps) and $CACARA_{WC/AC/C/f\ 0.2}$ (ClothoV2). Metrics are normalized R@1, R@5, R@10, and R@Avg, scaled to min-max values.

only text encoder, while CACARA uses a multilingual text encoder and is trained with substantially more data, approximately 10M samples, compared to the 3.5M samples used by CACARA. This means LanguageBind uses about 2.8 times as much training data. Therefore, the competitive performance of CACARA suggests that the proposed architecture is not only effective for cross-modal alignment but also more data-efficient.

Table 5.6 shows the classification results for the ESC-50 and UrbanSounds8K datasets. The performance differences across the strongest models are relatively small, especially on ESC-50. For ESC-50, LanguageBind achieved the highest mean accuracy among multimodal models, reaching 94.75%, followed closely by CACARA trained with WavCaps, AudioCaps, and ClothoV2 with filtering threshold $f\ 0.1$, which achieved 94.37%. The difference between these two models is only 0.38 pp. On UrbanSounds8K, CACARA achieved the best performance among multimodal models, reaching 79.51%, while LanguageBind achieved 79.24%, a difference of only 0.27 pp. These results indicate that CACARA remains competitive in zero-shot classification.

Emergent Audio-Image Alignment

The second evaluation setting investigates emergent audio-image alignment. In CACARA, audio is aligned with text through audio-text supervision, while the frozen image-text model already provides an aligned image-text space. Therefore, successful

Model		Audio-to-Text				Text-to-Audio			
		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
AudioCaps									
BM	CLAP (Microsoft)	15.75	44.7	61.62	40.69	6.32	24.9	38.11	23.11
	CLAP (LAION)	34.58	70.8	83.69	63.02	9.31	35.52	51.68	32.17
	WavCaps	38.70	73.32	86.05	66.02	10.57	38.38	53.21	34.05
MM	ImageBind	8.59	27.58	40.49	25.55	2.25	9.90	16.69	9.61
	VAST	19.75	46.30	57.98	41.34	4.99	19.89	29.25	18.04
	LanguageBind	13.05	45.96	63.76	40.92	4.61	18.94	30.80	18.12
CACARA	CACARA _{AA} /WC/AC/C	31.03 ± 0.51	64.57 ± 0.74	79.34 ± 0.53	58.31 ± 0.43	6.76 ± 0.45	25.52 ± 1.65	38.61 ± 1.30	23.63 ± 1.13
	CACARA _{WC} /AC/C	30.08 ± 0.31	64.95 ± 0.42	78.71 ± 0.75	57.91 ± 0.41	7.57 ± 0.16	28.02 ± 0.23	42.43 ± 0.14	26.00 ± 0.09
	CACARA _{WC} /AC/C/f 0.1	30.96 ± 0.05	64.78 ± 0.91	78.54 ± 0.35	58.10 ± 0.41	7.37 ± 0.20	27.88 ± 0.26	41.72 ± 0.61	25.66 ± 0.35
	CACARA _{WC} /AC/C/f 0.2	31.03 ± 0.60	65.32 ± 0.68	78.80 ± 1.10	58.38 ± 0.77	7.41 ± 0.09	28.40	42.68 ± 0.27	26.17 ± 0.31
	CACARA _{ASC} /AA/WC/AC/C	33.27 ± 0.33	67.46 ± 0.33	81.51 ± 0.30	60.75 ± 0.29	6.91 ± 0.15	26.18 ± 0.18	39.55 ± 0.15	24.21 ± 0.06
	CACARA _{ASC} /AA/WC/AC/C/f 0.1	33.64 ± 0.24	68.40 ± 0.24	81.84 ± 0.17	61.29 ± 0.23	7.34 ± 0.10	27.69 ± 0.32	41.27 ± 0.29	25.43 ± 0.05
	CACARA _{ASC} /AA/WC/AC/C/f 0.2	33.98 ± 0.64	68.30 ± 0.64	81.81 ± 0.21	61.36 ± 0.26	7.30 ± 0.15	27.87 ± 0.32	41.21 ± 0.29	25.46 ± 0.16
ClothoV2									
BM	CLAP (Microsoft)	15.46	38.74	51.52	35.24	4.61	16.90	26.43	15.98
	CLAP (LAION)	14.64	37.28	49.68	33.87	3.77	15.27	23.79	14.28
	WavCaps	18.78	45.15	57.72	40.55	4.38	18.76	28.61	17.25
MM	ImageBind	5.11	16.17	24.96	15.41	1.51	5.57	9.15	5.41
	VAST	11.02	26.64	35.48	24.38	2.30	8.73	13.99	8.34
	LanguageBind	16.11	41.07	53.05	36.74	3.75	16.38	24.65	14.93
CACARA	CACARA _{AA} /WC/AC/C	14.42 ± 0.88	37.21 ± 1.74	49.95 ± 1.60	33.86 ± 1.40	2.90 ± 0.32	12.17 ± 1.49	19.40 ± 1.75	11.49 ± 1.18
	CACARA _{WC} /AC/C	14.39 ± 0.19	35.87 ± 0.21	48.08 ± 0.32	32.78 ± 0.14	3.88 ± 0.13	14.69 ± 0.41	22.49 ± 0.12	13.69 ± 0.18
	CACARA _{WC} /AC/C/f 0.1	15.38 ± 0.49	37.40 ± 1.62	49.63 ± 2.13	34.13 ± 1.40	3.92 ± 0.35	14.99 ± 0.41	23.10 ± 0.54	14.00 ± 0.42
	CACARA _{WC} /AC/C/f 0.2	17.26 ± 2.33	<u>40.91</u> ± 5.84	53.85 ± 6.74	37.34 ± 4.96	4.03 ± 0.36	15.60 ± 1.25	24.78 ± 2.36	<u>14.80</u> ± 1.32
	CACARA _{ASC} /AA/WC/AC/C	11.40 ± 2.17	29.82 ± 6.17	41.12 ± 7.79	27.45 ± 5.37	2.22 ± 0.61	8.71 ± 2.41	14.42 ± 3.66	8.45 ± 2.22
	CACARA _{ASC} /AA/WC/AC/C/f 0.1	13.04 ± 3.12	33.53 ± 6.95	46.00 ± 8.70	30.86 ± 6.25	2.56 ± 0.87	9.84 ± 2.98	16.15 ± 4.43	9.52 ± 2.76
	CACARA _{ASC} /AA/WC/AC/C/f 0.2	13.21 ± 3.16	33.63 ± 6.91	45.67 ± 8.51	30.84 ± 6.19	2.59 ± 0.74	10.07 ± 2.87	16.62 ± 4.68	9.76 ± 2.76

Table 5.5: *Multimodal results, divided into two datasets: AudioCaps and ClothoV2, and two retrieval tasks: audio-to-text and text-to-audio. The evaluated models are categorized as BM (Bimodal Models), MM (Multimodal Models), and CACARA (CACARA Multimodal Models). **Bolded** results indicate the best performance among multimodal models, while underlined results represent the second-best values for the same dataset.*

audio-image retrieval would indicate that the audio modality can be connected to the image modality through the language anchor, even without explicit audio-image training pairs, achieving emergent alignment.

For this evaluation, we use the VGG-Sound Source (VGG-SS) benchmark [19]. VGG-SS is suitable for evaluating audio-image retrieval, as it was constructed to include visible sound sources and includes human verification to ensure that the sound corresponds to objects present in the visual frame.

The results are presented in Table 5.7. We also include a random baseline computed on the same valid VGG-SS test pool used in the retrieval experiments. Since each query has a single correct target, the expected random Recall@K is K/N , where N is the number of valid retrieval items. This baseline provides a lower-bound reference for interpreting the audio-image retrieval scores.

In this setting, ImageBind obtains the strongest performance, reaching 33.69 R@Avg for audio-to-image retrieval and 29.71 R@Avg for image-to-audio retrieval. This result is expected, since ImageBind was explicitly designed to bind multiple modalities to

		Accuracy	
		ESC-50	UrbanSounds8K
BM	CLAP (Microsoft)	93.85	82.74
	CLAP (LAION)	83.10	80.91
	WavCaps	94.25	82.28
MM	ImageBind	64.15	48.20
	VAST	76.80	68.00
	LanguageBind	94.75	<u>79.24</u>
CACARA	CACARA _{AA/WC/AC/C}	89.95 $\pm$ 2.14	77.04 $\pm$ 2.98
	CACARA _{WC/AC/C}	94.15 $\pm$ 0.10	77.40 $\pm$ 1.07
	CACARA _{WC/AC/C/f 0.1}	<u>94.37</u> $\pm$ 0.49	79.51 $\pm$ 1.06
	CACARA _{WC/AC/C/f 0.2}	94.00 $\pm$ 1.39	77.83 $\pm$ 0.11
	CACARA _{ASC/AA/WC/AC/C}	82.45 $\pm$ 0.70	70.04 $\pm$ 1.48
	CACARA _{ASC/AA/WC/AC/C/f 0.1}	89.97 $\pm$ 0.38	75.40 $\pm$ 1.19
	CACARA _{ASC/AA/WC/AC/C/f 0.2}	91.35 $\pm$ 0.69	74.99 $\pm$ 0.55

Table 5.6: Multimodal results, evaluated on ESC-50 and UrbanSounds8K datasets for the classification task. The evaluated models are categorized as BM (Bimodal Models), MM (Multimodal Models), and CACARA (CACARA Multimodal Models). **Bolded** results indicate the best performance among multimodal models, while underlined results represent the second-best values for the same set.

Model	Audio-to-Image				Image-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
Random Baseline	0.02	0.10	0.20	0.10	0.02	0.10	0.20	0.10
ImageBind	14.36	36.82	49.87	33.69	13.25	32.18	43.72	29.71
VAST	1.72	7.72	12.84	7.43	2.72	8.33	12.97	8.01
LanguageBind	<u>1.98</u>	<u>8.92</u>	<u>15.13</u>	<u>8.67</u>	<u>3.04</u>	<u>10.93</u>	<u>18.07</u>	<u>10.68</u>
CACARA _{WC/AC/C/f 0.2}	1.33 $\pm$ 0.14	5.53 $\pm$ 0.44	9.29 $\pm$ 0.64	5.38 $\pm$ 0.37	1.01 $\pm$ 0.40	3.96 $\pm$ 0.73	6.99 $\pm$ 1.25	3.99 $\pm$ 0.79
CACARA _{ASC/AA/WC/AC/C/f 0.2}	1.30 $\pm$ 0.08	5.14 $\pm$ 0.52	8.64 $\pm$ 0.72	5.03 $\pm$ 0.44	1.33 $\pm$ 0.10	5.01 $\pm$ 0.05	8.59 $\pm$ 0.38	4.98 $\pm$ 0.15

Table 5.7: Audio-image retrieval results on VGG-SS for multimodal models and CACARA. Results are reported for audio-to-image and image-to-audio retrieval using R@1, R@5, R@10, and R@Avg. The random baseline is computed on the same valid VGG-SS test pool, assuming one positive target per query. CACARA results are reported as mean and standard deviation over three runs. **Bolded** results indicate the best performance among multimodal models, while underlined results indicate the second-best values.

an image-centered representation space, making it particularly effective for image-related retrieval tasks. LanguageBind obtains the second-best performance, with 8.67 R@Avg for audio-to-image and 10.68 R@Avg for image-to-audio retrieval, followed by VAST.

CACARA obtains lower performance than ImageBind, LanguageBind, and VAST in the audio-image setting. However, these results should be interpreted in light of CACARA’s training strategy. Unlike the baseline multimodal models that explicitly tune their vision towers, CACARA aligns audio with a frozen multilingual image-text model

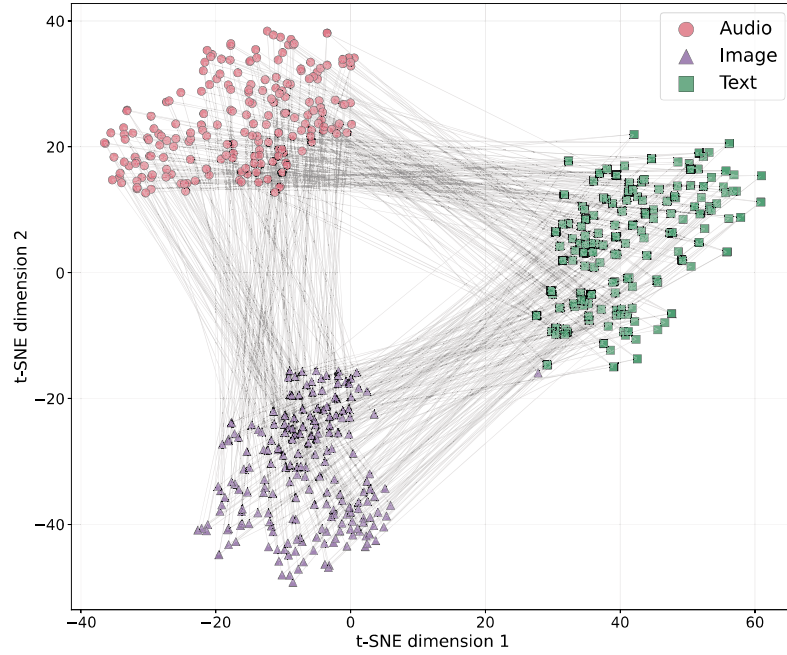


Figure 5.2: *t-SNE visualization of CACARA embeddings on VGG-SS, showing audio, middle-frame image, and text representations in the shared multimodal space. Each point corresponds to one sample, and gray lines connect embeddings belonging to the same underlying instance across modalities.*

using only audio-text supervision. Thus, any audio-image retrieval capability emerges indirectly from the alignment between audio and text, and from the pre-existing alignment between text and image in the frozen model. The fact that CACARA clearly outperforms the random baseline in both directions indicates that the proposed alignment strategy transfers part of the audio-text alignment to the image space in a nontrivial manner.

Among the CACARA variants, $\text{CACARA}_{WC/AC/C/f\ 0.2}$ achieves the highest audio-to-image R@Avg, with 5.38, while $\text{CACARA}_{ASC/AA/WC/AC/C/f\ 0.2}$ achieves the highest image-to-audio R@Avg, with 4.98. Overall, these results reinforce that audio-image alignment in CACARA is emergent and more challenging than audio-text alignment, but still measurable despite the absence of direct audio-image supervision.

To complement the quantitative retrieval results, we also qualitatively analyze the learned representation space using t-SNE [191] visualizations of audio, image, and text embeddings. For this visualization, we randomly sampled 200 examples from the VGG-SS dataset. For each sample, we extracted the audio embedding, the middle-frame image embedding, and the text embedding produced from the template “This is the sound of [class label].” as in CLAP [49]. The t-SNE projection was computed with two output dimensions, a perplexity of 30, a PCA initialization, and 1000 optimization iterations.

Figure 5.2 provides a qualitative visualization of the shared representation space

learned by CACARA on VGG-SS. The figure shows a t-SNE projection of audio, middle-frame image, and text embeddings, where each point corresponds to one sample and gray lines connect representations associated with the same underlying instance. Although the three modalities still form distinct clusters due to the modality gap, they remain aligned in the same latent space.

5.2.5 Multilingual Evaluation

Beyond its multimodal capabilities, CACARA is inherently multilingual, supporting approximately 100 languages without explicit training to align new languages with audio data. We selected twelve languages to evaluate this emergent alignment and analyzed performance in audio-text retrieval and classification tasks. Given the many results across multiple languages, we selected two versions of CACARA for each task to enable a focused comparison.

CACARA _{ASC/AA/WC/AC/C/f 0.2} in Audiocaps								
Language	Audio to Text				Text to Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	33.98 ± 0.64	68.30 ± 0.43	81.81 ± 0.21	61.36 ± 0.27	7.30 ± 0.15	27.87 ± 0.32	41.21 ± 0.29	25.46 ± 0.16
Portuguese	21.62 ± 0.70	51.27 ± 1.31	66.34 ± 1.23	46.41 ± 1.00	5.74 ± 0.18	21.46 ± 0.18	33.47 ± 0.35	20.22 ± 0.11
Spanish	<u>24.06</u> ± 0.48	53.94 ± 0.87	68.05 ± 0.35	48.68 ± 0.51	5.85 ± 0.16	22.58 ± 0.29	34.31 ± 0.25	20.91 ± 0.08
French	21.87 ± 0.70	51.20 ± 1.51	66.31 ± 1.29	46.46 ± 1.17	5.94 ± 0.05	22.97 ± 0.27	35.19 ± 0.47	21.37 ± 0.16
Russian	20.73 ± 0.47	48.96 ± 0.60	63.03 ± 0.74	44.24 ± 0.58	5.55 ± 0.31	19.48 ± 0.47	30.70 ± 0.39	18.58 ± 0.39
Arabic	15.25 ± 0.73	39.70 ± 1.05	53.90 ± 1.15	36.28 ± 0.97	4.71 ± 0.15	18.45 ± 0.23	28.27 ± 0.31	17.15 ± 0.10
Hindi	14.08 ± 0.41	37.47 ± 0.30	51.51 ± 0.80	34.35 ± 0.42	3.71 ± 0.08	14.97 ± 0.09	23.56 ± 0.07	14.08 ± 0.02
German	23.95 ± 0.41	<u>54.36</u> ± 0.06	<u>68.26</u> ± 0.14	<u>48.86</u> ± 0.12	<u>6.00</u> ± 0.09	<u>23.09</u> ± 0.37	<u>35.04</u> ± 0.45	<u>21.38</u> ± 0.25
Chinese(zh)	18.24 ± 0.28	46.59 ± 1.11	62.04 ± 0.42	42.29 ± 0.47	5.45 ± 0.12	20.52 ± 0.20	31.70 ± 0.23	19.23 ± 0.09
Swahili	1.11 ± 0.23	3.96 ± 0.11	6.48 ± 0.24	3.85 ± 0.19	0.65 ± 0.07	2.29 ± 0.06	3.67 ± 0.08	2.20 ± 0.02
Japanese	21.36 ± 0.49	51.46 ± 0.24	65.64 ± 0.38	46.15 ± 0.22	5.59 ± 0.24	21.87 ± 0.48	33.82 ± 0.20	20.42 ± 0.30
Turkish	17.04 ± 0.40	42.65 ± 0.63	56.80 ± 0.45	38.84 ± 0.47	4.42 ± 0.23	17.22 ± 0.32	27.15 ± 0.51	16.27 ± 0.33
CACARA _{WC/AC/C/f 0.2} in ClothoV2								
Language	Audio to Text				Text to Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	17.26 ± 2.33	40.91 ± 5.84	53.85 ± 6.74	37.34 ± 4.96	4.03 ± 0.36	15.60 ± 1.25	24.78 ± 2.36	14.80 ± 1.32
Portuguese	10.83 ± 0.60	28.98 ± 1.56	39.72 ± 1.93	26.51 ± 1.35	3.09 ± 0.16	11.61 ± 0.09	18.57 ± 0.12	11.09 ± 0.05
Spanish	11.37 ± 0.39	29.88 ± 1.73	40.54 ± 2.09	27.26 ± 1.34	3.18 ± 0.12	12.09 ± 0.17	19.38 ± 0.05	11.55 ± 0.05
French	10.83 ± 0.66	28.75 ± 1.43	39.55 ± 1.35	26.38 ± 1.10	3.00 ± 0.14	11.50 ± 0.23	18.05 ± 0.12	10.85 ± 0.12
Russian	8.94 ± 0.51	24.45 ± 1.88	34.62 ± 2.55	22.67 ± 1.64	2.78 ± 0.14	10.60 ± 0.33	16.82 ± 0.51	10.06 ± 0.24
Arabic	6.84 ± 0.61	20.34 ± 1.22	29.13 ± 1.31	18.77 ± 1.02	2.34 ± 0.08	8.85 ± 0.04	14.50 ± 0.08	8.56 ± 0.05
Hindi	6.02 ± 0.43	17.30 ± 1.24	25.69 ± 1.60	16.34 ± 1.05	2.04 ± 0.16	7.80 ± 0.30	12.33 ± 0.54	7.39 ± 0.31
German	<u>11.55</u> ± 0.72	<u>29.99</u> ± 1.50	<u>40.96</u> ± 1.49	<u>27.49</u> ± 1.21	<u>3.21</u> ± 0.15	<u>12.45</u> ± 0.01	<u>19.55</u> ± 0.27	<u>11.74</u> ± 0.13
Chinese(zh)	9.55 ± 0.38	25.94 ± 1.73	36.59 ± 1.80	24.03 ± 1.29	2.79 ± 0.17	11.02 ± 0.15	17.64 ± 0.43	10.48 ± 0.19
Swahili	1.03 ± 0.03	3.35 ± 0.16	5.39 ± 0.27	3.26 ± 0.14	0.52 ± 0.05	1.70 ± 0.13	2.65 ± 0.20	1.62 ± 0.09
Japanese	10.75 ± 0.73	29.19 ± 1.89	40.24 ± 2.10	26.72 ± 1.55	2.97 ± 0.19	11.50 ± 0.37	18.32 ± 0.34	10.93 ± 0.22
Turkish	7.93 ± 0.74	22.53 ± 1.23	32.12 ± 2.69	20.86 ± 1.54	2.51 ± 0.07	9.61 ± 0.20	15.51 ± 0.37	9.21 ± 0.17

Table 5.8: Recall value for the Audio-to-Text and Text-to-Audio retrieval tasks on two different CACARA models (CACARA_{ASC/AA/WC/AC/C/f 0.2} for the AudioCaps dataset and CACARA_{WC/AC/C/f 0.2} for the ClothoV2 dataset) across the twelve evaluated languages. **Bolded** results indicate the best performance among language models, while underlined results represent the second-best values for the same dataset.

For the retrieval task, Table 5.8 and Figure 5.3 present results for the models $CACARA_{ASC/AA/WC/AC/C/f\ 0.2}$ in the Audiocaps dataset and $CACARA_{WC/AC/C/f\ 0.2}$ in the ClothoV2 dataset. The audio encoder was aligned only in English, which leads to better performance in this language. Therefore, we use English results as the upper bound. The results vary depending on the target language, as factors such as translation quality, linguistic complexity, and modality-specific nuances can significantly influence retrieval performance.

Some languages, like Spanish, German, Portuguese, French, Russian, and Japanese, perform well, achieving R@1 above 20 for the $CACARA_{ASC/AA/WC/AC/C/f\ 0.2}$ model on AudioCaps. On average, the other languages achieve R@1 above 13, with performance variations primarily driven by the quantity and quality of the textual data used during pre-training of the text model [61]. This suggests that improving performance in low-resource languages does not require retraining the entire model. Only Swahili was below average because it has very few available resources. We observed the same behavior from the $CACARA_{WC/AC/C/f\ 0.2}$ model in ClothoV2.

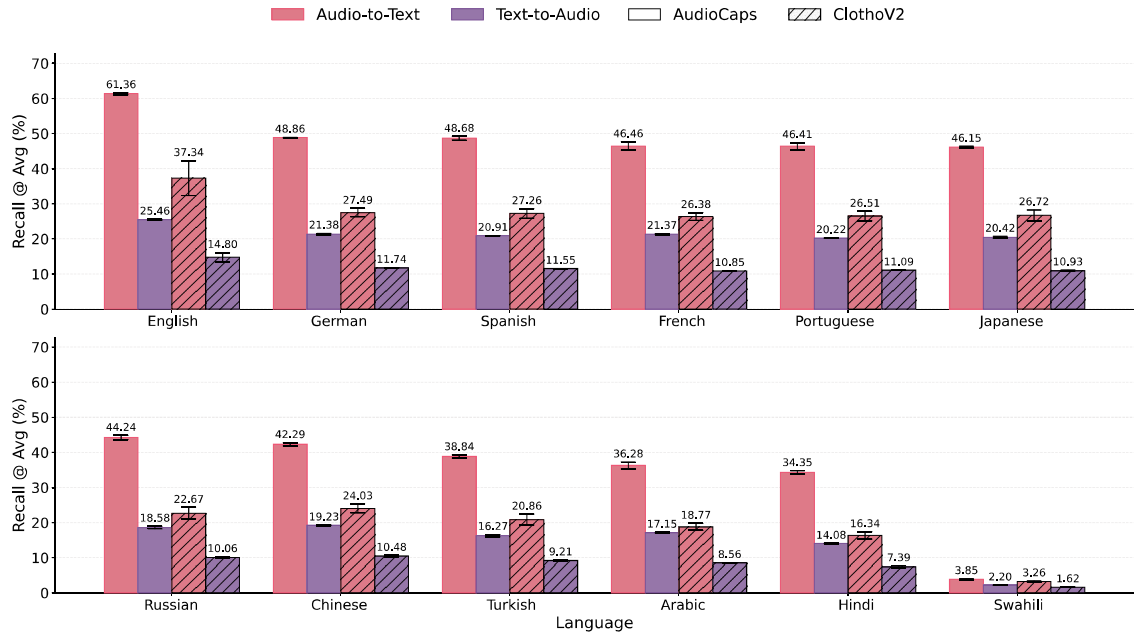


Figure 5.3: Multilingual audio-text retrieval performance of CACARA on AudioCaps and ClothoV2. Results are reported as Recall@Avg across 12 languages for both audio-to-text and text-to-audio retrieval. Colors indicate the retrieval direction, while hatching distinguishes the evaluated datasets.

For the classification task, Table 5.9 shows results for two models, $CACARA_{WC/AC/C/f\ 0.1}$ and $CACARA_{ASC/AA/WC/AC/C/f\ 0.2}$, evaluated on ESC-50 and UrbanSounds8K. As expected, languages with more resources showed strong classification performance, similar to retrieval results. In addition, Mandarin showed good results

compared to the previous task, while Portuguese showed a drop in results. However, the overall classification average of the different languages is 66.5% for CACARA_{WC/AC/C} using ESC-50.

Language	CACARA _{WC/AC/C/f 0.1}		CACARA _{ASC/AA/WC/AC/C/f 0.2}	
	ESC-50	UrbanSounds8K	ESC-50	UrbanSounds8K
English	94.37 ± 0.49	79.51 ± 1.06	91.35 ± 0.69	74.99 ± 0.55
Portuguese	79.63 ± 0.81	66.49 ± 0.79	80.92 ± 0.98	71.38 ± 1.35
Spanish	<u>86.25</u> ± 0.40	72.02 ± 1.78	<u>85.60</u> ± 0.78	71.02 ± 0.67
French	83.60 ± 0.69	69.56 ± 0.65	82.45 ± 0.49	67.72 ± 1.12
Russian	81.15 ± 0.33	71.77 ± 1.00	78.30 ± 1.08	67.16 ± 1.32
Arabic	65.28 ± 0.95	63.13 ± 0.64	63.65 ± 1.09	63.77 ± 0.60
Hindi	64.30 ± 0.97	58.19 ± 0.84	60.12 ± 0.83	58.11 ± 1.15
German	82.12 ± 0.87	<u>74.99</u> ± 0.12	75.92 ± 1.08	72.38 ± 0.10
Chinese(zh)	83.47 ± 1.68	70.14 ± 0.12	81.23 ± 1.07	67.22 ± 1.15
Swahili	20.55 ± 0.41	42.99 ± 1.08	20.48 ± 2.17	38.35 ± 1.97
Japanese	84.08 ± 1.63	68.79 ± 0.02	81.78 ± 0.75	<u>73.03</u> ± 0.74
Turkish	74.12 ± 1.83	64.94 ± 0.68	69.85 ± 1.97	63.91 ± 0.82

Table 5.9: Classification results on two different CACARA models (CACARA_{WC/AC/C/f 0.1} and CACARA_{ASC/AA/WC/AC/C/f 0.2}) in the datasets ESC-50 and UrbanSounds8K across the twelve evaluated languages. **Bolded** results indicate the best performance among language models, while underlined results represent the second-best values for the same dataset.

Cross-Lingual Gap

The multilingual results indicate that CACARA can transfer audio-text alignment from English to other languages, even though the audio encoder is aligned only with English audio-text pairs. However, this transfer does not occur with the same strength for all languages. To investigate this behavior, we examine whether retrieval performance correlates with the geometric position of each language in the frozen multilingual text embedding space.

This analysis is motivated by English’s role as the primary alignment anchor in CACARA. During training, the audio encoder is directly aligned with English captions, while multilingual generalization is inherited from the frozen text encoder. Therefore, we hypothesize that transfer to a target language depends, at least in part, on how close that language is to English in the text representation space. If semantically equivalent captions in another language are mapped near their English counterparts, the audio-text alignment learned from English data is expected to transfer more effectively. In contrast, if the target language occupies a more distant region of the embedding space, the transfer may become weaker.

This behavior could be interpreted as a cross-lingual gap in the representation space. The idea is analogous to the modality gap observed in multimodal models, where embeddings from different modalities may remain partially separated even after being projected into a shared space. In this case, however, the separation occurs between languages rather than between modalities. Thus, the multilingual text encoder may provide a shared representation space, but this space may still contain language-dependent structure that affects downstream cross-lingual transfer.

Therefore, we used the test set of the AudioCaps dataset, and extracted the latent language representations from the original English and translated captions with the pretrained $\text{CACARA}_{ASC/AA/WC/AC/C/f\ 0.2}$ model and measured the euclidian distance between the centroids of each language, to explore further this hypothesis.

To investigate this hypothesis, we used the complete AudioCaps test set, comprising 975 samples, along with its translated versions. For each language, we extracted text embeddings from the captions using the pretrained $\text{CACARA}_{ASC/AA/WC/AC/C/f\ 0.2}$ model. The embeddings were then averaged to compute one centroid per language. Finally, we measured the cosine distance between each language centroid and the English centroid and compared it with the corresponding multilingual audio-to-text retrieval performance. This analysis allows us to examine whether languages that are farther from English in the frozen text embedding space also tend to exhibit lower retrieval performance.

Figure 5.4 supports this interpretation. Languages such as Spanish, German, French, Portuguese, and Japanese are located relatively close to English in the frozen text embedding space and maintain stronger retrieval performance. In contrast, Swahili is the farthest language from the English centroid and obtains the lowest retrieval score. This pattern suggests that the cross-lingual position of each language relative to English is related to the effectiveness of multilingual audio-text transfer.

The correlation analysis also supports this relationship. We report both Pearson and Spearman correlations because they capture complementary aspects of the association between distance and performance. Pearson correlation measures the strength and direction of a linear relationship between two continuous variables. In this case, it evaluates whether increases in cosine distance to English are linearly associated with decreases in Recall@Avg. Spearman correlation, in turn, measures the strength and direction of a monotonic relationship based on ranks. Therefore, it evaluates whether languages that are farther from English also tend to rank lower in retrieval performance, even if the relationship is not perfectly linear.

The cosine distance to English shows a strong negative Pearson correlation with Recall@Avg ($r = -0.962$, $p = 2.347 \times 10^{-6}$), indicating a strong linear association between greater distance from English and lower retrieval performance. The Spearman

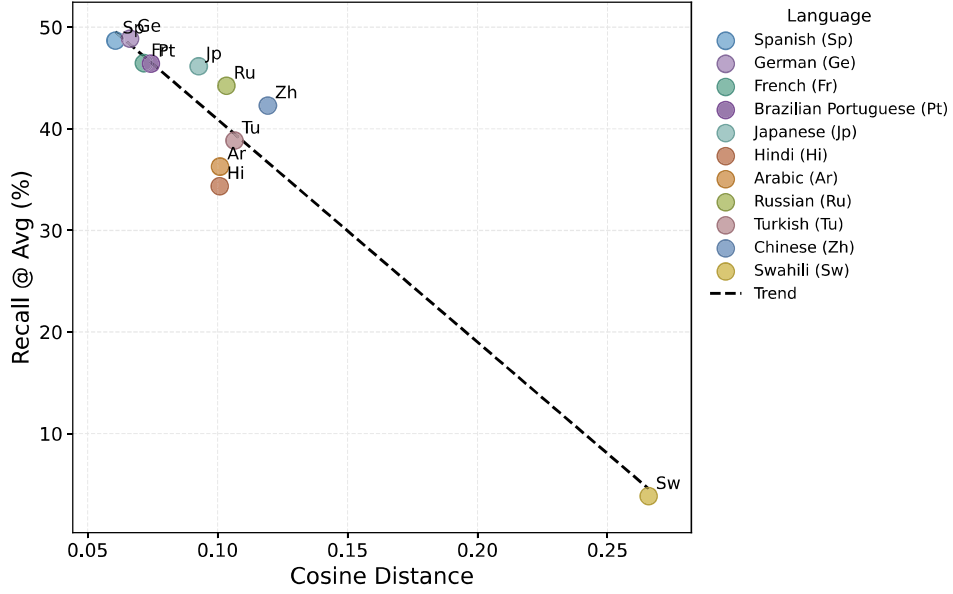


Figure 5.4: Relationship between distance to the English text-embedding centroid and multilingual audio-to-text retrieval performance on AudioCaps. Each point represents one target language. The x-axis shows the cosine distance between the language centroid and the English centroid in the frozen multilingual text embedding space, while the y-axis shows Recall@Avg obtained with $CACARA_{ASC/AA/WC/AC/C/f\ 0.2}$. English is used as the reference centroid and is therefore not shown. The dashed line indicates the fitted linear trend.

correlation is also strongly negative ($\rho = -0.836$, $p = 1.333 \times 10^{-3}$), showing that the same tendency is preserved at the rank level. Both p-values are below the conventional significance threshold of 0.05, which means that the negative relationship observed between distance to English and retrieval performance is unlikely to be due to chance. However, these results should still be interpreted as evidence of association rather than proof of causation, since factors such as translation quality, pretraining data distribution, and language-specific representation quality may also influence performance.

Overall, these results indicate that CACARA inherits not only the multilingual capacity of the frozen text encoder, but also its internal cross-lingual geometry. Multilingual generalization in CACARA can therefore be understood as an emergent transfer process mediated by the text encoder. The audio modality learns to align with English, and the extent to which this alignment transfers to other languages depends on how well those languages are already connected to English in the frozen representation space. In this sense, the cross-lingual gap helps explain why multilingual alignment emerges across languages, albeit with varying degrees of effectiveness.

5.2.6 Efficiency and Scaling

To quantify the computational savings achieved by our approach, we conducted a comparative analysis between the proposed CACARA method and a fully trained tri-modal baseline using a 5k-sample subset from the VAST [23] dataset. The baseline involved concurrently training text, vision, and audio encoders, whereas CACARA froze the text and vision encoders, tuning only the audio encoder. Both methods were trained for 100 epochs across three independent runs to account for variance.

We measured key metrics, including floating point operations (GFLOPs), multiply–accumulate operations (GMACs), parameter count, total training time, energy consumption (kWh), and estimated carbon emissions (kgCO₂eq), presented in Table 5.10 and Figure 5.5.

Model	GFLOPs	GMACS	Parameters	Time (hh:mm)	Energy (kWh)	Emission
CACARA	185,00	92,42	369,40	04:47 ± 00:05	1,91 ± 0,03	0,15 ± 0
Fully Tri-Modal	246,12	122,95	457,25	25:12 ± 2:11	6,66 ± 0,43	0,51 ± 0,03
Reduction (%)	24	24	19	81	71	70

Table 5.10: *Computational efficiency and environmental impact comparison between the proposed CACARA model, which tunes only the audio encoder, and a fully tri-modal baseline. Results are averaged over three runs on a 5k-sample VAST dataset subset.*

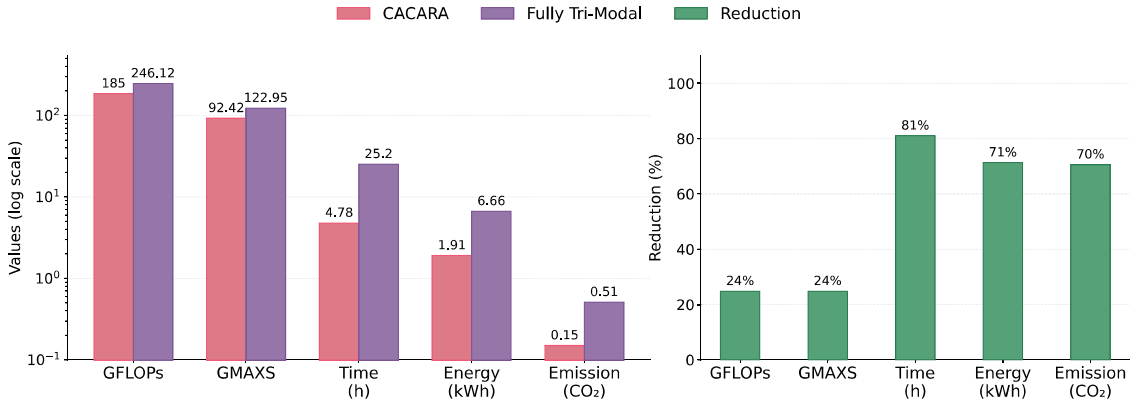


Figure 5.5: *Computational efficiency and environmental impact comparison between the proposed CACARA model, which tunes only the audio encoder, and a fully tri-modal baseline. Results are averaged over three runs on a 5k-sample VAST dataset subset.*

As shown, CACARA reduces computational requirements (GFLOPs and GMACs) by 24% and the parameter count by 19%. This results in a substantial 81% reduction in total training time, accompanied by a 71% decrease in energy consumption

and 70% in carbon emissions. These findings underscore the significant efficiency gains and environmental benefits afforded by our approach.

5.2.7 Expanded Resources

To achieve a single robust model applicable across tasks, we trained an expanded-resource version of CACARA: $\text{CACARA}_{ASC/AA/WC/AC/C/f\ 0.2}$. Unlike previous models trained with 3 epochs, a batch size of 64, and different datasets, this version was trained with 10 epochs, a batch size of 110, and all datasets together (SC + AA + WC + AC + C), with data filtered using a threshold of 0.2. In this phase, training was performed on a single A100 GPU with 80GB of VRAM.

Table 5.11 shows the retrieval performance of the model with expanded resources. A direct comparison of this expanded model against the best-performing optimized models can be seen in Figure 5.6 and Figure 5.7, with a complete comparison in Table D.1. Evaluation is done for audio-to-text and text-to-audio retrieval across 12 languages. This task in the AudioCaps dataset did not improve with more resources and training time, and continues to yield values lower than those obtained with the optimized model. This might be due to the trained sets being close to the test set’s distribution. However, it still improved over the same model with fewer resources. For the same task in the ClothoV2 dataset, there is a general improvement in audio-to-text retrieval, but text-to-audio retrieval remains below the optimized model’s results.

For the classification task, comparing the two datasets in Table 5.12, results are generally improved when using more resources and more varied data. This shows that this task benefits from more robust training. Despite benefiting from a more robust structure, the presented model continues to demonstrate excellent results with a modest computational cost, but can achieve better results with training on greater computational power and data availability. In this structure, we continued to train the model only in English; the other languages also improved.

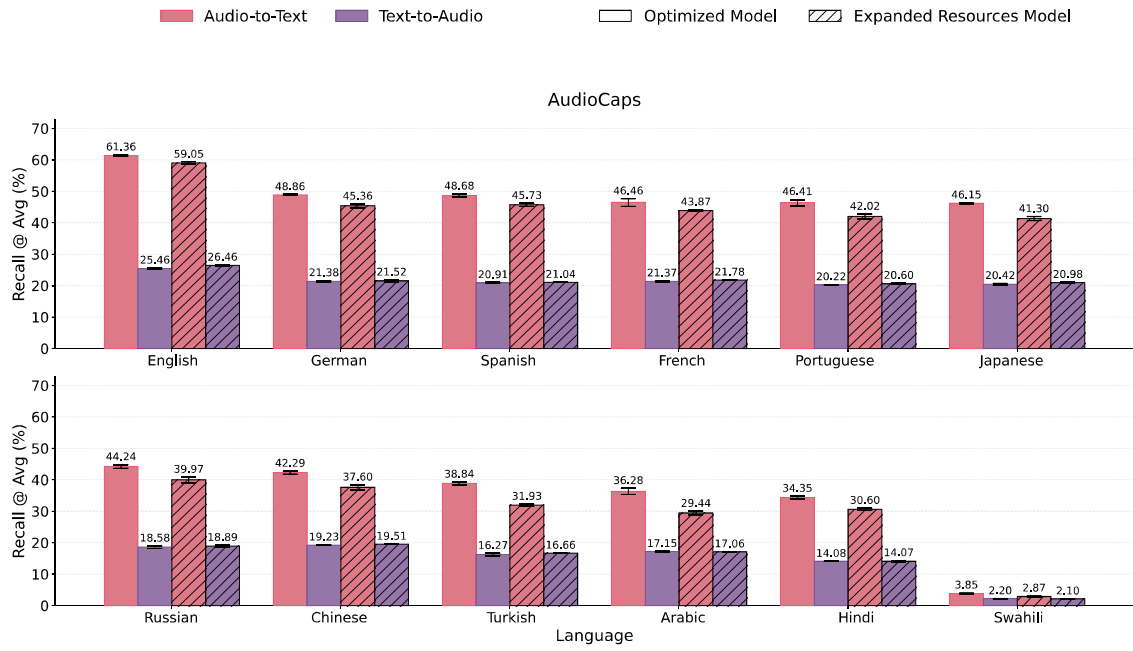


Figure 5.6: Multilingual audio-text retrieval performance of CACARA on AudioCaps. Results are reported as Recall@Avg across 12 languages for both audio-to-text and text-to-audio retrieval. Colors indicate the retrieval direction, while hatching distinguishes the evaluated model configuration.

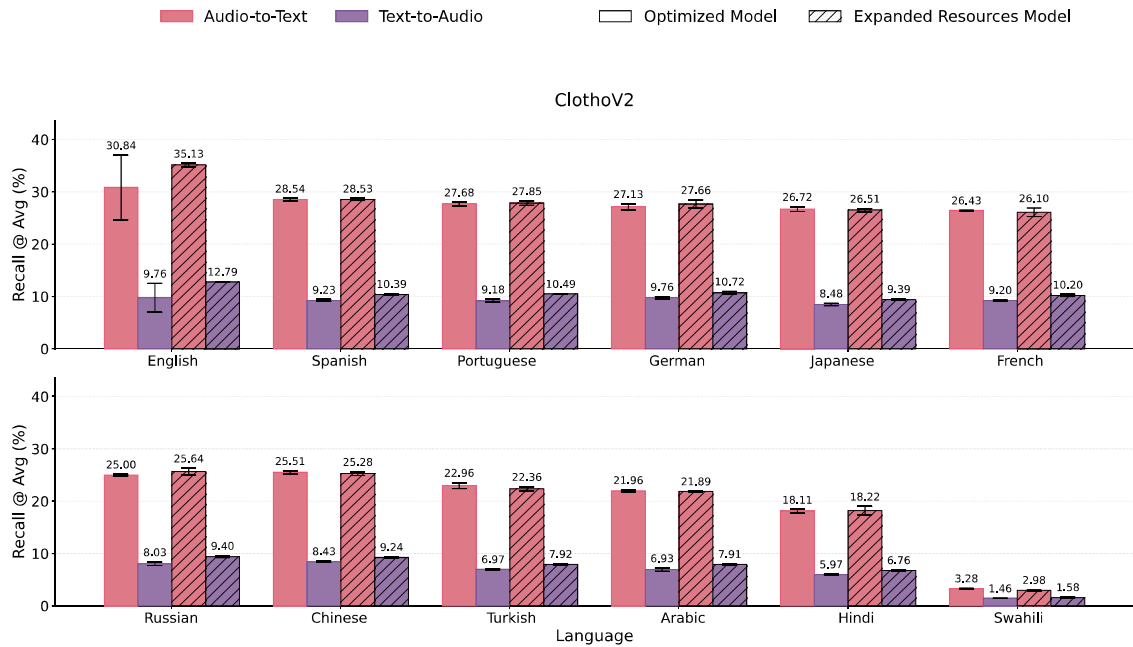


Figure 5.7: Multilingual audio-text retrieval performance of CACARA on ClothoV2. Results are reported as Recall@Avg across 12 languages for both audio-to-text and text-to-audio retrieval. Colors indicate the retrieval direction, while hatching distinguishes the evaluated model configuration.

AudioCaps								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	31.45 ± 0.14	66.09 ± 0.31	79.61 ± 0.45	59.05 ± 0.28	7.64 ± 0.09	28.78 ± 0.21	42.95 ± 0.14	26.46 ± 0.12
Portuguese	18.54 ± 0.98	46.57 ± 0.75	60.96 ± 0.53	42.02 ± 0.74	5.99 ± 0.10	22.07 ± 0.43	33.75 ± 0.06	20.60 ± 0.19
Spanish	21.03 ± 0.38	<u>50.85</u> ± 0.26	<u>65.32</u> ± 0.80	<u>45.73</u> ± 0.45	6.04 ± 0.10	22.46 ± 0.21	34.62 ± 0.05	21.04 ± 0.06
French	20.33 ± 0.42	48.61 ± 0.36	62.67 ± 0.38	43.87 ± 0.27	<u>6.25</u> ± 0.08	<u>23.44</u> ± 0.22	<u>35.66</u> ± 0.23	<u>21.78</u> ± 0.06
Russian	17.58 ± 0.35	44.19 ± 1.45	58.16 ± 0.89	39.97 ± 0.89	5.50 ± 0.32	19.88 ± 0.34	31.28 ± 0.42	18.89 ± 0.30
Arabic	11.18 ± 0.47	31.96 ± 0.86	45.17 ± 0.51	29.44 ± 0.58	4.71 ± 0.17	18.37 ± 0.19	28.10 ± 0.18	17.06 ± 0.07
Hindi	11.70 ± 0.30	33.33 ± 0.71	46.76 ± 0.76	30.60 ± 0.38	3.83 ± 0.11	15.10 ± 0.44	23.28 ± 0.23	14.07 ± 0.24
German	<u>21.33</u> ± 0.13	50.33 ± 0.90	64.41 ± 0.90	45.36 ± 0.62	5.85 ± 0.08	23.06 ± 0.38	35.64 ± 0.46	21.52 ± 0.30
Chinese(zh)	15.27 ± 0.74	41.17 ± 1.37	56.36 ± 0.43	37.60 ± 0.80	5.57 ± 0.31	20.79 ± 0.16	32.16 ± 0.52	19.51 ± 0.05
Swahili	0.61 ± 0.18	2.89 ± 0.32	5.10 ± 0.18	2.87 ± 0.21	0.59 ± 0.03	2.21 ± 0.06	3.49 ± 0.10	2.10 ± 0.06
Japanese	18.35 ± 0.52	45.76 ± 0.96	59.78 ± 0.80	41.30 ± 0.68	5.76 ± 0.18	22.74 ± 0.30	34.45 ± 0.05	20.98 ± 0.10
Turkish	12.76 ± 0.15	35.07 ± 0.38	47.96 ± 0.51	31.93 ± 0.34	4.52 ± 0.16	17.64 ± 0.22	27.81 ± 0.29	16.66 ± 0.06

ClothoV2								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	15.67 ± 0.50	38.46 ± 0.44	51.25 ± 0.06	35.13 ± 0.31	3.28 ± 0.17	13.57 ± 0.14	21.51 ± 0.05	12.79 ± 0.08
Portuguese	11.17 ± 0.20	30.45 ± 0.28	41.94 ± 0.72	27.85 ± 0.40	2.64 ± 0.10	10.92 ± 0.06	17.91 ± 0.10	10.49 ± 0.06
Spanish	<u>11.67</u> ± 0.40	<u>31.33</u> ± 0.25	<u>42.58</u> ± 0.39	<u>28.53</u> ± 0.23	2.63 ± 0.18	10.87 ± 0.18	17.66 ± 0.46	10.39 ± 0.15
French	10.45 ± 0.74	28.07 ± 1.20	39.78 ± 0.82	26.10 ± 0.80	2.56 ± 0.21	10.68 ± 0.32	17.36 ± 0.12	10.20 ± 0.19
Russian	10.09 ± 0.52	27.71 ± 0.76	39.13 ± 0.77	25.64 ± 0.66	2.54 ± 0.09	9.79 ± 0.31	15.85 ± 0.29	9.40 ± 0.20
Arabic	7.99 ± 0.10	23.64 ± 0.12	34.04 ± 0.28	21.89 ± 0.03	2.04 ± 0.02	8.22 ± 0.14	13.47 ± 0.21	7.91 ± 0.12
Hindi	6.81 ± 0.66	19.51 ± 0.89	28.32 ± 0.85	18.22 ± 0.79	1.85 ± 0.09	6.98 ± 0.32	11.46 ± 0.07	6.76 ± 0.11
German	11.22 ± 0.39	30.31 ± 0.74	41.44 ± 1.40	27.66 ± 0.81	<u>2.72</u> ± 0.07	<u>11.12</u> ± 0.31	<u>18.30</u> ± 0.48	<u>10.72</u> ± 0.27
Chinese(zh)	9.98 ± 0.15	27.28 ± 0.14	38.59 ± 0.80	25.28 ± 0.36	2.48 ± 0.10	9.71 ± 0.17	15.53 ± 0.21	9.24 ± 0.14
Swahili	0.75 ± 0.10	2.98 ± 0.07	5.20 ± 0.38	2.98 ± 0.10	0.55 ± 0.01	1.65 ± 0.12	2.54 ± 0.23	1.58 ± 0.12
Japanese	10.48 ± 0.23	28.87 ± 0.26	40.19 ± 0.44	26.51 ± 0.31	2.35 ± 0.20	9.80 ± 0.22	16.03 ± 0.16	9.39 ± 0.17
Turkish	8.45 ± 0.06	24.02 ± 0.50	34.61 ± 0.75	22.36 ± 0.41	2.14 ± 0.07	8.06 ± 0.24	13.56 ± 0.19	7.92 ± 0.13

Table 5.11: Recall values for the Audio-to-Text and Text-to-Audio retrieval tasks using the expanded resources model across AudioCaps and ClothoV2 for the twelve evaluated languages. **Bolded** results indicate the best performance for each metric within each dataset and retrieval task, while underlined results represent the second-best values.

5.2.8 Qualitative Analysis

To qualitatively evaluate our model’s cross-modal retrieval capabilities, we conducted experiments on the AudioCaps test set. We present representative examples of successful and unsuccessful retrieval outcomes for both text-to-audio and audio-to-text tasks in Figures 5.8 and 5.9. Furthermore, we investigate the emergent capabilities of our model in audio-to-image and image-to-audio retrieval tasks, demonstrating its ability to implicitly align the audio and image modalities that were not explicitly aligned during training. Specifically, we leveraged the test subset of the VGG-Sound dataset, extracting audio and the central video frame following the methodology outlined in Guzhov et al. [76].

Figure 5.8 presents qualitative examples for audio-text retrieval on AudioCaps. The top part shows text-to-audio retrieval, where a textual query is used to retrieve the most similar audio clips. Since audio clips are not directly visible, each retrieved result is

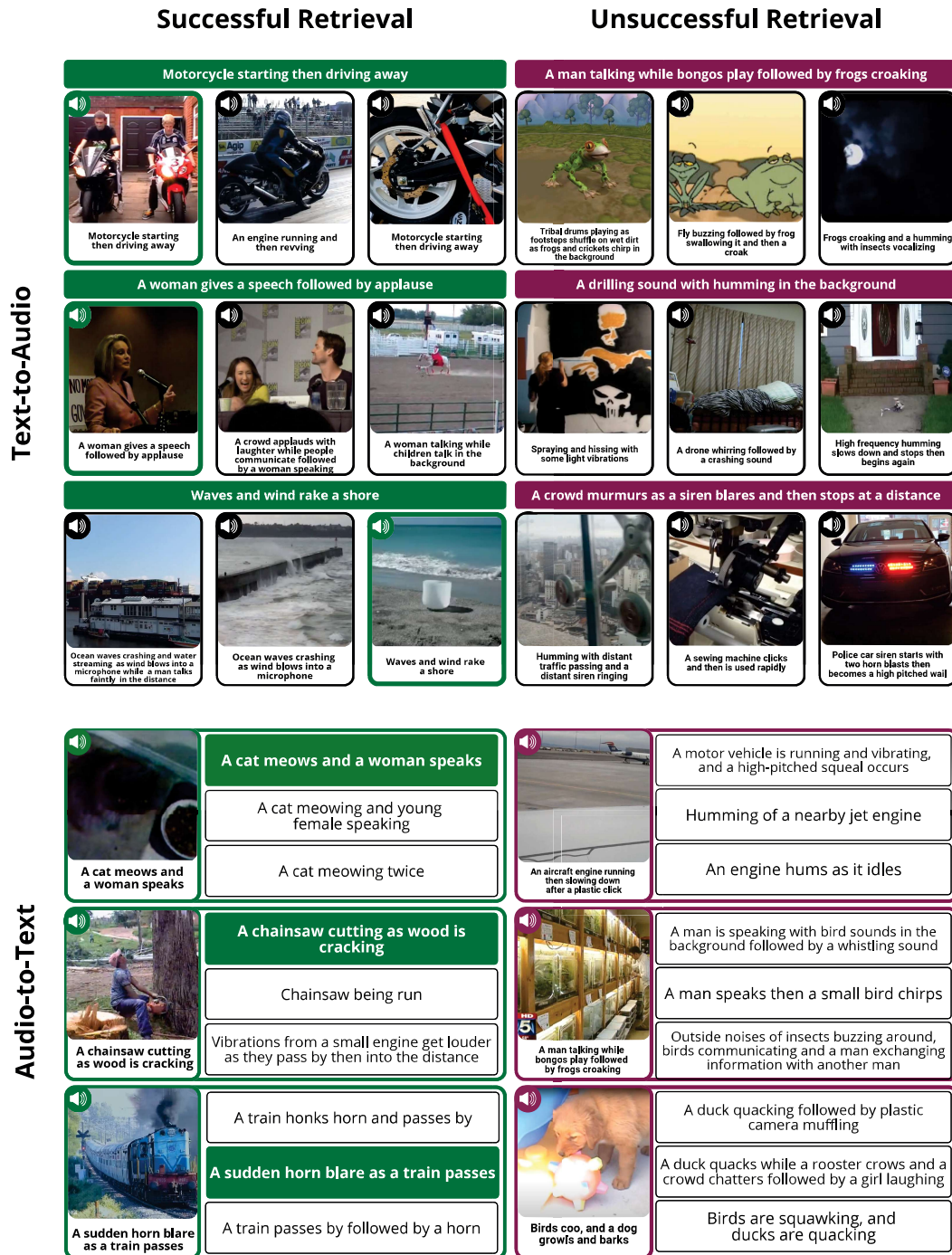


Figure 5.8: Examples of successful (left) and unsuccessful (right) text-to-audio retrieval, shown at the top of the image. The green-highlighted image and text indicate the ground-truth audio clip retrieved within the top 3 results for the given text query. For unsuccessful text queries, we present all returned audio clips as images. At the bottom, successful (left) and unsuccessful (right) audio-to-text retrievals are shown. The green-highlighted text indicates the ground-truth text description retrieved for the given audio query. For unsuccessful audio queries, all the returned text results are presented.

	OB Model	ER Model	OB Model	ER Model
	ESC-50		UrbanSounds8K	
Language	Accuracy			
English	91.35 $\pm$ 0.69	93.50 $\pm$ 0.18	74.99 $\pm$ 0.55	78.33 $\pm$ 2.75
Portuguese	80.92 $\pm$ 0.98	81.70 $\pm$ 0.88	71.38 $\pm$ 1.35	68.15 $\pm$ 1.63
Spanish	85.60 $\pm$ 0.78	87.65 $\pm$ 0.43	71.02 $\pm$ 0.67	72.72 $\pm$ 1.50
French	82.45 $\pm$ 0.49	83.85 $\pm$ 0.17	67.72 $\pm$ 1.12	70.47 $\pm$ 1.82
Russian	78.30 $\pm$ 1.08	80.62 $\pm$ 1.36	67.16 $\pm$ 1.32	69.32 $\pm$ 0.22
Arabic	63.65 $\pm$ 1.09	63.67 $\pm$ 0.77	63.77 $\pm$ 0.60	62.40 $\pm$ 1.48
Hindi	60.12 $\pm$ 0.83	62.57 $\pm$ 0.88	58.11 $\pm$ 1.15	56.55 $\pm$ 0.78
German	75.92 $\pm$ 1.08	78.37 $\pm$ 0.64	72.38 $\pm$ 0.10	74.44 $\pm$ 1.24
Chinese(zh)	81.23 $\pm$ 1.07	82.80 $\pm$ 0.93	67.22 $\pm$ 1.15	68.50 $\pm$ 0.59
Swahili	20.48 $\pm$ 2.17	21.80 $\pm$ 0.22	38.35 $\pm$ 1.97	36.28 $\pm$ 3.61
Japanese	81.78 $\pm$ 0.75	85.83 $\pm$ 1.46	73.03 $\pm$ 0.74	72.18 $\pm$ 1.88
Turkish	69.85 $\pm$ 1.97	72.92 $\pm$ 0.81	63.91 $\pm$ 0.82	64.52 $\pm$ 0.82

Table 5.12: *Classification task results on the ESC-50 and UrbanSound8K datasets across the twelve evaluated languages. **Bold** values indicate the best mean accuracy for each language across model configurations. OB denotes the Optimized Basic Model, while ER denotes the Expanded Resources Model. Values highlighted in blue indicate the best overall mean accuracy for each dataset across languages.*

represented by a frame extracted from the corresponding video segment, together with its associated caption. The bottom part shows audio-to-text retrieval, where an audio query is used to retrieve textual descriptions. In both directions, successful examples are shown on the left and unsuccessful examples on the right. Green-highlighted results indicate the ground-truth item retrieved within the top-ranked candidates.

Figure 5.9 illustrates examples of emergent audio-image retrieval on VGG-Sound. The left side shows image-to-audio retrieval, where an image query is used to retrieve the corresponding audio clip, represented visually by frames from the retrieved video segments. The right side shows audio-to-image retrieval, where an audio query is used to retrieve the corresponding image frame. The green-highlighted item indicates the ground-truth match among the retrieved candidates.

5.2.9 Final Remarks

The results presented in this section show that CACARA provides a viable strategy for expanding a frozen multilingual multimodal representation space with a new audio modality. By aligning the audio encoder with a frozen textual anchor, the model was able to establish connections not only between audio and text, but also between audio and image, even without explicit audio-image training pairs.

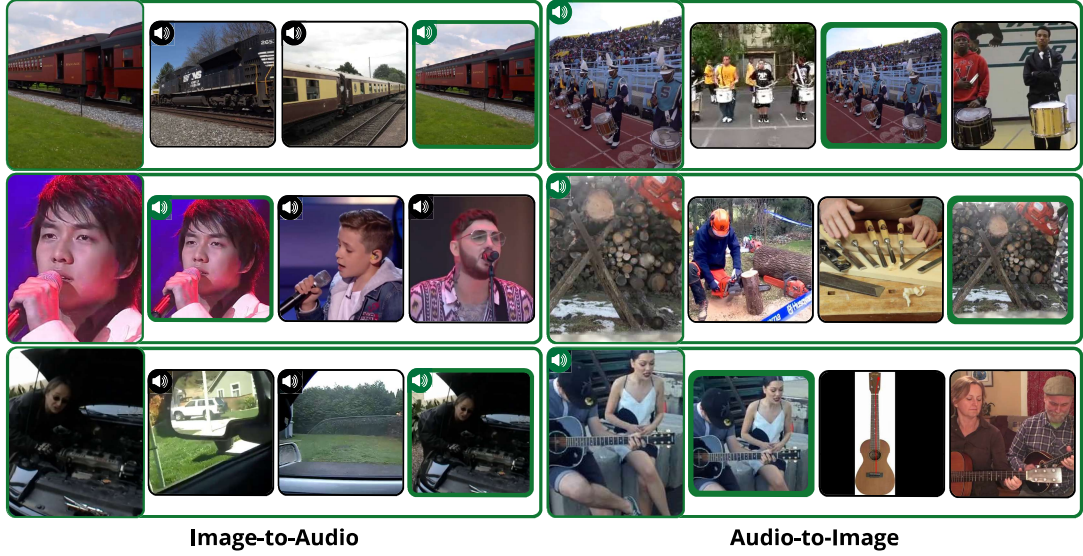


Figure 5.9: Examples of successful audio-to-image retrieval (left) and image-to-audio retrieval (right). The green-highlighted image shows the ground-truth image retrieved for the given audio query.

Experiments also show that this mechanism can transfer multilingual structure to the newly added audio modality. However, this transfer is not uniform across languages. The multilingual analysis revealed that the quality of audio-text alignment in non-English languages depends on how these languages are organized in the frozen multilingual text space before the addition of audio. This behavior motivated the concept of the cross-lingual gap, which helps explain why some languages remain closer to English in the shared space, while others exhibit greater alignment degradation. Therefore, CACARA demonstrates that multilinguality can emerge from English-only audio-text supervision, but also shows that the geometry and linguistic coverage of the original multilingual encoder constrain this emergence.

Finally, the efficiency analysis reinforces CACARA’s practical relevance. Instead of requiring full tri-modal retraining, the proposed framework reuses frozen pretrained components and concentrates training on the newly introduced audio branch. This makes multimodal expansion more accessible under limited computational resources, while still producing competitive retrieval and classification results. Overall, CACARA shows that sequential alignment is a promising path for expanding multilingual multimodal systems, although its effectiveness depends on the quality of the frozen representation space and on the degree of cross-lingual structure already present in it.

5.3 PALMA

This section presents the empirical results obtained with PALMA. The analysis evaluates how different architectural, optimization, and training choices affect bimodal audio-language representation learning. Unlike CACARA, which investigates the integration of audio into a frozen multilingual vision-language space, PALMA examines the audio-text setting directly to understand how audio encoders, multilingual text encoders, training strategies, layer aggregation mechanisms, data composition, batch size, and multilingual supervision influence cross-modal retrieval and zero-shot audio classification.

5.3.1 Experiments and Results

The experimental analysis of PALMA is organized as a multi-phase optimization pipeline. Each phase isolates a single key factor in audio-language alignment, enabling the final configuration to be selected progressively. First, Section 5.3.2 investigates the effect of learning rate and weight decay across different audio encoders. This phase establishes a stable optimization setting for PALMA and provides the basis for the following experiments. Then, Section 5.3.3 analyzes the impact of the multilingual text encoder by pairing different text backbones with the selected audio encoders to identify which audio-text combinations produce the strongest shared embedding space.

The next stages focus on training efficiency and architectural design. Section 5.3.4 evaluates LiT and Sigmoid loss, examining whether strategies adapted from the vision-language literature can improve the trade-off between performance and computational cost in the audio-language domain. Section 5.3.5 then introduces and evaluates the proposed Mixture-of-Layers mechanism, including its main retrieval and classification results, an analysis of learned layer-weight distributions, additional dual-encoder and projection-only ablations, and an exploratory study of data augmentation.

After selecting the main PALMA configuration, the analysis turns to data and scaling factors. Section 5.3.6 studies how different training-data combinations affect retrieval and zero-shot classification, highlighting the role of dataset compatibility, caption quality, and domain coverage. Section 5.3.7 investigates the effect of batch size, since contrastive learning depends strongly on the number of negative examples available during training. Finally, Section 5.3.8 examines multilingual supervision. This final phase evaluates whether PALMA can preserve and improve multilingual audio-text alignment through translated captions, compares different multilingual training strategies, and contrasts the strongest PALMA configurations with multilingual audio-language baselines.

5.3.2 Hyperparameter Tuning

The first stage of the PALMA optimization pipeline focused on hyperparameter tuning. In this phase, we analyzed two training hyperparameters: learning rate (LR) and weight decay (WD). These parameters were selected because of their strong influence on optimization stability, convergence behavior, and final model performance. Three pretrained audio encoders were evaluated under different LR and WD configurations: HTS-AT, BEATs, and CED. These encoders were chosen because they represent different trade-offs between performance, computational cost, and prior effectiveness in audio representation learning. HTS-AT was included due to its favorable balance between performance and efficiency, as discussed in Section 5.2.2. BEATs was also considered due to its strong performance in previous experiments with CACARA, while CED was included as a more recent SOTA acoustic encoder [39].

Following the learning-rate scale adopted in Section 5.2.2, we evaluated the set $\{5 \times 10^{-4}, 5 \times 10^{-5}, 5 \times 10^{-6}\}$, varying the LR by one order of magnitude. For weight decay, we evaluated the set $\{10^{-6}, 10^{-4}, 10^{-2}\}$, varying the values by two orders of magnitude. This wider spacing was adopted to cover substantially different regularization regimes while avoiding an excessively large combinatorial number of experiments.

The experiments in this phase were done with a fixed batch size of 48 for all models and configurations. This design choice differs from the strategy used in CACARA, which prioritized the maximum feasible batch size. In PALMA, the main goal of this phase is to keep the optimization setting controlled, allowing the effects of the encoder architecture, learning rate, and weight decay to be analyzed directly, to select hyperparameter values that are consistently effective across different audio encoders, providing a reliable basis for the following experimental phases. Each model was trained for three epochs, corresponding to approximately 111k optimization steps, using a combined training set composed of AudioSetCaps, AudioCaps, and ClothoV2. In this phase, we also adopted XLM-RoBERTa [31] as the text encoder, which was fine-tuned jointly with the audio encoders. This choice was motivated by its previous use in OpenCLIP [91], making it a reliable multilingual textual backbone for multimodal alignment.

Tables 5.13 and 5.14 summarize the results of the first PALMA optimization phase on audio-text retrieval using AudioCaps and ClothoV2, considering both retrieval directions: audio-to-text and text-to-audio, and in zero-shot classification on ESC-50 and UrbanSounds8K.

The results show that, as expected, the learning rate had a strong impact on both retrieval and classification performance. The largest learning rate, 5×10^{-4} , generally produced weak results across experiments. For instance, with CED, the AudioCaps audio-to-text R@Avg reached only 23.66%, 26.66%, and 15.07% for weight decay values of 10^{-6} , 10^{-4} , and 10^{-2} , respectively. These values are substantially lower than the

Encoder	Max LR	WD	AudioCaps								ClothoV2							
			Audio-to-Text				Text-to-Audio				Audio-to-Text				Text-to-Audio			
			R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
CED	5×10^{-4}	10^{-6}	7.65	25.53	37.80	23.66	6.79	22.56	35.23	21.53	3.25	9.47	14.26	9.00	2.22	8.69	14.24	8.38
CED	5×10^{-4}	10^{-4}	9.22	28.80	41.96	26.66	6.97	25.31	38.90	23.73	3.25	10.33	16.46	10.02	2.58	9.26	15.37	9.07
CED	5×10^{-4}	10^{-2}	5.06	16.20	23.96	15.07	4.34	13.84	22.20	13.46	1.72	6.79	11.77	6.76	1.42	6.56	10.53	6.17
CED	5×10^{-5}	10^{-6}	43.76	77.05	87.63	69.48	35.64	71.72	84.75	64.03	16.94	36.65	48.33	33.97	11.98	29.67	41.34	27.66
CED	5×10^{-5}	10^{-4}	35.32	67.49	79.87	60.89	28.26	61.35	76.60	55.40	11.00	28.52	38.47	26.00	7.71	22.43	32.90	21.01
CED	5×10^{-5}	10^{-2}	5.96	15.64	24.52	15.37	4.23	14.65	23.69	14.19	1.82	5.84	9.95	5.87	1.11	5.19	9.03	5.11
CED	5×10^{-6}	10^{-6}	<u>40.72</u>	<u>71.20</u>	<u>84.70</u>	<u>65.54</u>	<u>33.25</u>	<u>69.09</u>	<u>83.06</u>	<u>61.80</u>	14.45	<u>33.01</u>	<u>43.64</u>	<u>30.37</u>	<u>11.83</u>	30.74	43.18	28.58
CED	5×10^{-6}	10^{-4}	37.12	68.39	81.78	62.43	30.26	64.25	79.98	58.16	<u>15.98</u>	31.67	42.49	30.05	11.37	<u>29.93</u>	<u>41.44</u>	27.58
CED	5×10^{-6}	10^{-2}	10.46	27.22	38.13	25.27	7.94	24.45	36.78	23.06	3.83	10.33	16.46	10.21	2.64	9.72	15.31	9.22
HTS-AT	5×10^{-4}	10^{-6}	0.11	0.45	1.24	0.60	0.07	0.54	1.08	0.56	0.10	0.29	0.77	0.38	0.10	0.34	0.77	0.40
HTS-AT	5×10^{-4}	10^{-4}	0.11	0.56	1.01	0.56	0.09	0.54	0.94	0.52	0.10	0.48	0.96	0.51	0.13	0.46	0.90	0.50
HTS-AT	5×10^{-4}	10^{-2}	2.70	10.12	16.76	9.86	2.63	9.56	15.59	9.26	1.24	4.02	7.08	4.11	0.94	3.71	7.12	3.92
HTS-AT	5×10^{-5}	10^{-6}	23.73	54.11	69.52	49.12	20.09	49.65	65.51	45.08	8.80	23.06	34.93	22.26	6.68	20.06	30.33	19.02
HTS-AT	5×10^{-5}	10^{-4}	<u>20.36</u>	<u>50.06</u>	<u>65.35</u>	<u>45.26</u>	<u>18.31</u>	<u>47.72</u>	<u>62.59</u>	<u>42.87</u>	<u>6.70</u>	<u>19.33</u>	<u>27.56</u>	<u>17.86</u>	<u>5.51</u>	<u>18.43</u>	<u>27.29</u>	<u>17.08</u>
HTS-AT	5×10^{-5}	10^{-2}	4.50	16.09	23.96	14.85	4.88	17.82	27.29	16.66	1.44	5.36	9.86	5.55	1.80	6.53	11.52	6.62
HTS-AT	5×10^{-6}	10^{-6}	10.62	31.65	45.49	29.25	2.61	10.69	18.22	10.51	5.74	14.64	21.91	14.10	3.94	12.90	19.64	12.16
HTS-AT	5×10^{-6}	10^{-4}	11.25	32.44	45.65	29.78	2.56	11.11	18.83	10.83	4.59	12.15	20.00	12.25	3.46	12.80	20.00	12.09
HTS-AT	5×10^{-6}	10^{-2}	12.82	31.83	47.02	30.56	10.62	31.65	45.49	29.25	0.77	2.87	5.74	3.13	0.78	3.33	6.07	3.39
BEATs	5×10^{-4}	10^{-6}	6.41	20.13	29.92	18.82	4.72	18.31	29.76	17.60	2.97	8.90	13.97	8.61	1.76	7.10	12.92	7.26
BEATs	5×10^{-4}	10^{-4}	0.11	0.45	1.35	0.64	0.04	0.29	0.65	0.33	0.19	0.48	0.96	0.54	0.04	0.42	0.75	0.40
BEATs	5×10^{-4}	10^{-2}	4.27	13.27	22.27	13.27	3.96	12.04	19.48	11.83	1.53	6.03	10.33	5.96	1.32	5.34	9.70	5.45
BEATs	5×10^{-5}	10^{-6}	43.42	76.49	86.50	68.80	36.56	71.41	84.03	64.00	<u>15.31</u>	34.16	47.56	32.34	9.53	25.65	36.31	23.83
BEATs	5×10^{-5}	10^{-4}	34.20	65.69	79.98	59.96	28.03	60.52	76.22	54.92	10.72	28.61	39.62	26.32	7.56	22.24	32.54	20.78
BEATs	5×10^{-5}	10^{-2}	4.50	12.15	21.82	12.82	4.16	13.70	22.32	13.39	1.34	5.17	9.00	5.17	1.97	6.05	9.82	5.95
BEATs	5×10^{-6}	10^{-6}	<u>40.16</u>	<u>73.00</u>	<u>83.91</u>	<u>65.69</u>	<u>34.08</u>	<u>68.59</u>	<u>82.59</u>	<u>61.75</u>	15.98	32.92	43.83	<u>30.91</u>	11.94	29.84	42.56	28.11
BEATs	5×10^{-6}	10^{-4}	37.57	70.53	82.90	63.67	33.41	68.14	81.64	61.06	13.59	<u>33.30</u>	<u>44.50</u>	<u>30.46</u>	<u>11.48</u>	<u>29.67</u>	<u>41.45</u>	<u>27.53</u>
BEATs	5×10^{-6}	10^{-2}	1.69	5.06	7.65	4.80	2.09	6.95	12.49	7.18	0.77	3.92	5.17	3.29	0.88	4.19	7.66	4.24

Table 5.13: Retrieval task results for the first PALMA ablation stage, evaluated on AudioCaps and ClothoV2 across Audio-to-Text and Text-to-Audio retrieval. LR denotes learning rate, and WD denotes weight decay. **Bold** values indicate the best configuration for each encoder and metric, while underlined values indicate the second-best configuration for the same encoder and metric. Values highlighted in **blue** indicate the best overall result for each metric.

69.48% obtained with learning rate 5×10^{-5} and weight decay 10^{-6} , corresponding to drops of 45.82, 42.82, and 54.41 pp, respectively. A similar pattern appears for BEATs, where the same high learning rate produced AudioCaps audio-to-text R@Avg of 18.82%, 0.64%, and 13.27%, while the intermediate learning rate with weight decay 10^{-6} reached 68.80 pp.

This behavior was even more evident for HTS-AT. With a learning rate of 5×10^{-4} on the AudioCaps dataset for audio-to-text retrieval, the configuration achieved R@Avg scores of 0.60% and 0.56%. Even the best configuration at this learning rate, with weight decay of 10^{-2} , achieved only 9.86% R@Avg. In contrast, reducing the learning rate to 5×10^{-5} increased the same metric to 49.12% with weight decay 10^{-6} , an improvement of 48.52 pp over the corresponding 5×10^{-4} and 10^{-6} configuration. The same tendency appears in classification, where HTS-AT reached 1.55%, 2.00%, and 11.30% accuracy on ESC-50, and 10.35, 4.83 and 21.41 pp on UrbanSounds with learning rate 5×10^{-4} , while configurations with 5×10^{-5} reached 42.90% and 43.95%.

A possible explanation for this behavior is the difference in how the text encoder is handled across frameworks. In CACARA, the text encoder is frozen and serves as a

Encoder	LR	WD	Classification	
			ESC-50	UrbanSound8K
CED	5×10^{-4}	10^{-6}	27.25	41.19
	5×10^{-4}	10^{-4}	26.65	38.36
	5×10^{-4}	10^{-2}	15.95	21.98
	5×10^{-5}	10^{-6}	73.70	67.16
	5×10^{-5}	10^{-4}	60.70	65.09
	5×10^{-5}	10^{-2}	15.40	18.61
	5×10^{-6}	10^{-6}	<u>73.65</u>	68.94
	5×10^{-6}	10^{-4}	66.25	<u>67.27</u>
	5×10^{-6}	10^{-2}	17.55	26.17
HTS-AT	5×10^{-4}	10^{-6}	1.55	10.35
	5×10^{-4}	10^{-4}	2.00	4.83
	5×10^{-4}	10^{-2}	11.30	21.47
	5×10^{-5}	10^{-6}	<u>42.90</u>	59.98
	5×10^{-5}	10^{-4}	43.95	<u>58.73</u>
	5×10^{-5}	10^{-2}	18.20	31.96
	5×10^{-6}	10^{-6}	33.45	52.54
	5×10^{-6}	10^{-4}	36.20	54.41
	5×10^{-6}	10^{-2}	10.40	24.12
BEATs	5×10^{-4}	10^{-6}	20.10	42.10
	5×10^{-4}	10^{-4}	2.55	10.81
	5×10^{-4}	10^{-2}	15.15	28.47
	5×10^{-5}	10^{-6}	71.75	<u>68.34</u>
	5×10^{-5}	10^{-4}	61.35	68.72
	5×10^{-5}	10^{-2}	11.85	27.25
	5×10^{-6}	10^{-6}	<u>69.15</u>	67.63
	5×10^{-6}	10^{-4}	69.00	62.61
	5×10^{-6}	10^{-2}	66.35	64.03

Table 5.14: Classification task results for the first PALMA ablation stage, evaluated on the ESC-50 and UrbanSound8K datasets. LR denotes the maximum learning rate, and WD denotes weight decay. **Bold** values indicate the best configuration for each encoder and dataset, while underlined values indicate the second-best configuration for the same encoder and dataset. Values highlighted in blue indicate the best overall mean accuracy for each dataset.

fixed multilingual anchor, so optimization focuses on adapting the audio encoder to an already stable representation space. In the PALMA experiments of this phase, however, the text encoder is updated together with the audio encoder. Therefore, even if an audio encoder such as HTS-AT can benefit from higher learning rates in isolation, the joint optimization of the audio and text encoders may be less stable at high learning rates.

Furthermore, the intermediate learning rate, 5×10^{-5} , was the most reliable across the three audio encoders. For CED, this setting with weight decay 10^{-6} achieved the best AudioCaps audio-to-text R@Avg of 69.48%, the best ClothoV2 audio-to-text R@Avg of 33.97%, and the best ESC-50 accuracy of 73.70%. For BEATs, the same

learning rate and weight decay also produced strong results, with AudioCaps in the audio-to-text and text-to-audio scenarios. For HTS-AT, this learning rate produced its best overall retrieval behavior, reaching 49.12% AudioCaps audio-to-text R@Avg and 45.08% text-to-audio R@Avg with weight decay 10^{-6} .

The smallest learning rate, 5×10^{-6} , also produced competitive results in several cases, but its dominance was less consistent. In some scenarios, such as ClothoV2 text-to-audio results and UrbanSound8K classification, the smaller learning rate was competitive. Suggesting that the LR of 5×10^{-6} might also be a stable option.

Weight decay also had a clear effect on performance. The smallest value, 10^{-6} , was the most consistently effective across learning rates, encoders, and tasks. In most cases, the best results were obtained when 10^{-6} was combined with the intermediate learning rate 5×10^{-5} . For example, this combination produced the best or near-best retrieval results for CED, BEATs, and HTS-AT on AudioCaps, and also strong results on ClothoV2 and ESC-50. The intermediate weight decay value, 10^{-4} , was sometimes competitive but less consistent, while the largest weight decay value, 10^{-2} , substantially degraded performance in most configurations.

The results indicate that the bimodal audio-language setting required a more moderate learning rate for stable alignment. For this reason, the configuration 5×10^{-5} learning rate and 10^{-6} weight decay was adopted for the following experiments.

5.3.3 Text Encoder Selection

The second phase investigated the impact of the multilingual text encoder on PALMA. While Phase 1 focused on optimizing the audio side of the model, Phase 2 evaluates whether different text encoders produce better alignment when paired with each audio encoder.

The evaluated text encoders were XLM-RoBERTa, XLM-RoBERTa from OpenCLIP, E5 Multilingual-Base, LaBSE, and mGTE. These encoders were selected for their multilingual representation quality, relevance to prior multilingual and multimodal literature, and performance on text embedding benchmarks such as MTEB [135]. A summary of these text encoders is presented in Appendix A.4. To maintain computational feasibility and comparability with previous contrastive audio-language models, the selection prioritized models that were encoder-only, had fewer than one billion parameters, and did not depend on prompting strategies during inference.

Tables 5.15 and 5.16 present the results obtained by combining each audio encoder with each text encoder. The retrieval results show that the choice of text encoder indeed affects cross-modal alignment. For HTS-AT, mGTE produced the best results across all retrieval directions and datasets, increasing AudioCaps audio-to-text R@Avg

		Audio-to-Text				Text-to-Audio			
		AudioCaps							
Text Encoder		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
HTS-AT	XLM-RoBERTa	23.73	54.11	69.52	49.12	20.09	49.65	65.51	45.08
	XLM-RoBERTa (OpenCLIP)	<u>25.76</u>	<u>58.83</u>	71.65	<u>52.08</u>	<u>22.16</u>	53.27	67.65	<u>47.69</u>
	E5 Multilingual-Base	23.96	54.33	69.97	49.42	20.99	<u>51.95</u>	66.95	46.63
	LaBSE	25.65	56.69	<u>71.99</u>	51.44	19.96	50.51	65.78	45.41
	mGTE	30.26	58.94	74.58	54.59	23.10	53.27	<u>67.36</u>	47.91
BEATs	XLM-RoBERTa	43.42	76.49	86.50	68.80	36.56	71.41	84.03	64.00
	XLM-RoBERTa (OpenCLIP)	47.58	78.40	87.96	71.32	<u>37.86</u>	73.48	<u>85.69</u>	<u>65.68</u>
	E5 Multilingual-Base	<u>47.13</u>	78.07	88.64	<u>71.28</u>	36.99	72.17	84.72	64.63
	LaBSE	43.19	75.70	88.08	68.99	35.95	71.72	85.22	64.30
	mGTE	46.79	<u>78.18</u>	88.98	71.32	38.27	<u>73.03</u>	85.94	65.74
CED	XLM-RoBERTa	43.76	77.05	87.63	69.48	35.64	71.72	84.75	64.03
	XLM-RoBERTa (OpenCLIP)	44.88	77.28	87.74	69.97	36.33	72.98	85.15	64.82
	E5 Multilingual-Base	47.69	78.07	89.88	71.88	36.63	72.73	<u>85.40</u>	64.92
	LaBSE	44.54	79.87	<u>89.20</u>	71.20	<u>37.57</u>	<u>73.25</u>	84.50	<u>65.11</u>
	mGTE	<u>46.91</u>	<u>78.40</u>	89.09	<u>71.47</u>	38.76	73.77	85.78	66.10
ClothoV2									
Text Encoder		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
HTS-AT	XLM-RoBERTa	8.80	23.06	34.93	22.26	6.68	20.06	30.33	19.02
	XLM-RoBERTa (OpenCLIP)	<u>10.33</u>	24.88	36.08	<u>23.76</u>	<u>7.27</u>	<u>22.07</u>	<u>32.88</u>	<u>20.74</u>
	E5 Multilingual-Base	9.47	<u>24.98</u>	<u>36.84</u>	<u>23.76</u>	6.72	21.86	32.11	20.23
	LaBSE	9.28	24.69	36.17	23.38	6.85	20.21	30.83	19.30
	mGTE	10.62	28.52	38.47	25.87	7.94	23.18	33.19	21.44
BEATs	XLM-RoBERTa	15.31	34.16	47.56	32.34	9.53	25.65	36.31	23.83
	XLM-RoBERTa (OpenCLIP)	15.98	34.93	<u>48.90</u>	33.27	9.86	26.03	36.75	24.21
	E5 Multilingual-Base	15.98	34.93	<u>48.90</u>	33.27	9.86	26.03	36.75	24.21
	LaBSE	18.28	<u>37.51</u>	48.23	<u>34.67</u>	<u>10.43</u>	<u>27.18</u>	<u>37.78</u>	<u>25.13</u>
	mGTE	<u>17.61</u>	38.56	51.39	35.85	10.74	28.33	39.62	26.23
CED	XLM-RoBERTa	<u>16.94</u>	36.65	48.33	33.97	11.98	<u>29.67</u>	<u>41.34</u>	<u>27.66</u>
	XLM-RoBERTa (OpenCLIP)	15.98	37.51	<u>50.91</u>	34.80	11.08	28.50	39.64	26.41
	E5 Multilingual-Base	16.36	<u>38.76</u>	50.53	<u>35.22</u>	10.47	27.81	39.06	25.78
	LaBSE	17.51	37.22	50.33	35.02	<u>11.85</u>	29.05	40.69	27.20
	mGTE	16.84	39.43	51.67	35.98	11.67	29.84	41.91	27.81

Table 5.15: Recall@k results for Audio-to-Text and Text-to-Audio retrieval on AudioCaps and ClothoV2 across all audio and text encoder combinations. **Bold** values indicate the best configuration within each audio encoder for each metric, while underlined values indicate the second-best configuration within the same audio encoder. Values highlighted in **blue** indicate the best overall result for each metric across all configurations.

from 49.12% with XLM-RoBERTa to 54.59%, and ClothoV2 audio-to-text R@Avg from 22.26% to 25.87%. Suggesting that even when the audio encoder is less competitive, a stronger multilingual text encoder can improve the quality of the shared embedding space.

For BEATs, the strongest retrieval results were obtained with XLM-RoBERTa

Text Encoder		Accuracy (%)	
		ESC-50	UrbanSound8K
HTS-AT	XLM-RoBERTa	42.90	59.98
	XLM-RoBERTa (OpenCLIP)	49.35	61.69
	E5 Multilingual-Base	<u>48.85</u>	62.57
	LaBSE	43.20	61.14
	mGTE	44.80	<u>61.76</u>
BEATs	XLM-RoBERTa	71.75	68.34
	XLM-RoBERTa (OpenCLIP)	78.85	72.56
	E5 Multilingual-Base	<u>76.75</u>	<u>71.22</u>
	LaBSE	73.20	62.07
	mGTE	76.00	64.04
CED	XLM-RoBERTa	<u>73.70</u>	67.16
	XLM-RoBERTa (OpenCLIP)	72.40	69.57
	E5 Multilingual-Base	68.60	<u>68.58</u>
	LaBSE	78.65	64.22
	mGTE	67.15	64.61

Table 5.16: Classification accuracy for different audio–text encoder pairings on ESC-50 and UrbanSound8K. **Bold** values indicate the best configuration within each audio encoder for each dataset, while underlined values indicate the second-best configuration within the same audio encoder. Values highlighted in **blue** indicate the best overall result for each dataset across all configurations.

from OpenCLIP, E5 Multilingual-Base, and mGTE. On AudioCaps audio-to-text retrieval, BEATs with XLM-RoBERTa from OpenCLIP and BEATs with mGTE both reached R@Avg of 71.32%, while E5 Multilingual-Base obtained a very close R@Avg of 71.28%. In text-to-audio retrieval, BEATs with mGTE achieved the best result, with R@Avg of 65.74%. On ClothoV2, mGTE was also the strongest pairing for BEATs, reaching R@Avg of 35.85% for audio-to-text and 26.23% for text-to-audio retrieval.

For CED, the results show that the best text encoder depends slightly on the dataset and retrieval direction, but mGTE and E5 Multilingual-Base are the most consistent choices. On AudioCaps audio-to-text retrieval, CED with E5 Multilingual-Base obtained the best overall R@Avg of 71.88%, while CED with mGTE achieved a close R@Avg of 71.47%. However, in AudioCaps text-to-audio retrieval, CED with mGTE achieved the best overall performance, with R@Avg of 66.10%. On ClothoV2, CED with mGTE obtained the best overall R@Avg in both retrieval directions, reaching 35.98% for audio-to-text and 27.81% for text-to-audio.

These results suggest that CED and mGTE form the most balanced configuration for retrieval. Although CED with E5 Multilingual-Base achieved the highest Audio-

Caps audio-to-text R@Avg, CED with mGTE was more consistent across datasets and retrieval directions.

The classification results reveal a different pattern. In zero-shot classification, the best overall results were obtained by BEATs with XLM-RoBERTa from OpenCLIP, which achieved 78.85% accuracy on ESC-50 and 72.56% on UrbanSound8K. CED also obtained strong classification results, especially with LaBSE on ESC-50, reaching 78.65%. However, the CED + mGTE configuration, which was the most consistent in retrieval, did not obtain the highest classification accuracy. This difference suggests that encoder compatibility may depend on the target task, with some audio-text pairings being more effective for fine-grained retrieval, while others may be better suited for zero-shot classification.

Therefore, considering retrieval performance consistency and multilingual alignment rather than isolated classification accuracy, CED with mGTE was selected as the main configuration for the following experiments.

5.3.4 Locked Audio Pretraining and Sigmoid Loss

The third stage investigates whether training strategies adapted from the vision-language literature could improve PALMA’s performance-efficiency trade-off. In particular, we evaluated two strategies: Locked Audio Pretraining (LaT) and the Sigmoid loss used in SigLIP. LaT is inspired by Locked-image Tuning (LiT) [217], but adapted to the audio-language setting by freezing the pretrained audio encoder and training the text encoder and projection layers. In this setting, the audio encoder works like a fixed pre-trained acoustic backbone, while the text side learns to align with its representation space. SigLIP [216], in turn, replaces the standard softmax-based contrastive loss with a pairwise sigmoid objective.

The motivation for this phase was twofold. First, since one of PALMA’s central goals is efficiency, freezing the audio encoder could reduce the number of trainable parameters while preserving the structure of the pretrained acoustic representation. Second, since sigmoid-based contrastive learning has shown promising results in vision-language pretraining [17, 187, 193, 216], we investigated whether this objective could also transfer to audio-language alignment. For this reason, we evaluated four configurations for BEATs and CED using mGTE as the text encoder: the standard mGTE configuration, mGTE + LaT, mGTE + SigLIP, and mGTE + LaT + SigLIP. Each configuration was trained across three independent runs to account for variance. Tables 5.17 and 5.18 report the mean and standard deviation (std) across runs.

Table 5.17 shows that the standard fully fine-tuned configuration (mGTE) remains the strongest option for retrieval in different settings. For CED, the standard mGTE

		Audio-to-Text				Text-to-Audio			
		AudioCaps							
Configuration		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
BEATs	mGTE	47.58 $\pm$ 0.85	78.93 $\pm$ 0.66	89.28 $\pm$ 0.52	71.93 $\pm$ 0.53	38.41 $\pm$ 0.15	73.39 $\pm$ 0.83	85.92 $\pm$ 0.35	65.90 $\pm$ 0.37
	mGTE + LaT	45.18 $\pm$ 1.08	77.50 $\pm$ 0.63	88.64 $\pm$ 0.59	70.44 $\pm$ 0.67	33.85 $\pm$ 0.28	68.98 $\pm$ 0.97	82.52 $\pm$ 0.22	61.78 $\pm$ 0.35
	mGTE + SigLiP	48.07 $\pm$ 0.26	78.59 $\pm$ 0.81	88.79 $\pm$ 0.83	71.82 $\pm$ 0.44	38.83 $\pm$ 0.50	73.96 $\pm$ 0.34	85.71 $\pm$ 0.24	66.17 $\pm$ 0.31
	mGTE + LaT + SigLiP	46.08 $\pm$ 0.72	78.22 $\pm$ 1.02	88.86 $\pm$ 0.12	71.05 $\pm$ 0.55	34.03 $\pm$ 0.37	68.99 $\pm$ 0.21	82.89 $\pm$ 0.38	61.97 $\pm$ 0.18
CED	mGTE	47.47 $\pm$ 0.56	79.49 $\pm$ 0.95	89.77 $\pm$ 0.63	72.24 $\pm$ 0.67	38.88 $\pm$ 0.44	74.13 $\pm$ 0.40	86.17 $\pm$ 0.34	66.39 $\pm$ 0.26
	mGTE + LaT	46.42 $\pm$ 0.82	78.37 $\pm$ 0.28	89.91 $\pm$ 0.37	71.57 $\pm$ 0.41	36.41 $\pm$ 0.74	72.99 $\pm$ 0.75	85.62 $\pm$ 0.61	65.01 $\pm$ 0.63
	mGTE + SigLiP	46.08 $\pm$ 1.13	78.93 $\pm$ 0.96	89.87 $\pm$ 0.89	71.63 $\pm$ 0.46	38.06 $\pm$ 0.27	74.07 $\pm$ 0.13	85.53 $\pm$ 0.39	65.89 $\pm$ 0.18
	mGTE + LaT + SigLiP	47.47 $\pm$ 1.07	78.78 $\pm$ 1.01	89.46 $\pm$ 0.32	71.91 $\pm$ 0.41	36.96 $\pm$ 0.37	72.61 $\pm$ 0.39	85.58 $\pm$ 0.27	65.05 $\pm$ 0.11
ClothoV2									
Configuration		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
BEATs	mGTE	18.18 $\pm$ 0.58	38.72 $\pm$ 1.30	51.74 $\pm$ 0.45	36.22 $\pm$ 0.52	11.22 $\pm$ 0.58	28.40 $\pm$ 0.10	40.15 $\pm$ 0.57	26.59 $\pm$ 0.32
	mGTE + LaT	16.17 $\pm$ 0.69	36.91 $\pm$ 1.21	49.57 $\pm$ 1.63	34.22 $\pm$ 1.13	10.27 $\pm$ 0.16	26.90 $\pm$ 0.27	37.98 $\pm$ 0.62	25.05 $\pm$ 0.32
	mGTE + SigLiP	17.99 $\pm$ 0.73	38.91 $\pm$ 1.36	51.74 $\pm$ 1.05	36.22 $\pm$ 0.63	11.60 $\pm$ 0.13	30.28 $\pm$ 0.27	42.60 $\pm$ 0.76	28.16 $\pm$ 0.27
	mGTE + LaT + SigLiP	16.27 $\pm$ 0.16	35.57 $\pm$ 0.65	48.87 $\pm$ 0.93	33.57 $\pm$ 0.42	10.55 $\pm$ 0.56	27.20 $\pm$ 0.65	38.31 $\pm$ 1.03	25.36 $\pm$ 0.74
CED	mGTE	17.26 $\pm$ 0.80	39.65 $\pm$ 0.47	52.60 $\pm$ 1.01	36.50 $\pm$ 0.74	12.17 $\pm$ 0.45	30.02 $\pm$ 0.18	42.22 $\pm$ 0.27	28.14 $\pm$ 0.29
	mGTE + LaT	16.55 $\pm$ 0.35	37.73 $\pm$ 0.83	50.27 $\pm$ 0.80	34.85 $\pm$ 0.53	10.90 $\pm$ 0.67	28.41 $\pm$ 0.59	40.33 $\pm$ 0.86	26.55 $\pm$ 0.60
	mGTE + SigLiP	17.67 $\pm$ 0.36	38.82 $\pm$ 1.02	51.32 $\pm$ 1.44	35.94 $\pm$ 0.93	12.50 $\pm$ 0.27	31.50 $\pm$ 1.06	44.05 $\pm$ 0.94	29.35 $\pm$ 0.74
	mGTE + LaT + SigLiP	17.42 $\pm$ 0.59	38.85 $\pm$ 0.67	52.44 $\pm$ 1.34	36.24 $\pm$ 0.47	11.52 $\pm$ 0.49	30.41 $\pm$ 0.91	42.43 $\pm$ 0.74	28.12 $\pm$ 0.69

Table 5.17: Recall@k results for Audio-to-Text and Text-to-Audio retrieval on AudioCaps and ClothoV2 across PALMA configurations. Values are reported as mean $\pm$ standard deviation. **Bold** values indicate the best configuration within each audio encoder for each metric and dataset, while underlined values indicate the second-best configuration within the same audio encoder. Values highlighted in **blue** indicate the best overall result for each metric and dataset.

		Accuracy (%)	
		ESC-50	UrbanSound8K
BEATs	mGTE	70.35 ± 6.73	66.52 ± 3.79
	mGTE + LaT	71.98 ± 1.46	64.85 ± 1.05
	mGTE + SigLiP	77.92 ± 2.63	68.28 ± 3.11
	mGTE + LaT + SigLiP	<u>72.47</u> ± 1.53	65.88 ± 1.79
CED	mGTE	71.02 ± 3.42	67.45 ± 2.53
	mGTE + LaT	<u>80.17</u> ± 0.58	70.80 ± 4.34
	mGTE + SigLiP	75.30 ± 1.52	65.71 ± 4.93
	mGTE + LaT + SigLiP	80.95 ± 0.57	<u>68.26</u> ± 1.23

Table 5.18: Classification accuracy on ESC-50 and UrbanSound8K across PALMA configurations. Values are reported as mean $\pm$ standard deviation. **Bold** values indicate the best configuration within each audio encoder for each dataset, while underlined values indicate the second-best configuration within the same audio encoder. Values highlighted in **blue** indicate the best overall result for each dataset.

configuration achieved the best AudioCaps retrieval results, reaching R@Avg of 72.24% for audio-to-text and 66.39% for text-to-audio. When LaT was applied, these values decreased to 71.57% and 65.01%, corresponding to reductions of only 0.67 pp and 1.38 pp, respectively. On ClothoV2, the effect of LaT was more visible. For CED, audio-to-text R@Avg decreased from 36.50% to 34.85%, while text-to-audio R@Avg decreased from 28.14% to 26.55%. This indicates that freezing the audio encoder might introduce a small degradation, but does not strongly damage the cross-modal alignment learned by the model.

The results obtained with SigLIP were competitive in some cases, but less consistent across tasks. For example, with CED, SigLIP achieved the best ClothoV2 text-to-audio R@Avg, reaching 29.35%, which is 1.21 pp higher than the standard mGTE configuration. However, this advantage did not generalize to AudioCaps, where SigLIP remained below the standard configuration in both retrieval directions. A similar pattern appears for BEATs: SigLIP improved some text-to-audio results, especially on ClothoV2, but did not consistently improve all datasets and directions. Therefore, although Sigmoid-based alignment appears useful in specific retrieval settings, it did not provide the most stable overall trade-off.

The combination of LaT and SigLIP also did not produce consistent retrieval gains. In most cases, adding SigLIP to LaT did not recover the small performance lost by freezing the audio encoder. This behavior suggests that, under the current PALMA setup, the Sigmoid loss and LaT mechanisms do not act as complementary improvements for retrieval. Instead, the combined configuration often performed worse than either the standard fully fine-tuned model or the SigLIP-only configuration.

The classification results in Table 5.18, however, show a different tendency. For CED, LaT substantially improved zero-shot classification. On ESC-50, CED + mGTE increased from 71.02% to 80.17% with LaT, an improvement of 9.15 pp. On UrbanSound8K, the same configuration improved from 67.45% to 70.80%, corresponding to a gain of 3.35 pp. This result suggests that freezing the audio encoder helps preserve class-discriminative acoustic information learned during pretraining, which is particularly useful for zero-shot classification.

SigLIP also improved some classification results, especially for BEATs. However, for the selected CED + mGTE configuration, SigLIP did not outperform LaT. CED + mGTE + SigLIP reached 75.30% on ESC-50 and 65.71% on UrbanSound8K, both below the LaT configuration. The combination of LaT and SigLIP achieved the best ESC-50 result, with 80.95%, but it did not improve UrbanSound8K over LaT alone and did not improve retrieval consistently.

One possible explanation for the lack of consistent gains from the sigmoid loss is that the PALMA experimental setting differs substantially from the large-scale

vision-language scenarios in which SigLIP has shown stronger benefits [187, 216]. In the original vision-language setting, sigmoid-based contrastive learning is evaluated with very large training corpora, often involving billions of image-text pairs, and batch sizes reaching tens of thousands of samples. In contrast, PALMA is trained under a more constrained audio-language setup, with fewer paired examples and substantially smaller batch sizes. Therefore, we hypothesize that this experimental regime may not fully exploit the scalability advantages of the sigmoid objective, limiting its impact when compared with the gains reported in large-scale vision-language pretraining.

Additionally, LaT may not fully exploit the fine-grained acoustic information required for successful audio-language retrieval [169]. By freezing the audio encoder, the model preserves the broad acoustic and class-discriminative knowledge already encoded during pretraining, which can explain its strong performance in zero-shot classification. However, this same constraint may limit the model’s ability to adapt to more detailed audio-language correspondences, such as temporal events, subtle sound attributes, or more complex acoustic scenes. This could help explain why LaT improves classification but does not consistently improve retrieval, and why its benefits are less pronounced than those reported for its vision-language counterpart.

Overall, full fine-tuning remains stronger for retrieval, especially on ClothoV2, where the audio encoder model benefits from having a more fine-grained representation. However, LaT provides a more favorable performance-efficiency trade-off. It introduces only a small degradation in retrieval performance, substantially improves zero-shot classification, and reduces training cost by freezing the pretrained audio encoder. Since efficiency is one of PALMA’s main goals, CED + mGTE + LaT was selected as the basis for the following experiments.

5.3.5 Mixture-of-Layers (MoL)

The fourth phase evaluated the proposed Mixture-of-Layers (MoL) strategy, as shown in Figure 5.10. The motivation for this stage was to investigate whether relying only on the final encoder layer is sufficient to obtain a strong representation for audio-language alignment, or whether intermediate layers can provide complementary information that is discarded when only the last layer is used. This question is motivated by a growing body of evidence in different domains showing that the most useful representations are not always located at the end of the network. Details about the implementation of the different variations of MoL is presented in Appendix F.

In speech representation learning, layer-wise analyses of SSL speech models have shown that different encoder layers encode distinct types of information, including acoustic, phonetic, and semantic cues [145, 146, 170]. These results suggest that the

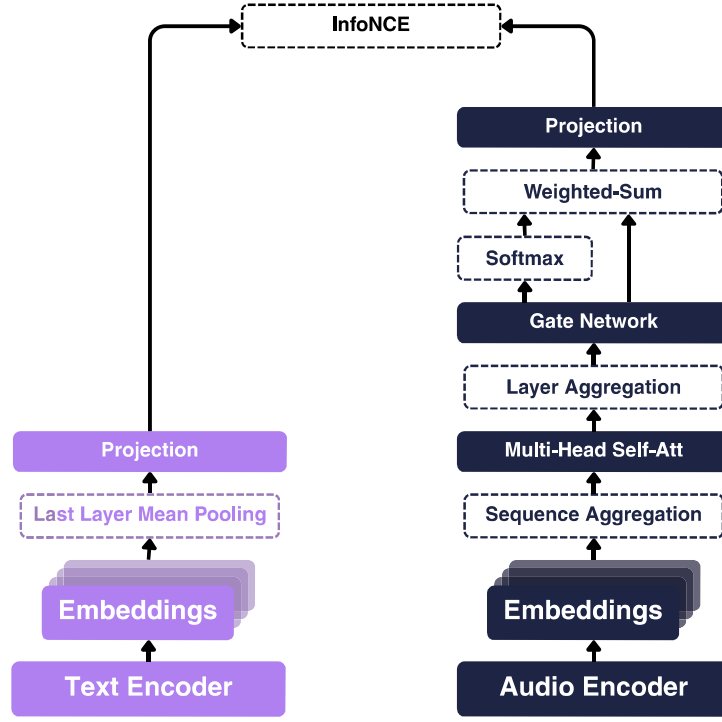


Figure 5.10: Representation of the flow of Mixture of Layers (MoL).

final layer alone may not provide a universally optimal representation for all speech-related tasks.

A similar trend has been reported in NLP and vision. In language models, layer-wise studies of BERT [37] and other Transformer-based models indicate that different layers encode different linguistic and task-specific properties [188]. More recent analyses further challenge the common assumption that final-layer embeddings are the best option, showing that intermediate layers can outperform final layers across a range of downstream embedding tasks [175]. In computer vision, recent work has also shown that the best visual embeddings may be found inside the network rather than at its output, with intermediate representations supporting different downstream tasks such as recognition, visual question answering, and spatial prediction [16].

In the audio-language domain, recent ALMs have also recently started to explore multi-layer aggregation. For instance, GAMA [63] integrates features from multiple layers and uses a multi-layer aggregator over AST (Audio Spectrogram Transformers) [69] model representations to improve audio understanding and reasoning [63]. However, this line of work focuses on LLM-based audio reasoning and instruction-following models, rather than contrastive dual-encoder models for audio-text retrieval and zero-shot classification. Therefore, to the best of our knowledge, the systematic use of a learnable layer aggregation mechanism for contrastive audio-language alignment remains underexplored.

Based on this motivation, PALMA introduces MoL as a mechanism to aggregate

representations from multiple encoder layers before projecting them into the shared audio-text embedding space. Instead of treating the final layer as the only source of information, MoL allows the model to learn how much each layer should contribute to the final representation. We hypothesize that by combining information across layers, MoL might produce a richer and more flexible acoustic representation for multimodal alignment.

The MoL mechanism operates over the hidden states produced by multiple encoder layers. Given a set of hidden states from different layers of an encoder, MoL aggregates information across the layer dimension before projecting it into the shared audio-language space. We evaluated simple aggregation strategies, such as Mean and Weighted-Sum, as well as gated MoL variants. In the gated variants, a routing network estimates layer weights conditioned on each input sample. We also evaluated variants that first refine layer representations using self-attention across layers, followed by either mean pooling or attentive pooling, denoted as Self-Att + Mean and Self-Att + Att, respectively. More details on the implementation of each of their variants can be found in Section F.

Results from experiments with MoL configurations are presented in Tables 5.19, 5.20, 5.21, and 5.22. All models were trained across three runs. We report the mean and standard deviation across runs. Additional ablations are reported in Tables 5.23 and 5.24. Finally, Tables 5.25 and 5.26 report exploratory augmentation experiments.

Main Mixture-of-Layers Results

Tables 5.19 and 5.20 report the retrieval results for MoL using CED and BEATs as audio encoders. The baseline configurations, mGTE and mGTE + LaT, correspond to models without MoL. The mGTE + MoL configuration applies MoL to the audio encoder while keeping the audio encoder trainable during contrastive learning. In contrast, mGTE + LaT + MoL applies MoL to the audio encoder under the LaT setting, where the audio backbone is frozen. The mGTE + LaT + Text MoL configuration freezes the audio encoder and applies MoL to the trainable text encoder, while mGTE + Text MoL applies MoL to the text encoder when all modules remain trainable.

For CED, MoL produced its most relevant gains when applied selectively. In the fully fine-tuned branch, mGTE + MoL achieved the best AudioCaps audio-to-text R@Avg, increasing the mGTE baseline from 72.24% to 72.82%. Although this gain is modest and remains within the standard deviation range, it suggests that aggregating multiple audio layers can preserve and exploit useful internal information from the audio encoder even when the encoder is trainable. On ClothoV2, the same tendency was slightly more visible, with audio-to-text R@Avg increasing from 36.50% to 37.16%, corresponding to a gain of 0.66 pp. The best text-to-audio configuration also improved from 28.14% to 28.69%.

		Audio-to-Text				Text-to-Audio			
		AudioCaps							
Configuration		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
CED	Baseline								
	mGTE	47.47 ± 0.56	79.49 ± 0.95	89.77 ± 0.63	72.24 ± 0.67	38.88 ± 0.44	74.13 ± 0.40	86.17 ± 0.34	66.39 ± 0.26
	mGTE + LaT	46.42 ± 0.82	78.37 ± 0.28	89.91 ± 0.37	71.57 ± 0.41	36.41 ± 0.74	72.99 ± 0.75	85.62 ± 0.61	65.01 ± 0.63
	mGTE + LaT + MoL								
	Mean	45.97 ± 0.94	77.58 ± 1.09	87.92 ± 1.13	70.49 ± 0.55	34.52 ± 0.38	71.08 ± 0.21	84.71 ± 0.48	63.44 ± 0.25
	Weighted-Sum	46.49 ± 0.80	78.63 ± 1.25	88.94 ± 0.28	71.35 ± 0.31	36.00 ± 0.69	72.01 ± 0.39	85.41 ± 0.33	64.47 ± 0.46
	Self-Att + Mean	47.43 ± 0.86	79.91 ± 0.96	89.91 ± 0.94	72.42 ± 0.87	36.51 ± 0.31	72.67 ± 0.51	85.16 ± 0.09	64.78 ± 0.26
	Self-Att + Att	46.12 ± 0.74	79.42 ± 1.24	89.57 ± 0.23	71.70 ± 0.67	36.36 ± 0.50	72.59 ± 0.38	85.85 ± 0.41	64.93 ± 0.09
	mGTE + MoL								
	Mean	46.61 ± 0.90	79.61 ± 0.75	90.10 ± 0.41	72.11 ± 0.59	37.77 ± 0.40	73.51 ± 0.59	85.89 ± 0.12	65.73 ± 0.19
	Weighted-Sum	47.09 ± 1.89	80.91 ± 1.02	89.88 ± 1.10	72.63 ± 1.31	37.65 ± 0.54	73.44 ± 0.73	85.72 ± 0.07	65.60 ± 0.40
	Self-Att + Mean	46.30 ± 0.90	78.55 ± 0.96	90.14 ± 0.72	71.67 ± 0.74	38.08 ± 0.63	73.54 ± 0.78	86.02 ± 0.25	65.88 ± 0.54
	Self-Att + Att	46.79 ± 2.44	80.61 ± 1.25	91.04 ± 0.80	72.82 ± 0.86	38.22 ± 0.27	73.77 ± 0.06	86.14 ± 0.55	66.04 ± 0.24
	mGTE + LaT + Text MoL								
	Mean	46.60 ± 0.46	79.83 ± 0.69	89.50 ± 0.28	71.98 ± 0.42	36.67 ± 0.58	72.07 ± 0.03	85.47 ± 0.39	64.73 ± 0.16
	Weighted-Sum	46.38 ± 1.29	78.86 ± 1.07	89.61 ± 1.25	71.61 ± 0.79	36.73 ± 0.29	72.40 ± 0.76	85.29 ± 0.43	64.81 ± 0.46
	Self-Att + Mean	48.48 ± 1.43	79.53 ± 0.39	89.80 ± 0.85	72.60 ± 0.24	37.33 ± 0.51	72.50 ± 0.09	85.24 ± 0.45	65.02 ± 0.26
	Self-Att + Att	48.03 ± 1.87	79.19 ± 0.56	90.33 ± 0.68	72.52 ± 0.90	37.65 ± 1.05	72.65 ± 0.21	85.48 ± 0.38	65.26 ± 0.52
	mGTE + Text MoL								
	Mean	46.72 ± 1.89	80.02 ± 0.97	90.33 ± 0.00	72.35 ± 0.95	38.12 ± 0.45	73.53 ± 0.38	85.76 ± 0.06	65.80 ± 0.07
	Weighted-Sum	46.72 ± 1.79	79.79 ± 0.23	89.84 ± 0.13	72.12 ± 0.57	37.94 ± 0.28	73.80 ± 0.32	85.82 ± 0.13	65.85 ± 0.06
	Self-Att + Mean	47.92 ± 0.59	79.53 ± 0.79	90.36 ± 0.52	72.61 ± 0.55	39.00 ± 0.71	74.23 ± 0.42	86.10 ± 0.19	66.44 ± 0.43
	Self-Att + Att	46.46 ± 3.10	79.27 ± 0.77	89.76 ± 1.22	71.83 ± 1.56	38.65 ± 0.15	74.01 ± 0.34	85.86 ± 0.36	66.17 ± 0.26
ClothoV2									
Configuration		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
CED	Baseline								
	mGTE	17.26 ± 0.80	39.65 ± 0.47	52.60 ± 1.01	36.50 ± 0.74	12.17 ± 0.45	30.02 ± 0.18	42.22 ± 0.27	28.14 ± 0.29
	mGTE + LaT	16.55 ± 0.35	37.73 ± 0.83	50.27 ± 0.80	34.85 ± 0.53	10.90 ± 0.67	28.41 ± 0.59	40.33 ± 0.86	26.55 ± 0.60
	mGTE + LaT + MoL								
	Mean	16.40 ± 0.34	36.27 ± 1.26	48.87 ± 0.48	33.84 ± 0.47	11.06 ± 0.48	29.30 ± 0.31	41.71 ± 0.31	27.35 ± 0.36
	Weighted-Sum	16.75 ± 0.63	37.99 ± 2.23	50.72 ± 1.25	35.15 ± 1.33	11.88 ± 0.76	29.92 ± 1.10	41.93 ± 1.15	27.91 ± 0.99
	Self-Att + Mean	17.58 ± 1.17	37.10 ± 0.80	50.53 ± 0.86	35.07 ± 0.73	11.20 ± 0.40	29.75 ± 0.88	41.53 ± 0.30	27.49 ± 0.51
	Self-Att + Att	18.08 ± 0.53	37.51 ± 1.69	50.59 ± 2.03	35.40 ± 1.29	11.51 ± 0.35	30.02 ± 1.03	41.93 ± 1.22	27.82 ± 0.82
	mGTE + MoL								
	Mean	18.28 ± 1.06	40.45 ± 0.87	52.76 ± 0.45	37.16 ± 0.62	11.62 ± 0.76	30.25 ± 1.56	42.53 ± 1.92	28.13 ± 1.40
	Weighted-Sum	17.67 ± 0.39	39.27 ± 0.82	51.77 ± 1.26	36.23 ± 0.36	11.45 ± 1.04	30.05 ± 1.44	42.38 ± 1.06	27.96 ± 1.16
	Self-Att + Mean	17.58 ± 0.91	38.88 ± 0.53	51.61 ± 1.63	36.02 ± 0.96	12.07 ± 0.48	30.69 ± 0.72	43.29 ± 0.55	28.69 ± 0.52
	Self-Att + Att	17.73 ± 0.64	38.98 ± 0.31	51.90 ± 0.81	36.20 ± 0.45	11.50 ± 0.43	29.45 ± 0.89	41.37 ± 0.83	27.44 ± 0.61
	mGTE + LaT + Text MoL								
	Mean	16.97 ± 0.97	38.08 ± 1.35	50.98 ± 0.77	35.34 ± 0.36	11.53 ± 0.46	29.87 ± 0.72	41.76 ± 0.90	27.72 ± 0.64
	Weighted-Sum	16.94 ± 0.58	37.96 ± 1.23	51.48 ± 1.16	35.46 ± 0.98	11.07 ± 0.28	29.13 ± 0.39	41.05 ± 0.35	27.09 ± 0.32
	Self-Att + Mean	18.25 ± 0.36	37.93 ± 0.47	50.94 ± 0.87	35.71 ± 0.20	11.07 ± 0.16	28.84 ± 0.78	40.59 ± 0.08	26.83 ± 0.28
	Self-Att + Att	16.78 ± 0.90	38.05 ± 0.78	51.61 ± 0.15	35.48 ± 0.43	11.52 ± 0.29	29.40 ± 0.35	41.26 ± 0.72	27.39 ± 0.37
	mGTE + Text MoL								
	Mean	17.48 ± 0.20	38.76 ± 1.16	52.03 ± 0.46	36.09 ± 0.34	11.85 ± 1.11	30.32 ± 1.33	42.42 ± 1.90	28.20 ± 1.44
	Weighted-Sum	17.45 ± 0.34	38.95 ± 1.76	51.87 ± 1.17	36.09 ± 0.86	11.65 ± 0.70	30.74 ± 0.74	43.26 ± 0.91	28.55 ± 0.78
	Self-Att + Mean	18.12 ± 0.43	38.79 ± 1.52	52.47 ± 0.64	36.46 ± 0.54	11.46 ± 0.35	29.50 ± 0.61	41.79 ± 0.94	27.58 ± 0.61
	Self-Att + Att	17.92 ± 1.58	38.28 ± 1.24	51.87 ± 1.80	36.02 ± 1.51	11.87 ± 0.24	30.67 ± 0.53	42.67 ± 0.58	28.40 ± 0.39

Table 5.19: Recall@k results for Audio-to-Text and Text-to-Audio retrieval on AudioCaps and ClothoV2 for PALMA configurations using the CED audio encoder. Values are grouped according to the main configuration family and reported as mean ± standard deviation. **Bold** values indicate the best result within each configuration family, while underlined values indicate the second-best result within the same family. Values highlighted in **blue** indicate the best overall result for each metric and dataset among the reported CED configurations.

The BEATs results show a similar, but less favorable, pattern for the locked-audio setting. With BEATs, the mGTE baseline reached 71.93% audio-to-text R@Avg on AudioCaps, while mGTE + LaT decreased this value to 70.44%. Adding MoL to the locked-audio branch partially recovered performance, with the best mGTE + LaT + MoL configuration reaching 70.97%. However, the fully fine-tuned mGTE + MoL branch

		Audio-to-Text				Text-to-Audio			
		AudioCaps							
Configuration		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
BEATs	Baseline								
	mGTE	47.58 ± 0.85	78.93 ± 0.66	89.28 ± 0.52	71.93 ± 0.53	38.41 ± 0.15	73.39 ± 0.83	85.92 ± 0.35	65.90 ± 0.37
	mGTE + LaT	45.18 ± 1.08	77.50 ± 0.63	88.64 ± 0.59	70.44 ± 0.67	33.85 ± 0.28	68.98 ± 0.97	82.52 ± 0.22	61.78 ± 0.35
	mGTE + LaT + MoL								
	Mean	42.71 ± 1.63	77.50 ± 0.85	88.27 ± 0.23	69.49 ± 0.82	31.97 ± 0.60	67.05 ± 0.43	81.69 ± 0.43	60.24 ± 0.48
	Weighted-Sum	44.51 ± 1.45	77.50 ± 0.12	88.49 ± 0.75	70.17 ± 0.46	32.83 ± 0.31	68.68 ± 0.43	82.48 ± 0.50	61.33 ± 0.22
	Self-Att + Mean	45.93 ± 1.19	78.03 ± 0.75	88.12 ± 0.45	70.69 ± 0.56	33.59 ± 0.38	68.76 ± 0.53	82.65 ± 0.13	61.66 ± 0.26
	Self-Att + Att	46.76 ± 0.72	78.10 ± 0.77	88.04 ± 1.26	70.97 ± 0.83	33.40 ± 0.56	68.80 ± 0.12	82.90 ± 0.10	61.70 ± 0.17
	mGTE + MoL								
	Mean	48.52 ± 0.96	80.35 ± 0.72	89.47 ± 0.90	72.78 ± 0.41	38.48 ± 0.20	73.60 ± 0.51	85.62 ± 0.22	65.90 ± 0.31
	Weighted-Sum	47.88 ± 1.04	79.64 ± 0.30	89.32 ± 1.07	72.28 ± 0.38	38.40 ± 0.45	73.78 ± 0.68	85.68 ± 0.92	65.95 ± 0.29
	Self-Att + Mean	47.92 ± 1.68	79.23 ± 0.36	89.99 ± 1.19	72.38 ± 0.95	39.01 ± 0.12	73.98 ± 0.38	86.16 ± 0.38	66.38 ± 0.22
	Self-Att + Att	46.91 ± 1.62	79.01 ± 1.23	88.64 ± 0.97	71.52 ± 0.46	38.27 ± 0.21	74.05 ± 0.26	85.79 ± 0.36	66.03 ± 0.20
	mGTE + LaT + Text MoL								
	Mean	44.62 ± 1.65	78.10 ± 1.02	88.60 ± 0.45	70.44 ± 0.55	33.83 ± 0.29	69.09 ± 0.25	82.45 ± 0.41	61.79 ± 0.19
	Weighted-Sum	44.54 ± 1.76	78.07 ± 1.95	88.30 ± 0.45	70.31 ± 1.24	33.74 ± 0.33	68.83 ± 0.46	82.29 ± 0.24	61.62 ± 0.24
	Self-Att + Mean	44.96 ± 1.02	77.47 ± 1.24	87.21 ± 0.47	69.88 ± 0.64	33.75 ± 0.28	69.10 ± 0.43	82.76 ± 0.41	61.87 ± 0.33
	Self-Att + Att	45.44 ± 0.79	76.90 ± 1.37	88.26 ± 0.91	70.20 ± 0.97	33.79 ± 0.80	69.28 ± 0.50	82.88 ± 0.22	61.98 ± 0.45
	mGTE + Text MoL								
	Mean	47.32 ± 1.17	79.83 ± 0.62	90.25 ± 0.80	72.47 ± 0.62	38.75 ± 0.80	74.12 ± 0.35	85.90 ± 0.23	66.25 ± 0.45
	Weighted-Sum	46.31 ± 1.83	78.29 ± 1.07	88.49 ± 0.07	71.03 ± 0.47	38.22 ± 0.18	73.36 ± 0.22	85.50 ± 0.16	65.69 ± 0.05
	Self-Att + Mean	47.40 ± 0.40	79.04 ± 0.64	89.31 ± 0.68	71.91 ± 0.45	38.80 ± 0.20	73.64 ± 0.21	85.99 ± 0.09	66.14 ± 0.09
	Self-Att + Att	47.81 ± 1.76	79.15 ± 1.32	89.46 ± 0.34	72.14 ± 0.54	38.41 ± 0.43	74.01 ± 0.41	86.33 ± 0.52	66.25 ± 0.43
ClothoV2									
Configuration		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
BEATs	Baseline								
	mGTE	18.18 ± 0.58	38.72 ± 1.30	51.74 ± 0.45	36.22 ± 0.52	11.22 ± 0.58	28.40 ± 0.10	40.15 ± 0.57	26.59 ± 0.32
	mGTE + LaT	16.17 ± 0.69	36.91 ± 1.21	49.57 ± 1.63	34.22 ± 1.13	10.27 ± 0.16	26.90 ± 0.27	37.98 ± 0.62	25.05 ± 0.32
	mGTE + LaT + MoL								
	Mean	15.88 ± 0.42	37.45 ± 0.95	50.21 ± 0.92	34.51 ± 0.65	10.47 ± 0.30	27.44 ± 0.22	39.12 ± 0.32	25.68 ± 0.08
	Weighted-Sum	16.36 ± 0.75	36.46 ± 0.44	49.06 ± 0.69	33.96 ± 0.14	10.24 ± 0.51	26.58 ± 0.33	37.92 ± 0.28	24.91 ± 0.32
	Self-Att + Mean	16.30 ± 0.45	37.10 ± 0.62	49.92 ± 0.63	34.44 ± 0.55	10.83 ± 0.23	27.60 ± 0.62	38.82 ± 0.36	25.75 ± 0.36
	Self-Att + Att	16.49 ± 0.45	36.53 ± 0.56	48.96 ± 0.77	33.99 ± 0.24	10.25 ± 0.23	26.84 ± 0.82	37.93 ± 0.75	25.01 ± 0.58
	mGTE + MoL								
	Mean	18.41 ± 0.68	37.22 ± 0.35	51.74 ± 0.68	35.79 ± 0.40	11.02 ± 0.35	28.84 ± 0.63	40.96 ± 0.27	26.94 ± 0.41
	Weighted-Sum	16.65 ± 0.91	37.58 ± 0.53	52.34 ± 0.48	35.52 ± 0.29	10.70 ± 0.18	29.02 ± 0.08	41.07 ± 0.76	26.93 ± 0.33
	Self-Att + Mean	17.67 ± 0.98	38.88 ± 0.52	52.03 ± 0.56	36.19 ± 0.49	11.16 ± 0.84	29.06 ± 0.95	40.73 ± 0.95	26.98 ± 0.91
	Self-Att + Att	17.70 ± 0.17	37.74 ± 0.58	51.26 ± 1.34	35.57 ± 0.58	10.99 ± 1.15	28.42 ± 1.42	40.04 ± 1.36	26.48 ± 1.24
	mGTE + LaT + Text MoL								
	Mean	15.28 ± 0.44	36.36 ± 0.48	49.79 ± 0.52	33.81 ± 0.39	10.12 ± 0.27	26.99 ± 0.63	38.25 ± 0.57	25.12 ± 0.49
	Weighted-Sum	15.63 ± 0.11	36.11 ± 0.89	48.90 ± 0.10	33.55 ± 0.29	10.42 ± 0.37	26.64 ± 0.45	37.91 ± 0.62	24.99 ± 0.38
	Self-Att + Mean	16.08 ± 0.53	35.92 ± 0.72	48.07 ± 0.15	33.35 ± 0.24	10.35 ± 0.10	26.65 ± 0.43	37.68 ± 0.70	24.90 ± 0.38
	Self-Att + Att	15.63 ± 0.38	35.73 ± 1.39	47.97 ± 1.08	33.11 ± 0.77	9.94 ± 0.24	26.33 ± 0.40	37.42 ± 0.71	24.56 ± 0.44
	mGTE + Text MoL								
	Mean	17.61 ± 0.19	38.44 ± 1.11	50.56 ± 0.39	35.54 ± 0.53	11.37 ± 0.15	28.81 ± 0.62	40.32 ± 0.83	26.83 ± 0.41
	Weighted-Sum	18.76 ± 0.10	38.79 ± 0.47	51.74 ± 1.29	36.43 ± 0.45	11.78 ± 0.41	29.19 ± 1.23	41.07 ± 0.87	27.35 ± 0.83
	Self-Att + Mean	16.91 ± 1.13	38.59 ± 0.06	51.00 ± 0.79	35.50 ± 0.58	11.23 ± 0.19	28.92 ± 0.35	40.78 ± 0.61	26.98 ± 0.38
	Self-Att + Att	17.58 ± 1.39	38.69 ± 1.70	52.22 ± 2.61	36.16 ± 1.87	11.60 ± 0.18	29.25 ± 1.22	40.82 ± 0.93	27.23 ± 0.77

Table 5.20: Recall@k results for Audio-to-Text and Text-to-Audio retrieval on AudioCaps and ClothoV2 for PALMA configurations using the BEATs audio encoder. Values are grouped according to the main configuration family and reported as mean ± standard deviation. **Bold** values indicate the best result within each configuration family, while underlined values indicate the second-best result within the same family. Values highlighted in **blue** indicate the best overall result for each metric and dataset among the reported BEATs configurations.

remained stronger, reaching 72.78% on AudioCaps. On ClothoV2, BEATs also benefited more from fully fine-tuned MoL or Text MoL than from locked-audio MoL. This indicates that, although MoL is useful for both audio encoders, the interaction between LaT and audio-side aggregation is more favorable for CED than for BEATs.

A complementary pattern appears in the LaT branch for CED. On AudioCaps,

	Configuration	Classification	
		ESC-50	UrbanSound8K
CED	<i>Baseline</i>		
	mGTE	71.02 $\pm$ 3.42	67.45 $\pm$ 2.53
	mGTE + LaT	80.17 $\pm$ 0.58	70.80 $\pm$ 4.34
	<i>mGTE + LaT + MoL</i>		
	Mean	69.25 $\pm$ 9.05	62.15 $\pm$ 5.11
	Weighted-Sum	77.02 $\pm$ 1.30	65.63 $\pm$ 5.61
	Self-Att + Mean	80.37 $\pm$ 1.99	70.29 $\pm$ 1.71
	Self-Att + Att	<u>78.68 $\pm$ 3.76</u>	<u>69.47 $\pm$ 1.88</u>
	<i>mGTE + MoL</i>		
	Mean	71.77 $\pm$ 1.18	67.53 $\pm$ 3.78
	Weighted-Sum	68.48 $\pm$ 1.87	66.40 $\pm$ 6.70
	Self-Att + Mean	73.32 $\pm$ 0.64	69.05 $\pm$ 1.34
	Self-Att + Att	68.50 $\pm$ 1.90	<u>68.05 $\pm$ 2.87</u>
	<i>mGTE + LaT + Text MoL</i>		
	Mean	<u>82.22 $\pm$ 2.37</u>	73.16 $\pm$ 2.04
	Weighted-Sum	82.82 $\pm$ 2.24	70.14 $\pm$ 1.40
	Self-Att + Mean	81.43 $\pm$ 2.32	69.11 $\pm$ 2.55
	Self-Att + Att	80.33 $\pm$ 0.46	<u>71.57 $\pm$ 1.17</u>
	<i>mGTE + Text MoL</i>		
	Mean	<u>75.85 $\pm$ 1.91</u>	69.17 $\pm$ 2.95
	Weighted-Sum	77.37 $\pm$ 0.92	<u>68.72 $\pm$ 2.44</u>
	Self-Att + Mean	74.60 $\pm$ 1.28	67.15 $\pm$ 1.04
	Self-Att + Att	75.28 $\pm$ 3.66	68.54 $\pm$ 2.23

Table 5.21: Classification accuracy on ESC-50 and UrbanSound8K for PALMA configurations using the CED audio encoder. The table is filtered to the same configuration families reported in the retrieval analysis. Values are reported as mean $\pm$ standard deviation. **Bold** values indicate the best result within each configuration family, while underlined values indicate the second-best result within the same family. Values highlighted in **blue** indicate the best overall result for each dataset among the reported CED configurations.

CED + mGTE + LaT obtained an audio-to-text R@Avg of 71.57%, while CED + mGTE + LaT + MoL with Self-Att + Mean reached 72.42%. This represents an improvement of 0.85 pp over LaT alone, recovering part of the retrieval performance lost by locking the audio encoder. On ClothoV2, the same configuration improved audio-to-text R@Avg from 34.85% to 35.07% and text-to-audio R@Avg from 26.55% to 27.49%. These gains are modest, but they are consistent across MoL variations.

The classification results in Tables 5.21 and 5.22 show that the strongest classification configurations are not necessarily the strongest retrieval configurations. For CED, the best classification results were obtained with LaT + Text MoL, reaching 82.82% on ESC-50 and 73.16% on UrbanSound8K, compared to simpler aggregation mechanisms like Mean pooling and Weighted-Sum. For BEATs, the best ESC-50 result was obtained with mGTE + Text MoL, reaching 77.88%, while the best UrbanSound8K result was ob-

	Configuration	Classification	
		ESC-50	UrbanSound8K
BEATs	<i>Baseline</i>		
	mGTE	70.35 $\pm$ 6.73	66.52 $\pm$ 3.79
	mGTE + LaT	71.98 $\pm$ 1.46	64.85 $\pm$ 1.05
	<i>mGTE + LaT + MoL</i>		
	Mean	<u>70.75</u> $\pm$ 0.82	59.99 $\pm$ 4.90
	Weighted-Sum	66.30 $\pm$ 2.08	66.69 $\pm$ 1.62
	Self-Att + Mean	70.07 $\pm$ 0.30	65.41 $\pm$ 4.13
	Self-Att + Att	71.32 $\pm$ 2.51	<u>66.04</u> $\pm$ 3.71
	<i>mGTE + MoL</i>		
	Mean	68.53 $\pm$ 1.64	<u>70.14</u> $\pm$ 3.96
	Weighted-Sum	70.68 $\pm$ 2.93	70.68 $\pm$ 2.24
	Self-Att + Mean	<u>72.02</u> $\pm$ 3.72	65.44 $\pm$ 2.98
	Self-Att + Att	72.33 $\pm$ 4.08	67.70 $\pm$ 3.44
	<i>mGTE + LaT + Text MoL</i>		
	Mean	73.90 $\pm$ 2.70	72.37 $\pm$ 3.22
	Weighted-Sum	71.17 $\pm$ 1.73	66.98 $\pm$ 5.42
	Self-Att + Mean	72.70 $\pm$ 0.13	<u>71.05</u> $\pm$ 2.44
	Self-Att + Att	<u>73.83</u> $\pm$ 0.94	67.91 $\pm$ 2.32
	<i>mGTE + Text MoL</i>		
	Mean	<u>76.90</u> $\pm$ 1.77	<u>70.64</u> $\pm$ 0.46
	Weighted-Sum	76.68 $\pm$ 0.91	68.16 $\pm$ 4.48
	Self-Att + Mean	77.88 $\pm$ 3.13	70.45 $\pm$ 0.93
	Self-Att + Att	75.75 $\pm$ 3.18	72.07 $\pm$ 2.21

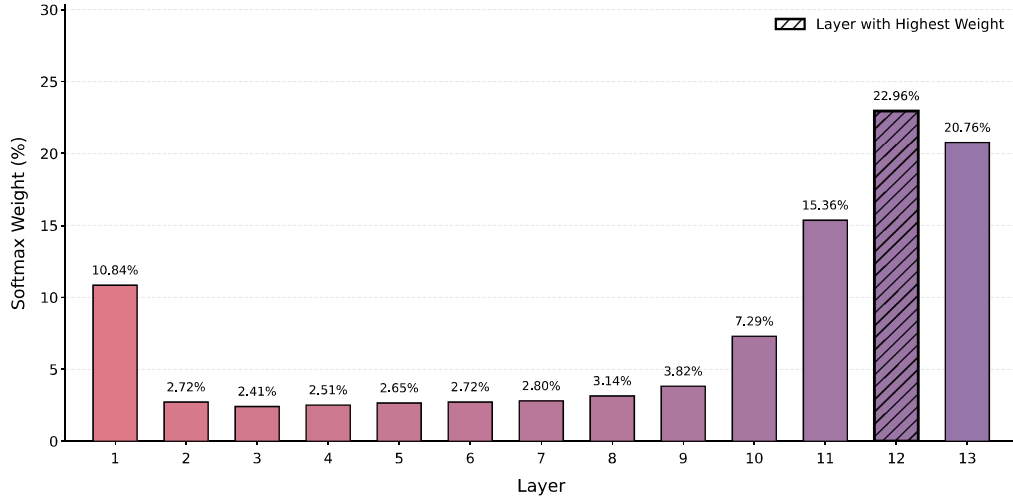
Table 5.22: Classification accuracy on ESC-50 and UrbanSound8K for PALMA configurations using the BEATs audio encoder. The table is filtered to the same configuration families reported in the retrieval analysis. Values are reported as mean $\pm$ standard deviation. **Bold** values indicate the best result within each configuration family, while underlined values indicate the second-best result within the same family. Values highlighted in **blue** indicate the best overall result for each dataset among the reported BEATs configurations.

tained with mGTE + LaT + Text MoL, reaching 72.37%. These results suggest that text-side MoL can be effective for zero-shot classification, especially when the task depends on matching audio embeddings to short class-level prompts.

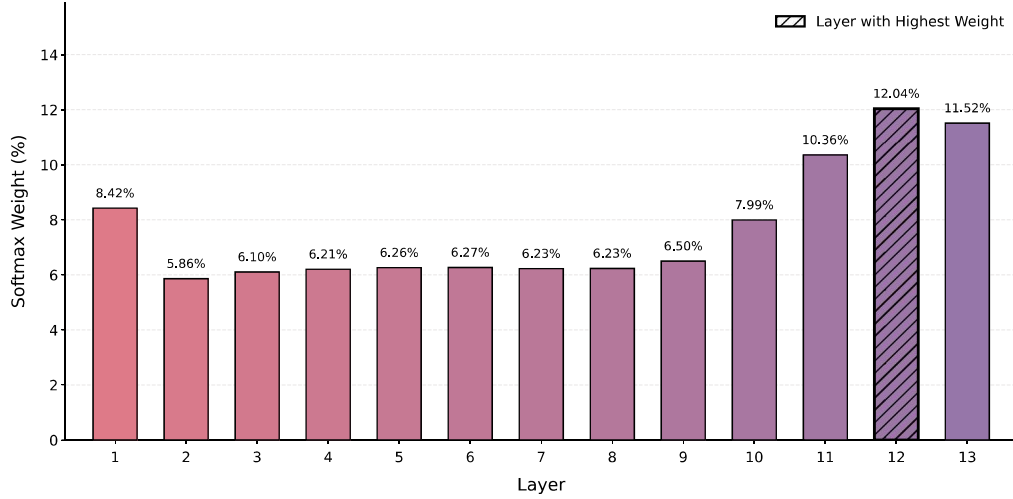
Layer-Weight Importance Exploration

To better understand the behavior of MoL and provide further insight into its performance across different scenarios, we analyzed the layer weights learned by the Weighted-Sum configuration. This configuration was chosen specifically because it offers a direct interpretability mechanism in which the learned softmax weights represent the relative contribution of each encoder layer to the aggregated representation before projection into the shared audio-text space. Therefore, this analysis allows us to examine

which layers are emphasized during audio-language alignment and how this emphasis changes depending on whether the encoder is frozen or fine-tuned during training.



(a) *CED + mGTE + LaT + MoL*



(b) *CED + mGTE + MoL*

Figure 5.11: Layer-weight distributions learned by the Weighted-Sum MoL configuration for the CED audio encoder. The top plot (a) shows the LaT setting, in which the audio encoder is frozen, while the bottom plot (b) shows the full fine-tuning setting, in which the audio encoder is updated during training.

Figures 5.11 and 5.12 show that deeper layers receive the largest weights in most audio-side configurations. For CED under LaT, shown in Figure 5.11 (a), the distribution becomes concentrated in the final layers, with layers 12 and 13 receiving the largest weights. BEATs under LaT presents an even stronger concentration, as shown in Figure 5.12 (a), where the final layer receives the dominant weight.

An interesting difference between the two audio encoders is observed in the contribution of the first layer. In CED, the first layer receives a higher weight than

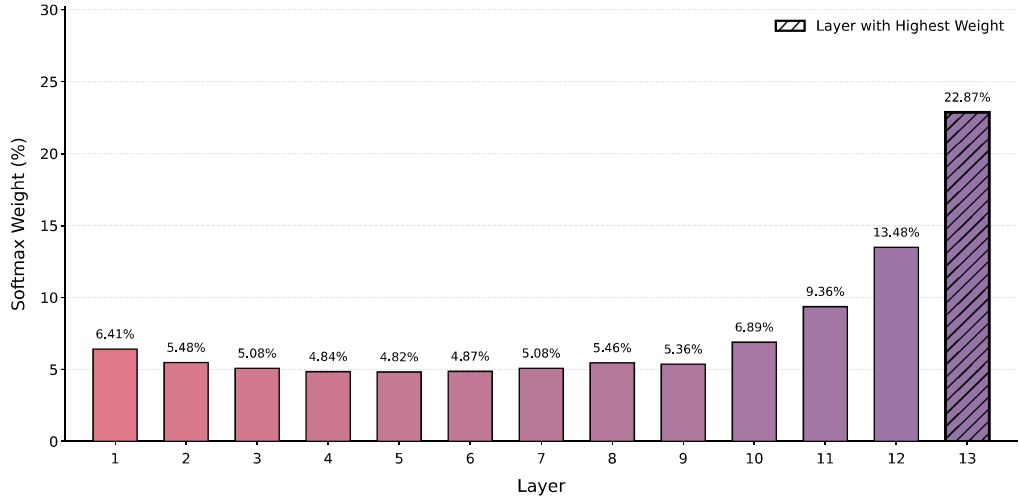
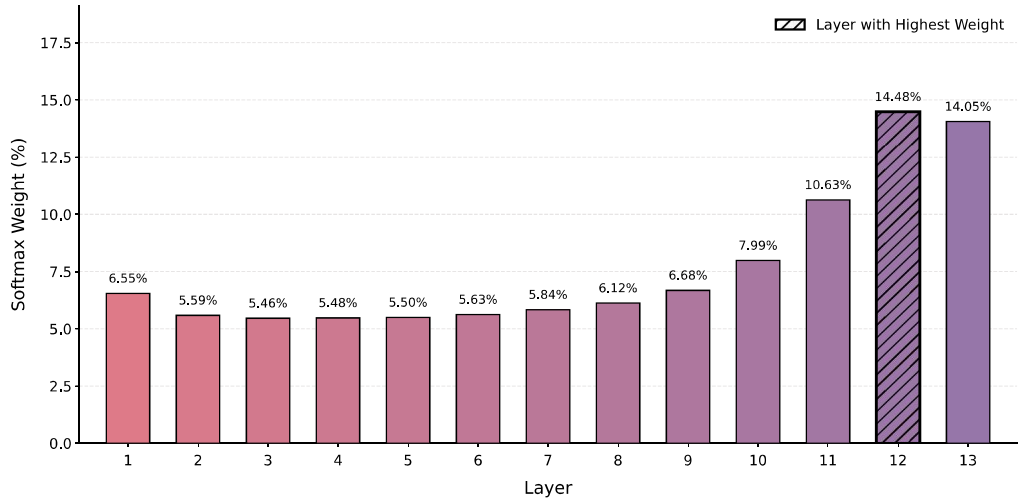
(a) *BEATs + mGTE + LaT + MoL*(b) *BEATs + mGTE + MoL*

Figure 5.12: Layer-weight distributions learned by the Weighted-Sum MoL configuration for the BEATs audio encoder. The top plot (a) shows the LaT setting, in which the audio encoder is frozen, while the bottom plot (b) shows the full fine-tuning setting, in which the audio encoder is updated during training.

several intermediate layers, suggesting that early representations still contain information useful for audio-language alignment. This pattern is not observed in BEATs, where the weights are more strongly biased toward the final layers, especially in the LaT setting. This suggests that, for BEATs, the final representations may already contain most of the information needed for the alignment task, reducing the potential benefit of aggregating earlier layers.

This difference may help explain why applying MoL to BEATs under LaT did not improve performance as consistently as it did for CED. Since BEATs assigns most of the importance to the last layers, especially the final layer, MoL has fewer

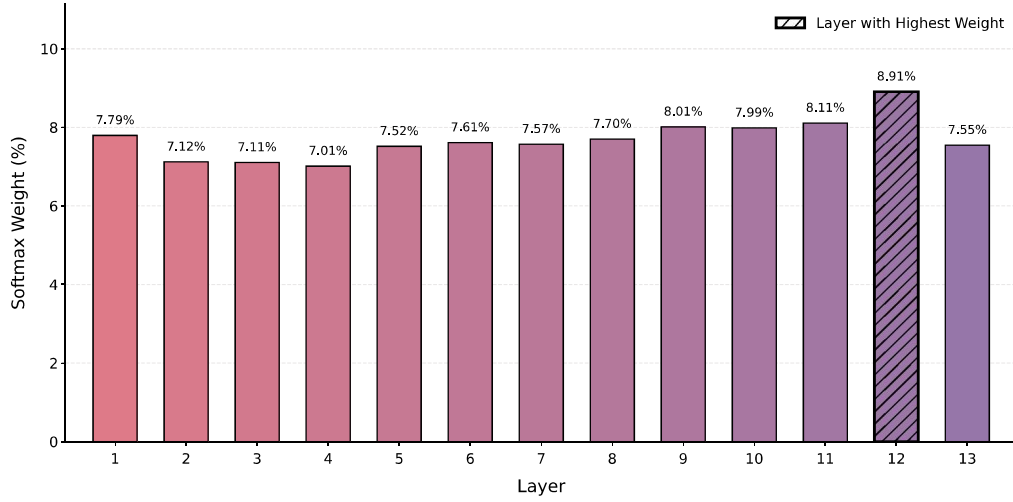


Figure 5.13: *Layer-weight distribution learned by the Weighted-Sum MoL configuration for the mGTE text encoder. The weights indicate the relative contribution of each text-encoder layer to the final representation used for audio-language alignment.*

complementary representations to exploit across depth. In contrast, CED assigns non-negligible importance to both early and deeper layers, indicating that useful information for audio-language alignment is available at different levels of the encoder. Therefore, CED appears to benefit more from MoL because the aggregation mechanism can combine early-layer information with the more task-relevant semantic representations found in the final layers.

This observation might also help to explain why MoL produced only moderate gains overall. The learned weights are not uniformly distributed across layers. Instead, they remain strongly biased toward the final layers, particularly in the LaT setting. This suggests that, for these audio encoders, the internal hierarchy may be less evenly exploitable for audio-language alignment than what has been observed in other domains, such as speech processing, natural language processing, and computer vision, where different layers often encode more clearly differentiated levels of information [16, 146, 175]. As a result, MoL might still provide useful flexibility, but its potential may be limited when most of the alignment-relevant information is already heavily concentrated in a few layers.

In contrast, when the audio encoder is fully fine-tuned, the weights are more evenly distributed across layers, as can be seen in Figures 5.11 and 5.12 (b). This suggests that, when the audio encoder is frozen, the aggregation module learns to rely more strongly on a small set of high-level layers that are already useful for the target alignment task. When the encoder is trainable, however, the model can reshape the internal representations during training, leading to a more distributed use of layers. This may

explain why the learned weights become more concentrated in the locked-audio setting.

Figure 5.13 shows that the mGTE text encoder presents a more balanced layer-weight distribution than its audio counterparts. Suggesting that language information useful for audio-language alignment is not concentrated in a few layers at the end of the network, but is more evenly distributed across the depth of the text encoder. This behavior may help explain the more pronounced improvements observed in the classification task when Text MoL is applied. Since relevant textual information is available across multiple intermediate and deeper layers, the MoL mechanism can more effectively aggregate complementary representations from different levels of abstraction. As a result, text-side MoL appears to provide a stronger benefit for class-level transfer.

These analyses and results help support the main motivation behind MoL. The final layer is important, but it is not the only useful source of information. Earlier and intermediate layers still retain non-negligible information, suggesting that the model might benefit from combining multiple abstraction levels. In the audio encoder, these layers may preserve acoustic and event-level information that complements the more semantic information encoded in deeper layers. Therefore, MoL provides a mechanism to exploit the internal hierarchy of pretrained encoders rather than assuming that the final layer alone is sufficient.

Dual-Encoder MoL and Projection-Only Ablations

Tables 5.23 and 5.24 provide additional ablations with simultaneous audio and text MoL application and with projection-only training. In the projection-only setting, both encoders are frozen, and only the projection layers are tuned. This setup obtained much lower retrieval performance than the main configurations. On AudioCaps, mGTE + Proj. only reached 61.98% audio-to-text R@Avg, compared with 72.24% for the fully fine-tuned mGTE baseline and 72.42% for CED + mGTE + LaT + MoL with Self-Att + Mean. Showing that projection learning alone, using frozen features from both modalities, is insufficient to produce a strong audio-language alignment latent space.

Adding MoL to both encoders under projection-only training also did not recover the lost retrieval performance. The dual-encoder MoL projection-only configuration reached 61.10% audio-to-text R@Avg on AudioCaps and 27.59% on ClothoV2, remaining below the projection-only baseline in these directions. This indicates that MoL is not simply a plug-in replacement for encoder adaptation. Instead, it is most useful when the model can learn meaningful aggregation behavior together with either a trainable branch or a carefully designed locked-audio strategy.

The results also indicate that applying MoL to both audio and text simultaneously did not provide the best retrieval performance. The Audio MoL + Text MoL configura-

		Audio-to-Text				Text-to-Audio			
		AudioCaps							
Configuration		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
CED	mGTE + A-MoL (SA+Att) + T-MoL (SA+Att)	46.49 ± 1.31	78.97 ± 0.85	89.42 ± 0.49	71.63 ± 0.86	37.01 ± 0.36	72.76 ± 0.73	85.76 ± 0.62	65.18 ± 0.55
	mGTE + Proj. only	36.00 ± 0.60	68.54 ± 0.68	81.40 ± 0.91	61.98 ± 0.68	29.53 ± 0.31	65.27 ± 0.43	79.71 ± 0.10	58.17 ± 0.19
	mGTE + A-MoL (SA+Att) + T-MoL (SA+Att) + Proj. only	34.57 ± 0.75	67.60 ± 0.78	81.14 ± 1.07	61.10 ± 0.48	29.27 ± 0.19	65.77 ± 0.08	80.26 ± 0.12	58.44 ± 0.12
		ClothoV2							
Configuration		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
CED	mGTE + A-MoL (SA+Att) + T-MoL (SA+Att)	17.48 ± 0.77	37.64 ± 0.78	51.20 ± 0.58	35.44 ± 0.53	11.57 ± 0.26	29.65 ± 0.30	41.33 ± 0.36	27.52 ± 0.14
	mGTE + Proj. only	12.44 ± 0.29	30.50 ± 1.13	42.62 ± 0.72	28.51 ± 0.58	10.87 ± 0.31	29.00 ± 0.35	41.05 ± 0.10	26.97 ± 0.21
	mGTE + A-MoL (SA+Att) + T-MoL (SA+Att) + Proj. only	12.63 ± 0.26	29.44 ± 0.98	40.70 ± 0.83	27.59 ± 0.51	10.65 ± 0.42	27.78 ± 0.17	39.39 ± 0.39	25.94 ± 0.30

Table 5.23: Retrieval ablation for PALMA using the CED audio encoder and mGTE text encoder; comparing simultaneous MoL aggregation in both encoders, projection-only training, and projection-only training with dual-encoder MoL. Values are reported as mean $\pm$ standard deviation. Values highlighted in blue indicate the best result for each metric within each dataset. A-MoL denotes Audio MoL, T-MoL denotes Text MoL, SA denotes self-attention, Att denotes attentive pooling, and Proj. only denotes training only the projection layers.

Configuration		ESC-50	UrbanSound8K
CED	mGTE + A-MoL (SA+Att) + T-MoL (SA+Att)	79.70 ± 0.23	69.43 ± 3.60
	mGTE + Proj. only	69.42 ± 1.11	69.54 ± 0.77
	mGTE + A-MoL (SA+Att) + T-MoL (SA+Att) + Proj. only	71.23 ± 1.22	70.83 ± 0.76

Table 5.24: Classification ablation for PALMA using the CED audio encoder and mGTE text encoder; comparing simultaneous MoL aggregation in both encoders, projection-only training, and projection-only training with dual-encoder MoL. Values are reported as mean $\pm$ standard deviation. Values highlighted in blue indicate the best result for each dataset.

tion remained below the strongest single-side aggregation settings on both AudioCaps and ClothoV2.

Exploratory Data Augmentation Analysis

Finally, Tables 5.25 and 5.26 report the exploratory augmentation analysis. We conducted a low-cost exploratory analysis of data augmentation in PALMA. To avoid substantially increasing the computational cost of the ablation study, each augmentation configuration was trained with a single run. Therefore, these results are not intended to provide a definitive comparison, but rather to estimate whether the tested augmentation strategies provide consistent evidence of being beneficial for PALMA. The analysis is done for the selected CED + mGTE + LaT + MoL configuration with Self-Att + Mean Pooling. The evaluated strategies include SpecAug, Random truncation (RT), their combination, and text masking as used in AudioSetCaps [10]. In the text-masking experiments, [MASK] refers to replacing selected words with the literal string [MASK],

while <mask> refers to replacing selected words with the actual mask token used by the tokenizer.

The retrieval results show that no individual augmentation strategy, nor any combination of strategies, consistently outperformed the non-augmented configuration across datasets, retrieval directions, and metrics. On AudioCaps, the original configuration without augmentation achieved the best audio-to-text R@Avg, while SpecAug obtained the best text-to-audio R@Avg. On ClothoV2, SpecAug achieved the best audio-to-text R@Avg, whereas text masking with [MASK] produced the best text-to-audio R@Avg. However, these improvements are small and metric-dependent, suggesting that augmentation can benefit specific retrieval directions but does not provide a uniformly stronger training strategy.

Augmentation	Audio-to-Text				Text-to-Audio			
	AudioCaps							
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
<i>CED + mGTE + LaT + A-MoL (SA+Mean)</i>								
None	47.43	79.91	89.91	72.42	36.51	72.67	85.16	64.78
SpecAug	46.79	80.09	89.76	72.22	36.69	73.32	85.51	65.17
RT	46.23	79.30	90.10	71.88	36.51	72.85	85.29	64.88
SpecAug + RT	47.81	78.97	89.76	72.18	36.29	72.78	85.78	64.95
Text Masking [MASK]	47.36	78.52	88.64	71.50	36.24	72.67	85.08	64.66
Text Masking <mask>	46.68	78.85	88.86	71.47	36.36	72.08	84.97	64.47
SpecAug + RT + Text Masking <mask>	45.56	79.53	89.65	71.58	36.15	72.80	85.69	64.88
ClothoV2								
Augmentation	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
<i>CED + mGTE + LaT + A-MoL (SA+Mean)</i>								
None	17.58	37.10	50.53	35.07	11.20	29.75	41.53	27.49
SpecAug	18.18	37.80	51.00	35.66	11.41	29.76	41.26	27.48
RT	16.27	36.75	50.24	34.42	11.02	28.52	40.21	26.58
SpecAug + RT	17.32	38.09	50.91	35.44	11.00	29.47	42.01	27.50
Text Masking [MASK]	16.36	37.22	50.14	34.58	11.71	29.67	41.97	27.78
Text Masking <mask>	17.61	37.42	51.20	35.41	11.08	29.00	40.50	26.86
SpecAug + RT + Text Masking <mask>	16.84	34.83	47.75	33.14	10.26	26.93	37.80	25.00

Table 5.25: Full retrieval analysis of data augmentation strategies for PALMA using the CED audio encoder, mGTE text encoder, LaT, and Audio MoL with Self-Att+Mean aggregation. SigLIP-based configurations are omitted from this table. Values are reported without standard deviation. Values highlighted in blue indicate the best result for each metric and dataset, while underlined values indicate the second-best result. A-MoL denotes Audio MoL, SA denotes self-attention, SpecAug denotes spectrogram augmentation, RT denotes random truncation, and Text Masking denotes token masking applied to the textual input.

The classification results show an even clearer limitation. The non-augmented configuration obtained the best results on both ESC-50 and UrbanSound8K, indicating that the tested augmentations did not improve zero-shot classification transfer. In partic-

Augmentation	Classification	
	ESC-50	UrbanSound8K
<i>CED + mGTE + LaT + A-MoL (SA+Mean)</i>		
None	80.37	70.29
SpecAug	76.45	<u>66.18</u>
RT	79.05	56.86
SpecAug + RT	<u>79.70</u>	58.88
Text Masking [MASK]	79.50	66.20
Text Masking <mask>	77.75	65.65

Table 5.26: *Classification analysis of data augmentation strategies for PALMA using the CED audio encoder, mGTE text encoder, LaT, and Audio MoL with Self-Att+Mean aggregation. SigLIP-based configurations are omitted from this table. Values are reported without standard deviation. Values highlighted in blue indicate the best result for each dataset, while underlined values indicate the second-best result.*

ular, RT and SpecAug + RT reduced UrbanSound8K performance noticeably, suggesting that augmentations that modify the temporal structure of the audio may harm class-level transfer in this setting. Therefore, although augmentation remains a promising direction for future exploration in the PALMA framework, the current results do not justify replacing the non-augmented configuration as the main configuration.

This behavior contrasts with the results observed for CACARA, where data augmentation provided clearer benefits. One possible explanation is that PALMA already includes an additional regularizing mechanism through MoL. Since the MoL module must learn how to combine representations from different encoder depths, especially in variants that use self-attention and sample-dependent aggregation, it may already encourage the model to use more robust internal representations. Under this interpretation, external augmentation provides less additional benefit for PALMA than it did for CACARA.

Overall, this phase shows that MoL can modestly improve PALMA when applied selectively. Although Text MoL achieved the strongest classification results, CED + mGTE + LaT + MoL with Self-Att + Mean was selected for the following phase because it offers the best balance between efficiency, retrieval performance, and classification transfer. This configuration keeps the audio encoder frozen, preserving the advantages of LaT, while using MoL to recover part of the retrieval performance lost by locking the acoustic backbone. The layer-weight analysis further supports this choice, showing that MoL learns to emphasize the most relevant pretrained audio layers under LaT, especially deeper layers, while still preserving contributions from other depths.

5.3.6 Training Data Combination

This phase investigates the effect of training-data composition on PALMA. After selecting the configuration CED + mGTE + LaT + MoL with Self-Att + Mean

in the previous phase, we analyze how different dataset combinations affect retrieval and zero-shot classification performance. The goal is to identify whether adding more data consistently improves performance or whether the final training mixture depends more strongly on dataset compatibility, caption quality, and domain coverage.

All configurations in this phase include AudioCaps (AC) and ClothoV2 (C), since these datasets contain high-quality human-written captions and are directly related to the retrieval evaluation setting. Starting from this shared base, we evaluate different combinations with Auto-ACD (AA), WavCaps (WC), and AudioSetCaps (ASC), allowing us to measure the contribution of each additional data source to the final training subset.

Each configuration was trained three times, and Tables 5.27 and 5.28 report the mean and standard deviation across runs.

Training Data	Audio-to-Text				Text-to-Audio			
	AudioCaps							
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
AA + AC + C	45.93 ± 0.51	79.68 ± 1.40	89.95 ± 0.94	71.85 ± 0.57	35.18 ± 0.51	71.77 ± 0.39	85.42 ± 0.23	64.12 ± 0.34
WC + AC + C	44.99 ± 0.39	78.33 ± 0.26	89.80 ± 0.46	71.04 ± 0.19	34.98 ± 0.44	70.17 ± 0.33	83.76 ± 0.63	62.97 ± 0.46
AA + WC + AC + C	48.97 ± 0.72	81.96 ± 0.40	90.82 ± 0.51	73.92 ± 0.42	37.14 ± 0.69	73.16 ± 0.73	86.27 ± 0.04	<u>65.52 ± 0.46</u>
ASC + WC + AC + C	50.24 ± 1.07	81.02 ± 0.68	90.92 ± 0.32	74.06 ± 0.63	38.48 ± 0.13	73.78 ± 0.38	86.72 ± 0.36	66.32 ± 0.25
ASC + AA + AC + C	39.33 ± 0.91	70.15 ± 1.24	82.94 ± 1.01	64.14 ± 0.95	30.69 ± 0.60	65.06 ± 0.28	79.92 ± 0.40	58.56 ± 0.39
ASC + AA + WC + AC + C	39.22 ± 0.73	72.14 ± 0.71	83.84 ± 0.36	65.07 ± 0.12	31.05 ± 0.64	65.88 ± 0.75	80.16 ± 0.15	59.03 ± 0.39

Training Data	ClothoV2							
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
AA + AC + C	16.30 ± 0.69	36.14 ± 1.36	49.06 ± 1.37	33.83 ± 0.90	10.60 ± 0.64	28.12 ± 0.80	39.96 ± 1.21	26.23 ± 0.84
WC + AC + C	21.02 ± 0.56	47.11 ± 0.24	60.86 ± 0.97	43.00 ± 0.56	16.40 ± 0.32	41.16 ± 0.41	54.47 ± 0.46	37.35 ± 0.18
AA + WC + AC + C	20.19 ± 0.50	43.19 ± 0.29	57.23 ± 0.51	40.20 ± 0.35	15.27 ± 0.04	37.53 ± 0.38	50.91 ± 0.30	34.57 ± 0.23
ASC + WC + AC + C	<u>20.80 ± 0.87</u>	<u>45.29 ± 1.21</u>	<u>58.25 ± 1.34</u>	<u>41.45 ± 1.05</u>	<u>16.11 ± 0.48</u>	<u>38.56 ± 0.67</u>	<u>52.05 ± 0.89</u>	<u>35.57 ± 0.67</u>
ASC + AA + AC + C	14.45 ± 0.42	31.96 ± 0.75	43.73 ± 0.67	30.05 ± 0.31	11.43 ± 0.08	29.47 ± 0.20	40.41 ± 0.19	27.11 ± 0.08
ASC + AA + WC + AC + C	14.64 ± 0.85	31.52 ± 1.18	42.94 ± 1.21	29.70 ± 1.01	11.66 ± 0.52	29.88 ± 0.17	41.51 ± 0.45	27.69 ± 0.32

Table 5.27: Recall@k results for Audio-to-Text and Text-to-Audio retrieval on AudioCaps and ClothoV2 for CED + mGTE + MoL (Self-Att + Mean) under different training-data combinations. Dataset abbreviations are as follows: AudioSetCaps (ASC), Auto-ACD (AA), WavCaps (WC), AudioCaps (AC), and ClothoV2 (C). Values are reported as mean ± standard deviation. Values highlighted in blue indicate the best result for each metric and dataset, while underlined values indicate the second-best result.

Table 5.27 shows that the best combination is strongly dependent on the target evaluation dataset. On AudioCaps, the strongest results were obtained with ASC + WC + AC + C, which reached an audio-to-text R@Avg of 74.06% and a text-to-audio R@Avg of 66.32%. This configuration outperformed AA + WC + AC + C by only 0.14 pp in audio-to-text R@Avg and 0.80 pp in text-to-audio R@Avg, showing that both combinations are highly competitive on AudioCaps. These results suggest that AudioSetCaps, Auto-ACD and WavCaps provide useful supervision for AudioCaps retrieval performance, likely because they increase the diversity and scale of the training audio-caption pairs while remaining compatible with the AudioCaps domain.

Training Data	Classification	
	ESC-50	UrbanSound8K
AA + AC + C	80.40 $\pm$ 2.43	78.05 $\pm$ 4.21
WC + AC + C	91.33 $\pm$ 0.85	79.04 $\pm$ 1.48
AA + WC + AC + C	88.98 $\pm$ 0.72	81.74 $\pm$ 1.54
ASC + WC + AC + C	86.72 $\pm$ 1.22	75.98 $\pm$ 1.81
ASC + AA + AC + C	78.70 $\pm$ 1.53	72.77 $\pm$ 1.30
ASC + AA + WC + AC + C	78.88 $\pm$ 1.89	69.33 $\pm$ 2.30

Table 5.28: *Classification accuracy on ESC-50 and UrbanSound8K for CED + mGTE + MoL (Self-Att + Mean) under different training-data combinations. Dataset abbreviations are as follows: AudioSetCaps (ASC), Auto-ACD (AA), WavCaps (WC), AudioCaps (AC), and ClothoV2 (C). Values are reported as mean $\pm$ standard deviation. Values highlighted in blue indicate the best result for each dataset, while underlined values indicate the second-best result.*

However, the same tendency does not hold for ClothoV2. On ClothoV2, the best results were obtained with WC + AC + C, reaching 43.00% audio-to-text R@Avg and 37.35% text-to-audio R@Avg. Adding Auto-ACD to this combination reduced performance to 40.20% and 34.57%, while adding AudioSetCaps reduced it to 41.45% and 35.57%. This indicates that WavCaps is particularly important for improving generalization on ClothoV2, but that adding additional synthetic or AudioSet-derived data does not necessarily help.

A relevant factor in these results is the origin of the training datasets. AudioCaps is derived from the AudioSet [99] dataset, and both AudioSetCaps and Auto-ACD are derived from AudioSet, although they generate captions through different automatic pipelines [10, 183]. This may help explain why configurations containing AudioSetCaps or Auto-ACD perform competitively on AudioCaps. The training data and the evaluation data share a similar source distribution. However, combining AudioSetCaps and Auto-ACD does not show a complementary effect. Although they provide different captions for AudioSet-related audio, their shared origin may introduce redundancy, while differences in caption-generation pipelines may also introduce heterogeneous annotation styles.

This lack of synergy is most visible when all available datasets are combined. The largest training mixture, ASC + AA + WC + AC + C, performed below the strongest configurations on both AudioCaps and ClothoV2. This suggests that simply increasing the amount of training data is not sufficient. When multiple automatically generated or weakly aligned datasets are combined, the model may receive more training examples, but also more noise, repeated source bias, and inconsistent caption styles. Therefore, the quality and compatibility of the data mixture appear to be more important than its absolute size.

The ClothoV2 results further support this interpretation. ClothoV2 is built from

Freesound audio [45], and WavCaps also includes Freesound among its sources [127]. This source-level overlap may help explain why WC + AC + C was the strongest configuration on ClothoV2. In this case, WavCaps appears to provide useful additional coverage while remaining closer to the ClothoV2 distribution than AudioSet-derived datasets. Therefore, the results indicate that PALMA benefits most from training mixtures that combine scale with domain compatibility, rather than from adding all available datasets indiscriminately.

A similar pattern appears in the classification results shown in Table 5.28. The best ESC-50 accuracy was obtained with WC + AC + C, while the best UrbanSound8K accuracy was obtained with AA + WC + AC + C. This shows that Auto-ACD can still be beneficial when combined with WavCaps, AudioCaps, and ClothoV2, especially for UrbanSound8K. However, when Auto-ACD is combined with AudioSetCaps, the results decrease substantially. Overall, these results suggest that WavCaps is the most consistently useful complementary dataset, while AudioSet-derived automatic datasets must be combined more carefully to avoid redundancy and domain imbalance.

Based on these results, the selected training combination for the following experiments was ASC + WC + AC + C. This choice prioritizes the strongest AudioCaps performance while preserving competitive results on the remaining evaluation settings. AudioSetCaps was preferred over Auto-ACD because it provides a larger caption set, with approximately 1.9M sample pairs compared with 1.5M in Auto-ACD, and because its annotation pipeline is more directly grounded in audio-language modeling rather than in visual information [10, 183]. Visual cues may be useful for generating rich descriptions, but it may also introduce biases toward visually salient scene attributes that are not necessarily present or dominant in audio signals. Therefore, the combination ASC + WC + AC + C was selected for the following experiments.

5.3.7 Batch Size Exploration

The main goal of this phase is to explore the impact of batch size while keeping the training-data composition and model architecture fixed. All experiments use ASC + WC + AC + C as the training mixture and CED + mGTE + MoL with Self-Att+Mean as the base architecture. Since contrastive learning depends strongly on the number of negative examples available in each optimization step, this analysis investigates whether increasing the batch size improves audio-language alignment in the PALMA framework.

Table 5.29 and Figure 5.14 show that increasing the batch size substantially improves AudioCaps retrieval when moving from very small batches to intermediate batch sizes. Audio-to-text R@Avg increased from 70.84% with batch size 24 to 76.18% with batch size 128. Text-to-audio retrieval followed a similar trend, increasing from

Batch size	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
AudioCaps								
24	46.12 ± 1.09	78.22 ± 0.29	88.19 ± 1.00	70.84 ± 0.45	35.17 ± 0.21	70.96 ± 0.20	84.21 ± 0.49	63.45 ± 0.19
48	50.24 ± 1.07	81.02 ± 0.68	90.92 ± 0.32	74.06 ± 0.63	38.48 ± 0.13	73.78 ± 0.38	86.72 ± 0.36	66.32 ± 0.25
64	51.26 ± 2.18	82.64 ± 0.36	91.52 ± 0.45	75.14 ± 0.95	39.11 ± 0.32	74.45 ± 0.13	87.21 ± 0.48	66.92 ± 0.31
96	52.01 ± 1.43	81.74 ± 0.53	91.23 ± 0.45	74.99 ± 0.78	39.07 ± 0.19	75.02 ± 0.32	87.59 ± 0.25	67.23 ± 0.14
128	52.46 ± 0.72	<u>83.57</u> ± 1.67	92.50 ± 0.32	76.18 ± 0.56	40.17 ± 0.30	75.06 ± 0.29	87.30 ± 0.48	67.51 ± 0.14
192	<u>52.04</u> ± 1.16	82.94 ± 1.11	91.68 ± 0.98	75.55 ± 0.49	<u>40.04</u> ± 0.52	75.56 ± 0.60	<u>87.76</u> ± 0.22	67.78 ± 0.44
256	51.89 ± 0.96	84.14 ± 0.51	<u>91.86</u> ± 0.26	<u>75.97</u> ± 0.25	40.01 ± 0.30	<u>75.40</u> ± 0.31	87.77 ± 0.19	<u>67.73</u> ± 0.10
384	50.55 ± 1.30	82.38 ± 0.97	91.60 ± 0.52	74.84 ± 0.85	39.64 ± 0.60	74.93 ± 0.75	87.00 ± 0.59	67.19 ± 0.58
512	50.92 ± 0.23	81.77 ± 0.70	90.70 ± 0.80	74.47 ± 0.47	39.61 ± 0.37	74.73 ± 0.21	86.37 ± 0.14	66.90 ± 0.17
ClothoV2								
24	18.50 ± 0.91	41.40 ± 0.89	53.91 ± 0.80	37.94 ± 0.62	14.87 ± 0.17	35.86 ± 0.45	49.33 ± 0.72	33.36 ± 0.36
48	20.80 ± 0.87	45.29 ± 1.21	58.25 ± 1.34	41.45 ± 1.05	16.11 ± 0.48	38.56 ± 0.67	52.05 ± 0.89	35.57 ± 0.67
64	21.34 ± 1.41	44.98 ± 1.22	59.71 ± 0.33	42.01 ± 0.80	16.45 ± 0.57	39.53 ± 0.26	52.93 ± 0.47	36.30 ± 0.27
96	21.82 ± 0.42	45.45 ± 0.69	59.65 ± 0.40	42.31 ± 0.32	16.73 ± 0.36	40.10 ± 0.27	53.23 ± 0.04	36.69 ± 0.13
128	<u>22.07</u> ± 0.40	<u>46.15</u> ± 1.62	<u>60.03</u> ± 1.39	<u>42.75</u> ± 1.05	16.80 ± 0.27	40.00 ± 0.58	53.55 ± 0.36	36.79 ± 0.37
192	<u>22.07</u> ± 0.40	<u>46.15</u> ± 1.62	<u>60.03</u> ± 1.39	<u>42.75</u> ± 1.05	16.80 ± 0.27	40.00 ± 0.58	53.55 ± 0.36	36.79 ± 0.37
256	21.92 ± 1.54	45.90 ± 1.30	<u>60.03</u> ± 1.12	42.62 ± 1.10	16.39 ± 0.35	40.42 ± 0.56	53.92 ± 0.66	<u>36.91</u> ± 0.46
384	21.85 ± 0.67	45.45 ± 0.75	59.55 ± 1.23	42.29 ± 0.85	16.81 ± 0.18	40.33 ± 0.57	<u>53.77</u> ± 0.52	36.97 ± 0.30
512	22.17 ± 0.38	46.51 ± 0.67	60.41 ± 0.80	43.03 ± 0.24	16.92 ± 0.27	<u>40.39</u> ± 0.38	53.17 ± 0.21	36.83 ± 0.18

Table 5.29: Recall@k results for Audio-to-Text and Text-to-Audio retrieval on AudioCaps and ClothoV2 across different batch sizes. All configurations use CED+mGTE+MoL with Self-Att+Mean trained on AudioSetCaps, WavCaps, AudioCaps, and ClothoV2. Values are reported as mean ± standard deviation. Values highlighted in blue indicate the best result for each metric and dataset, while underlined values indicate the second-best result.

Batch size	ESC-50	UrbanSound8K
24	83.90 ± 0.97	77.40 ± 1.02
48	86.72 ± 1.22	75.98 ± 1.81
64	88.90 ± 1.05	79.17 ± 1.17
96	90.08 ± 0.77	77.05 ± 4.02
128	91.42 ± 0.52	77.27 ± 1.66
192	90.00 ± 0.95	76.33 ± 1.77
256	<u>90.22</u> ± 0.38	<u>79.48</u> ± 1.16
384	89.62 ± 1.48	79.21 ± 2.28
512	90.13 ± 0.45	79.60 ± 0.53

Table 5.30: Zero-shot classification accuracy on ESC-50 and UrbanSound8K across different batch sizes. All configurations use CED+mGTE+MoL with Self-Att+Mean trained on AudioSetCaps, WavCaps, AudioCaps, and ClothoV2. Values are reported as mean ± standard deviation. Values highlighted in blue indicate the best result for each dataset, while underlined values indicate the second-best result.

63.45% to 67.51% in the same interval. This confirms the importance of using sufficiently large batches for contrastive audio-language learning, since larger batches provide more

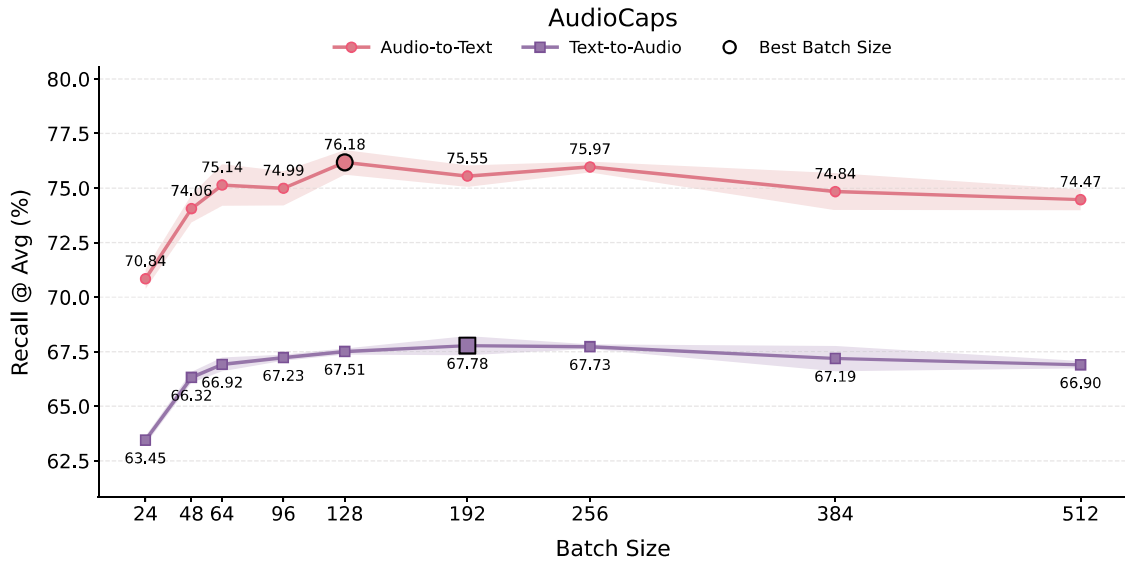


Figure 5.14: *Effect of batch size on retrieval performance on AudioCaps. The plot shows Recall@Avg for Audio-to-Text and Text-to-Audio retrieval across different batch sizes. The shaded regions indicate the standard deviation across runs.*

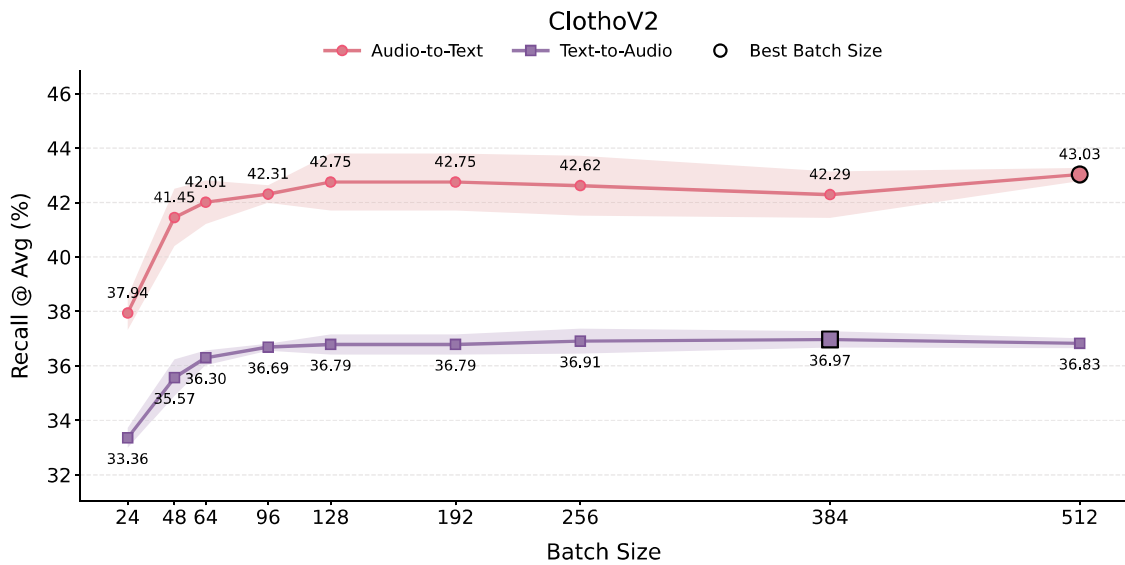


Figure 5.15: *Effect of batch size on retrieval performance on ClothoV2. The plot shows Recall@Avg for Audio-to-Text and Text-to-Audio retrieval across different batch sizes. The shaded regions indicate the standard deviation across runs.*

negative examples and a stronger alignment signal during training.

The AudioCaps curve also shows that the main gains occur before batch size 128. After this point, performance no longer improves consistently. The best audio-to-text R@Avg was obtained with batch size 128, while the best text-to-audio R@Avg was

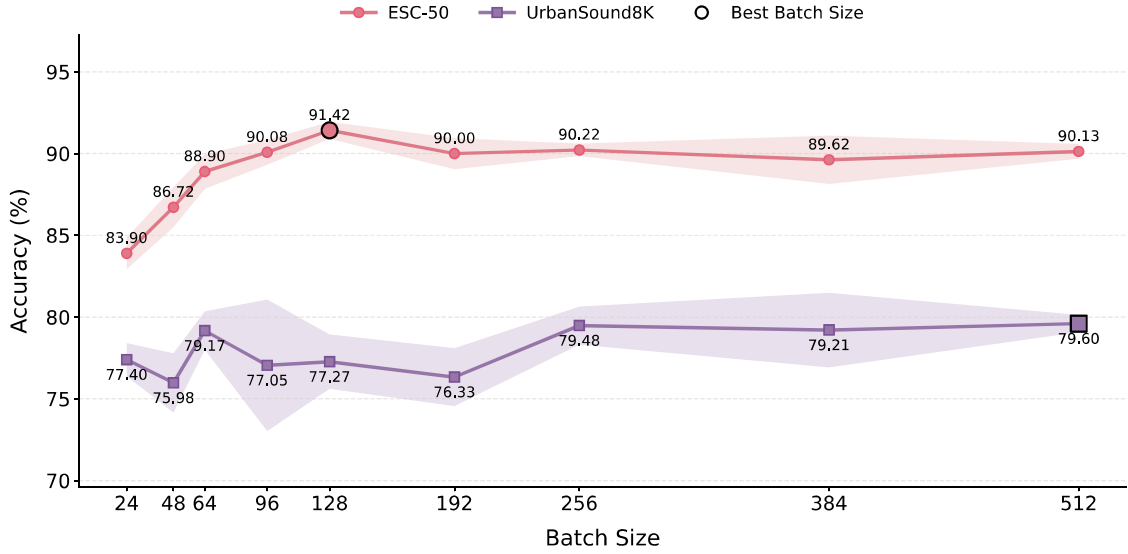


Figure 5.16: Effect of batch size on zero-shot classification performance. The plot shows classification accuracy on ESC-50 and UrbanSound8K across different batch sizes. The shaded regions indicate the standard deviation across runs.

obtained with batch size 192. Larger batch sizes, such as 384 and 512, slightly reduced Recall@Avg in both retrieval directions. This suggests that larger batches are useful up to a certain point, but overly large batches do not necessarily improve optimization.

A similar trend is observed on ClothoV2, as shown in Figure 5.15. Audio-to-text R@Avg increased from 37.94% with batch size 24 to 42.75% with batch size 128, while text-to-audio R@Avg increased from 33.36% to 36.79%. In contrast to AudioCaps, ClothoV2 continued to show small improvements at larger batch sizes. The best audio-to-text R@Avg was obtained with batch size 512, and the best text-to-audio R@Avg was obtained with batch size 384. However, the gains after batch size 128 are small, indicating that ClothoV2 also reaches a plateau once the batch is sufficiently large.

Table 5.30 and Figure 5.16 show that classification performance follows a related but not identical pattern. ESC-50 accuracy improved markedly from batch size 24 to 128, reaching a best of 91.42% at 128. UrbanSound8K was more variable, with the best accuracy achieved at a batch size of 512. Nevertheless, batch sizes of 64, 256, 384, and 512 yielded relatively close UrbanSound8K results, whereas very small batches were less competitive. This indicates that the optimal batch size can vary with the downstream task, but very small batches are consistently suboptimal.

Overall, this phase shows that batch size is an important factor for PALMA, especially in the contrastive retrieval setting. Very small batches provide weaker alignment, while intermediate and large batches substantially improve both retrieval and classification. However, the gains tend to saturate after batch size 128, and larger batches do not

consistently improve all metrics. Therefore, batch size 128 was selected as the most balanced configuration: it achieves the best AudioCaps audio-to-text retrieval, the best ESC-50 classification accuracy, competitive ClothoV2 performance, and avoids the additional computational cost of larger batches.

5.3.8 Multilingual Supervision

Finally, this phase investigates how much PALMA benefits from multilingual textual supervision. Previously, PALMA was only trained using English captions. Since the text encoder is multilingual, an important question is whether the model can retain or recover multilingual capabilities after explicit audio-language alignment. To analyze this, we evaluate the model using English captions and machine-translated captions in eleven additional languages: Portuguese, Spanish, French, Russian, Arabic, Hindi, German, Chinese, Swahili, Japanese, and Turkish. The complete per-language retrieval and classification results are reported in [Appendix E](#).

Multilingual Training Strategies

We compare three types of multilingual supervision. The first is the English-only configuration, trained only with English captions for 10 epochs. This setting is used to test whether multilingual capabilities remain available after tuning the model exclusively on English audio-text pairs, and to have a strong English baseline. The second strategy is Random Sampling, in which the caption language is randomly selected from the twelve available languages during training. This exposes the model to all languages throughout training while keeping one positive text per audio at each step. The third strategy is LE-Mixture, inspired by the language-enhancement mixture strategy used in ML-CLAP [207] and GLAP [40]. In this setting, translated captions are mixed with the original English captions at different ratios, allowing us to analyze how the amount of multilingual supervision affects retrieval and classification performance.

Effect on Multilingual Retrieval

Table 5.31 summarizes the retrieval results under the different multilingual training strategies. The English-only configuration performs competitively when evaluated in English, reaching 73.84% and 65.99% R@Avg on AudioCaps for audio-to-text and text-to-audio retrieval, respectively. However, its performance collapses under multilingual evaluation, reaching only 3.87% and 2.56% on AudioCaps ML Avg. and 2.27% and 1.51% on ClothoV2 ML Avg, indicating that training only on English captions severely weakens the multilingual alignment of the shared audio-text latent space, even though the text encoder itself is originally multilingual.

Model	AudioCaps				ClothoV2			
	EN		ML Avg.		EN		ML Avg.	
	A2T	T2A	A2T	T2A	A2T	T2A	A2T	T2A
English-only	73.84 ± 1.03	65.99 ± 0.44	3.87 ± 0.23	2.56 ± 0.17	40.33 ± 0.60	33.53 ± 0.19	2.27 ± 0.09	1.51 ± 0.08
Random Sampling	76.89 ± 0.43	67.36 ± 0.45	74.92 ± 0.12	64.95 ± 0.26	40.76 ± 0.92	36.08 ± 0.17	38.98 ± 0.45	34.79 ± 0.14
LE-Mixture 10%	75.31 ± 0.34	66.35 ± 0.37	69.13 ± 0.35	58.28 ± 0.25	38.13 ± 0.40	33.41 ± 0.50	31.93 ± 0.44	27.27 ± 0.11
LE-Mixture 20%	76.02 ± 1.12	66.94 ± 0.37	71.98 ± 0.29	61.34 ± 0.56	38.63 ± 0.66	33.64 ± 0.45	34.75 ± 0.07	29.92 ± 0.25
LE-Mixture 30%	75.92 ± 0.67	66.97 ± 0.39	73.21 ± 0.23	62.96 ± 0.10	39.54 ± 0.67	34.20 ± 0.34	36.41 ± 0.59	31.60 ± 0.18
LE-Mixture 40%	76.77 ± 0.78	66.64 ± 0.69	73.89 ± 0.33	63.31 ± 0.22	39.25 ± 0.87	34.61 ± 0.03	36.74 ± 0.43	32.33 ± 0.12
LE-Mixture 50%	75.96 ± 0.38	66.75 ± 0.38	73.54 ± 0.62	63.36 ± 0.19	40.21 ± 0.52	<u>35.25</u> ± 0.37	37.24 ± 0.41	32.60 ± 0.17
LE-Mixture 100%	77.27 ± 0.88	67.44 ± 0.08	74.61 ± 0.39	64.37 ± 0.05	40.14 ± 0.24	35.17 ± 0.22	<u>37.58</u> ± 0.23	<u>33.10</u> ± 0.29

Table 5.31: Retrieval performance under different multilingual training strategies. EN denotes evaluation on English captions, while ML (Multilingual) Avg. denotes the average across the non-English evaluation languages. A2T and T2A correspond to Audio-to-Text and Text-to-Audio retrieval, respectively. All values are reported using Recall@Avg as mean ± standard deviation across three runs. The best result for each metric is highlighted in blue and shown in bold, while the second-best result is underlined.

In contrast, multilingual supervision substantially improves cross-lingual retrieval. Random Sampling achieves the best multilingual retrieval performance across all ML Avg. metrics, reaching 74.92% and 64.95% on AudioCaps, and 38.98% and 34.79% on ClothoV2. These results show that exposing the model to different languages during training is essential for maintaining multilingual retrieval capability. Importantly, Random Sampling also obtains the best English performance on ClothoV2 and remains highly competitive on AudioCaps, suggesting that multilingual training does not harm English retrieval when the language exposure is balanced.

The LE-Mixture configurations also improve multilingual performance compared with the English-only model, although their behavior depends on the mixture ratio. On AudioCaps, increasing the LE-Mixture ratio generally improves multilingual retrieval, with LE-Mixture 100% obtaining the second-best ML Avg. results in both retrieval directions. This suggests that increasing the proportion of translated captions can strengthen the alignment between non-English text and audio representations. This behavior differs from ML-CLAP [207], where LE-Mixture was originally proposed and where smaller mixture ratios, between 10% and 30%, were reported as optimal. In that work, the authors argue that larger ratios may make contrastive learning more difficult because the same audio sample is associated with multiple multilingual captions. In PALMA, however, the gains from increasing the LE-Mixture ratio are more noticeable, with the best LE-Mixture results obtained at 100%. In this configuration, the amount of training text is effectively doubled, with one half corresponding to the original English captions and the other half corresponding to translated captions. One possible explanation for this difference is translation quality. ML-CLAP uses SONAR [47] to generate multilingual captions, whereas

our experiments use Google Translate. Therefore, even though the model observes multiple textual views of the same audio, higher-quality translations may provide semantically consistent multilingual supervision and mitigate the negative effect of caption duplication.

Nevertheless, LE-Mixture does not surpass Random Sampling in multilingual evaluation. This is a relevant result because previous contrastive ALMs, ML-CLAP [207] and GLAP [40], that explore multilingual supervision, rely on LE-Mixture-style training to incorporate multilingual data. In contrast, PALMA shows that a simpler strategy, in which one caption is randomly sampled from the twelve available languages for each audio sample, is sufficient to match or surpass the performance of the more complex LE-Mixture strategy. A possible explanation is that LE-Mixture increases the number of textual views associated with the same audio, which can make the contrastive objective more ambiguous. Random Sampling, on the other hand, preserves a simpler one-audio-and-one-text structure at each optimization step while still exposing the model to all languages throughout training. This may provide a more stable multilingual alignment signal, which helps explain why Random Sampling achieves stronger overall multilingual performance.

Figure 5.17 provides a view of the AudioCaps retrieval results with emphasis on the evaluated languages. The plot shows that Random Sampling is consistently strong across languages and retrieval directions, whereas lower LE-Mixture ratios lead to larger performance drops in some languages. This behavior is especially evident in the Text-to-Audio direction, where LE-Mixture 10% exhibits clearer degradation across several languages. As the LE-Mixture ratio increases, the curves become more competitive, but Random Sampling remains the most stable strategy overall.

Another relevant pattern in the multilingual supervision experiments is the improvement observed for English itself. All multilingual configurations outperform the English-only baseline in the English evaluation setting, indicating that multilingual training not only improves cross-lingual transfer but also benefits performance in the datasets’ original languages. This suggests that exposing the model to semantically equivalent captions in multiple languages may provide an additional regularization effect, encouraging the model to learn more language-agnostic audio-text representations rather than overfitting to English-specific lexical patterns, a pattern also observed in previous works [40, 207].

Effect on Zero-Shot Classification

A similar pattern is observed in zero-shot classification, as shown in Table 5.32. The English-only configuration achieves reasonable English performance, but its multilingual average is very low, especially on ESC-50. Random Sampling achieves the best results across all classification metrics, reaching 90.12% on ESC-50 and 74.92% on Ur-

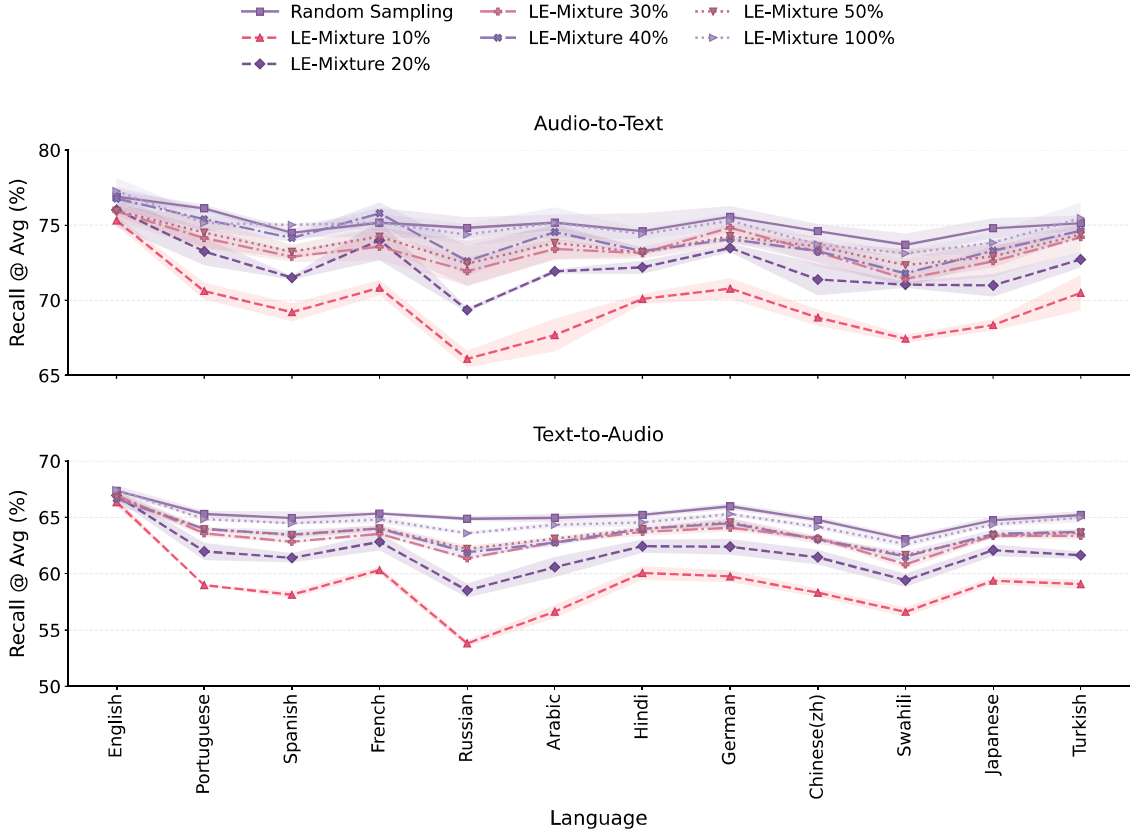


Figure 5.17: *AudioCaps multilingual retrieval performance under different multilingual training strategies. The top plot shows Audio-to-Text retrieval and the bottom plot shows Text-to-Audio retrieval, both using Recall@Avg across the twelve evaluation languages. Shaded regions indicate the standard deviation across runs.*

banSound8K in English, and 86.37% and 73.70% in ML Avg., respectively. This indicates that multilingual supervision not only improves retrieval but also transfers to prompt-based zero-shot classification. Among the LE-Mixture configurations, higher ratios generally improve the average multilingual classification, with LE-Mixture 50% achieving the second-best ML Avg. results on both ESC-50 and UrbanSound8K. However, these configurations remain below Random Sampling.

Comparison with Multilingual Baselines

Tables 5.33 and 5.34 compare the strongest PALMA multilingual configurations with ML-CLAP [207] and GLAP [40]. In this comparison, PALMA is represented by Random Sampling, the strongest multilingual configuration overall, and by LE-Mixture 100%, which achieved the strongest English retrieval results among the LE-Mixture variants. The goal of this analysis is not only to compare aggregate performance but also

Model	ESC-50		UrbanSound8K	
	EN	ML Avg.	EN	ML Avg.
English-only	83.50 $\pm$ 2.85	6.58 $\pm$ 0.76	71.44 $\pm$ 5.10	16.27 $\pm$ 1.11
Random Sampling	90.12 $\pm$ 1.35	86.37 $\pm$ 0.60	74.92 $\pm$ 4.60	73.70 $\pm$ 3.15
LE-Mixture 10%	81.43 $\pm$ 2.39	68.92 $\pm$ 1.19	73.90 $\pm$ 4.09	66.49 $\pm$ 2.52
LE-Mixture 20%	<u>86.53 $\pm$ 2.39</u>	77.20 $\pm$ 1.00	73.29 $\pm$ 1.90	70.21 $\pm$ 0.12
LE-Mixture 30%	86.27 $\pm$ 3.34	80.99 $\pm$ 1.09	<u>74.12 $\pm$ 2.36</u>	71.19 $\pm$ 3.75
LE-Mixture 40%	84.57 $\pm$ 1.83	81.26 $\pm$ 0.99	72.30 $\pm$ 2.22	69.16 $\pm$ 1.47
LE-Mixture 50%	84.05 $\pm$ 1.01	<u>82.29 $\pm$ 0.55</u>	70.69 $\pm$ 1.53	<u>71.60 $\pm$ 0.70</u>
LE-Mixture 100%	82.53 $\pm$ 2.36	80.90 $\pm$ 0.64	72.68 $\pm$ 5.53	70.04 $\pm$ 0.85

Table 5.32: Zero-shot classification performance under different multilingual training strategies. EN denotes evaluation using English class prompts, while ML (Multilingual) Avg. denotes the average across the non-English evaluation languages. All values are reported as mean $\pm$ standard deviation across three runs. The best result for each metric is highlighted in blue and shown in bold, while the second-best result is underlined.

to assess whether PALMA strikes a competitive balance among English performance, multilingual robustness, retrieval, and zero-shot classification.

Model	AudioCaps				ClothoV2			
	EN		ML Avg.		EN		ML Avg.	
	A2T	T2A	A2T	T2A	A2T	T2A	A2T	T2A
PALMA Random Sampling	76.89 $\pm$ 0.43	67.36 $\pm$ 0.45	74.92 $\pm$ 0.12	64.95 $\pm$ 0.26	40.76 $\pm$ 0.92	36.08 $\pm$ 0.17	38.98 $\pm$ 0.45	<u>34.79 $\pm$ 0.14</u>
PALMA LE-Mixture 100%	77.27 $\pm$ 0.88	<u>67.44 $\pm$ 0.08</u>	74.61 $\pm$ 0.39	64.37 $\pm$ 0.05	40.14 $\pm$ 0.24	35.17 $\pm$ 0.22	37.58 $\pm$ 0.23	33.10 $\pm$ 0.29
ML-CLAP	<u>76.98</u>	67.90	67.35 $\pm$ 3.71	56.81 $\pm$ 4.18	50.49	46.51	45.07 $\pm$ 1.40	39.59 $\pm$ 1.58
GLAP	71.24	54.93	54.82 $\pm$ 7.79	40.19 $\pm$ 8.47	27.88	26.04	21.36 $\pm$ 4.31	18.23 $\pm$ 5.81

Table 5.33: Retrieval performance comparison between PALMA multilingual training strategies and multilingual audio-language baselines. EN denotes evaluation on English captions, while ML Avg. denotes the average across non-English evaluation languages. A2T and T2A correspond to Audio-to-Text and Text-to-Audio retrieval, respectively. All values are reported using Recall@Avg. PALMA values are reported as mean $\pm$ standard deviation across three runs. ML-CLAP and GLAP contain one run per language; therefore, their EN values are reported without standard deviation, while ML Avg. values are reported as mean $\pm$ standard deviation across non-English languages. The best result for each metric is highlighted in blue and shown in bold, while the second-best result is underlined.

For retrieval, the results show a clear dataset-dependent behavior. On AudioCaps, PALMA is the strongest model under multilingual evaluation. PALMA Random Sampling obtains the best ML Avg. performance in both retrieval directions, reaching 74.92% for Audio-to-Text and 64.95% for Text-to-Audio. These results are substantially higher than ML-CLAP, which reaches 67.35% and 56.81%, and GLAP, which reaches 54.82% and

Model	ESC-50		UrbanSound8K	
	EN	ML Avg.	EN	ML Avg.
PALMA Random Sampling	90.12 $\pm$ 1.35	86.37 $\pm$ 0.60	74.92 $\pm$ 4.60	73.70 $\pm$ 3.15
PALMA LE-Mixture 100%	82.53 $\pm$ 2.36	80.90 $\pm$ 0.64	72.68 $\pm$ 5.53	70.04 $\pm$ 0.85
ML-CLAP	95.70 $\pm$ 0.66	85.17 $\pm$ 5.79	83.84 $\pm$ 3.80	77.24 $\pm$ 5.45
GLAP	81.10 $\pm$ 1.91	60.97 $\pm$ 13.68	51.40 $\pm$ 1.39	56.22 $\pm$ 9.60

Table 5.34: *Zero-shot classification comparison between PALMA multilingual training strategies and multilingual audio-language baselines. EN denotes evaluation using English class prompts, while ML Avg. denotes the average across non-English evaluation languages. PALMA values are reported as mean $\pm$ standard deviation across three runs. ML-CLAP and GLAP values are reported as mean $\pm$ standard deviation across folds for EN, while their ML Avg. values are computed across non-English language averages. The best result for each metric is highlighted in **blue** and shown in **bold**, while the second-best result is underlined.*

40.19%. This indicates that the multilingual supervision strategy used in PALMA is particularly effective for preserving cross-lingual retrieval performance on AudioCaps. In English, PALMA LE-Mixture 100% obtains the best Audio-to-Text result, while ML-CLAP obtains the best Text-to-Audio result. Therefore, on AudioCaps, PALMA is especially competitive, achieving the strongest multilingual average while remaining close to, or better than, the baselines.

The behavior is different on ClothoV2. ML-CLAP obtains the best results in all English and multilingual retrieval metrics, reaching 50.49% and 46.51% in English, and 45.07% and 39.59% in ML Avg. for Audio-to-Text and Text-to-Audio, respectively. PALMA Random Sampling is the second-best model across all ClothoV2 retrieval metrics, with 40.76% and 36.08% in English, and 38.98% and 34.79% in ML Avg. This suggests that ML-CLAP transfers particularly well to ClothoV2, while PALMA remains competitive but does not close the gap on this dataset. One possible explanation is that ML-CLAP is explicitly designed for multilingual audio-text retrieval and may benefit from a training configuration that is especially well-matched to the AudioCaps–Clotho evaluation setting. In contrast, GLAP performs substantially below the other models on both retrieval datasets, especially under multilingual evaluation, suggesting weaker cross-lingual audio-text alignment in this benchmark setting.

The complete per-language retrieval Tables E.2, 5.36 and 5.37 provide additional insight into these aggregate results. For PALMA Random Sampling, the AudioCaps performance remains relatively stable across languages, with only a small gap between English and most non-English languages. This indicates that Random Sampling not only improves the average result but also produces a consistently multilingual retrieval space. ML-CLAP also shows strong multilingual retrieval for the languages available in its

evaluation set, especially on ClothoV2. GLAP, however, shows larger drops across several languages. These complete results suggest that PALMA Random Sampling is particularly effective at reducing the gap between English and non-English retrieval on AudioCaps.

Random Sampling								
Language	AudioCaps							
	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	53.81 $\pm$ 0.81	83.62 $\pm$ 0.51	93.25 $\pm$ 1.08	76.89 $\pm$ 0.43	39.59 $\pm$ 0.75	75.33 $\pm$ 0.80	87.16 $\pm$ 0.24	67.36 $\pm$ 0.45
Portuguese	<u>53.39</u> $\pm$ 0.06	<u>83.24</u> $\pm$ 0.70	91.72 $\pm$ 0.06	<u>76.12</u> $\pm$ 0.23	37.80 $\pm$ 0.40	73.01 $\pm$ 0.26	85.07 $\pm$ 0.21	65.29 $\pm$ 0.29
Spanish	50.62 $\pm$ 1.19	81.40 $\pm$ 0.66	91.45 $\pm$ 0.60	74.49 $\pm$ 0.28	37.32 $\pm$ 1.08	72.55 $\pm$ 0.58	84.97 $\pm$ 0.29	64.94 $\pm$ 0.58
French	52.38 $\pm$ 0.28	82.04 $\pm$ 1.24	91.07 $\pm$ 1.62	75.16 $\pm$ 1.00	37.66 $\pm$ 0.50	72.93 $\pm$ 0.26	85.44 $\pm$ 0.27	65.34 $\pm$ 0.14
Russian	50.84 $\pm$ 0.88	81.66 $\pm$ 1.47	91.94 $\pm$ 0.40	74.82 $\pm$ 0.71	37.57 $\pm$ 0.20	72.30 $\pm$ 0.36	84.75 $\pm$ 0.55	64.87 $\pm$ 0.21
Arabic	51.52 $\pm$ 0.70	81.92 $\pm$ 0.88	92.05 $\pm$ 0.28	75.17 $\pm$ 0.55	37.23 $\pm$ 0.63	72.68 $\pm$ 0.41	84.94 $\pm$ 0.25	64.95 $\pm$ 0.39
Hindi	51.78 $\pm$ 1.77	81.07 $\pm$ 1.50	90.93 $\pm$ 0.94	74.59 $\pm$ 1.20	37.35 $\pm$ 0.52	72.82 $\pm$ 0.27	85.50 $\pm$ 0.30	65.22 $\pm$ 0.23
German	52.16 $\pm$ 0.17	82.30 $\pm$ 1.01	<u>92.20</u> $\pm$ 1.05	75.55 $\pm$ 0.71	<u>38.20</u> $\pm$ 0.70	<u>73.51</u> $\pm$ 0.43	<u>86.25</u> $\pm$ 0.17	<u>65.98</u> $\pm$ 0.29
Chinese(zh)	50.88 $\pm$ 1.24	81.48 $\pm$ 0.45	91.41 $\pm$ 0.26	74.59 $\pm$ 0.49	36.84 $\pm$ 0.35	72.21 $\pm$ 0.07	85.28 $\pm$ 0.43	64.78 $\pm$ 0.28
Swahili	49.12 $\pm$ 1.54	80.61 $\pm$ 1.80	91.34 $\pm$ 0.49	73.69 $\pm$ 0.75	35.69 $\pm$ 0.29	70.32 $\pm$ 0.32	83.19 $\pm$ 0.19	63.06 $\pm$ 0.25
Japanese	51.26 $\pm$ 0.28	81.96 $\pm$ 1.04	91.15 $\pm$ 0.72	74.79 $\pm$ 0.67	36.81 $\pm$ 0.77	72.05 $\pm$ 0.19	85.36 $\pm$ 0.49	64.74 $\pm$ 0.34
Turkish	51.74 $\pm$ 1.14	82.04 $\pm$ 0.13	91.60 $\pm$ 0.69	75.13 $\pm$ 0.53	37.39 $\pm$ 0.52	72.42 $\pm$ 0.28	85.83 $\pm$ 0.37	65.21 $\pm$ 0.35

ClothoV2								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	21.05 $\pm$ 1.23	43.67 $\pm$ 0.48	57.57 $\pm$ 1.07	40.76 $\pm$ 0.92	16.31 $\pm$ 0.20	39.39 $\pm$ 0.35	52.56 $\pm$ 0.25	36.08 $\pm$ 0.17
Portuguese	<u>20.38</u> $\pm$ 0.72	<u>43.64</u> $\pm$ 1.34	<u>56.46</u> $\pm$ 0.42	<u>40.16</u> $\pm$ 0.77	15.83 $\pm$ 0.28	38.98 $\pm$ 0.44	<u>52.15</u> $\pm$ 0.44	35.65 $\pm$ 0.26
Spanish	20.29 $\pm$ 0.82	42.87 $\pm$ 0.78	55.95 $\pm$ 0.38	39.70 $\pm$ 0.40	15.87 $\pm$ 0.45	<u>39.02</u> $\pm$ 0.18	52.10 $\pm$ 0.80	<u>35.67</u> $\pm$ 0.29
French	19.07 $\pm$ 0.72	41.98 $\pm$ 0.20	55.50 $\pm$ 1.11	38.85 $\pm$ 0.45	15.33 $\pm$ 0.57	38.02 $\pm$ 0.47	51.53 $\pm$ 0.06	34.96 $\pm$ 0.29
Russian	19.46 $\pm$ 1.25	41.57 $\pm$ 0.87	55.09 $\pm$ 1.21	38.71 $\pm$ 1.10	15.03 $\pm$ 0.56	37.69 $\pm$ 0.35	50.65 $\pm$ 0.16	34.46 $\pm$ 0.20
Arabic	19.17 $\pm$ 0.80	41.63 $\pm$ 1.08	55.22 $\pm$ 0.77	38.67 $\pm$ 0.80	14.81 $\pm$ 0.30	37.10 $\pm$ 0.47	50.42 $\pm$ 0.17	34.11 $\pm$ 0.12
Hindi	19.97 $\pm$ 0.69	42.71 $\pm$ 1.55	55.66 $\pm$ 1.14	39.45 $\pm$ 1.04	<u>15.97</u> $\pm$ 0.33	38.15 $\pm$ 0.76	51.13 $\pm$ 0.44	35.09 $\pm$ 0.46
German	19.20 $\pm$ 0.40	42.55 $\pm$ 1.45	55.85 $\pm$ 1.36	39.20 $\pm$ 1.03	15.96 $\pm$ 0.38	38.76 $\pm$ 0.44	51.62 $\pm$ 0.52	35.45 $\pm$ 0.20
Chinese(zh)	19.30 $\pm$ 1.30	41.88 $\pm$ 0.81	54.26 $\pm$ 1.63	38.48 $\pm$ 0.94	15.13 $\pm$ 0.42	38.06 $\pm$ 0.18	51.18 $\pm$ 0.22	34.79 $\pm$ 0.06
Swahili	18.41 $\pm$ 1.38	41.69 $\pm$ 0.45	54.45 $\pm$ 1.41	38.18 $\pm$ 1.02	14.17 $\pm$ 0.41	36.38 $\pm$ 0.21	48.92 $\pm$ 0.07	33.16 $\pm$ 0.19
Japanese	19.11 $\pm$ 0.14	42.17 $\pm$ 0.82	54.80 $\pm$ 0.58	38.69 $\pm$ 0.33	15.19 $\pm$ 0.18	38.48 $\pm$ 0.52	51.55 $\pm$ 0.29	35.07 $\pm$ 0.22
Turkish	19.55 $\pm$ 1.15	41.79 $\pm$ 0.48	54.71 $\pm$ 1.36	38.68 $\pm$ 0.32	15.25 $\pm$ 0.18	37.47 $\pm$ 0.26	50.01 $\pm$ 0.37	34.24 $\pm$ 0.18

Table 5.35: *Language-wise retrieval performance for the Random Sampling strategy. AudioCaps and ClothoV2 are shown in separate blocks. A2T and T2A denote Audio-to-Text and Text-to-Audio retrieval, respectively. Values are reported as Recall@k mean $\pm$ standard deviation across three runs. The best value for each metric within each dataset block is shown in bold, while the second-best value is underlined.*

For zero-shot classification, represented in Table 5.34 and in more details in Table 5.38, the comparison shows a different trade-off. ML-CLAP obtains the best English performance on both ESC-50 and UrbanSound8K, reaching 95.70% and 83.84%, respectively. It also obtains the best UrbanSound8K ML Avg. result, with 77.24%. PALMA Random Sampling, however, obtains the best ESC-50 ML Avg. result, reaching 86.37%, slightly above ML-CLAP’s 85.17%. This result also shows that PALMA’s multilingual supervision transfers not only to retrieval, but also to prompt-based classification. Although ML-CLAP remains stronger in English classification and on UrbanSound8K, PALMA Random Sampling provides the strongest multilingual ESC-50 result and remains the second-best model on the remaining classification metrics.

		Audio-to-Text				Text-to-Audio			
		AudioCaps							
Language		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
ML-CLAP	English	56.13	83.13	91.68	76.98	40.70	75.77	87.22	67.90
	Spanish	46.68	76.04	86.84	69.85	<u>34.47</u>	<u>67.31</u>	<u>81.03</u>	<u>60.94</u>
	French	<u>47.81</u>	75.48	86.16	69.82	33.68	65.74	78.67	59.36
	German	47.13	<u>77.17</u>	<u>87.06</u>	<u>70.45</u>	32.64	65.71	78.58	58.98
	Chinese(zh)	39.82	68.73	80.09	62.88	26.82	56.90	70.10	51.27
	Japanese	39.93	69.07	82.23	63.74	28.48	59.26	72.82	53.52
		ClothoV2							
Language		R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
ML-CLAP	English	29.38	54.55	67.56	50.49	23.67	51.10	64.75	46.51
	Spanish	23.35	<u>50.72</u>	<u>65.07</u>	<u>46.38</u>	<u>19.83</u>	45.19	<u>58.93</u>	<u>41.31</u>
	French	<u>25.07</u>	<u>50.72</u>	63.16	46.32	19.29	<u>45.22</u>	58.24	40.92
	German	24.02	48.13	62.87	45.01	18.24	43.75	56.34	39.44
	Chinese(zh)	22.20	46.12	60.57	42.97	16.78	41.19	54.28	37.42
	Japanese	24.11	48.71	61.15	44.66	17.97	42.79	55.77	38.85

Table 5.36: Complete ML-CLAP retrieval results across evaluation languages on AudioCaps and ClothoV2. A2T and T2A correspond to Audio-to-Text and Text-to-Audio retrieval, respectively. **Bold** values indicate the best language for each metric and dataset, while underlined values indicate the second-best language.

GLAP shows lower classification performance than PALMA and ML-CLAP. Its English ESC-50 result is 81.10%, and its ML Avg. drops to 60.97%, with a high standard deviation across languages. On UrbanSound8K, GLAP obtains 51.40% in English and 56.22% in ML Avg. The complete classification table shows that GLAP is highly variable across languages, with strong results for some language-dataset combinations but much weaker results for others. This suggests that, although GLAP has multilingual capabilities, its multilingual transfer is less stable in the evaluated zero-shot classification setting.

English Performance Gains from Multilingual Supervision

Although the main goal of this phase is to improve multilingual audio-language alignment, an important secondary effect is observed in the English evaluation setting. As shown in Tables 5.39 and 5.40, the multilingual supervision, like the Random Sampling configuration consistently improves over the PALMA English-only baseline across retrieval and zero-shot classification.

For retrieval, PALMA Random ML improves over PALMA English-only on both AudioCaps and ClothoV2. On AudioCaps, the Audio-to-Text R@Avg increases from 73.84% to 76.89%, while Text-to-Audio R@Avg increases from 65.99% to 67.36%. On ClothoV2, the gains are smaller in Audio-to-Text retrieval, increasing from 40.33%

		Audio-to-Text				Text-to-Audio			
		AudioCaps							
	Language	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
GLAP	English	46.34	78.85	88.53	71.24	26.86	61.12	76.81	54.93
	Portuguese	36.00	70.19	81.89	62.69	19.96	48.93	63.24	44.04
	Spanish	35.77	65.92	78.97	60.22	<u>23.24</u>	<u>54.60</u>	<u>68.98</u>	<u>48.94</u>
	French	<u>36.56</u>	<u>70.53</u>	<u>84.25</u>	<u>63.78</u>	22.81	53.72	68.68	48.41
	Russian	20.92	42.74	54.33	39.33	8.21	21.93	30.73	20.29
	Arabic	31.95	61.75	74.35	56.02	17.91	44.39	58.83	40.37
	Hindi	30.48	57.37	71.77	53.21	18.63	46.16	61.60	42.13
	German	35.88	67.83	80.76	61.49	22.52	52.76	68.91	48.06
	Chinese(zh)	29.58	61.19	74.92	55.23	16.15	42.05	55.97	38.06
	Swahili	21.37	48.03	62.65	44.02	14.04	36.29	49.65	33.33
	Japanese	33.52	62.09	75.93	57.18	20.81	48.30	62.50	43.87
	Turkish	27.78	53.99	67.83	49.87	14.26	37.93	51.68	34.62
ClothoV2									
	Language	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
GLAP	English	<u>11.29</u>	29.86	42.49	27.88	10.09	27.89	40.13	26.04
	Portuguese	9.57	23.64	34.55	22.58	8.04	23.18	33.19	21.47
	Spanish	9.38	26.32	37.22	24.31	<u>8.73</u>	24.88	36.44	23.35
	French	11.10	25.93	37.22	24.75	8.71	<u>25.42</u>	<u>36.63</u>	<u>23.59</u>
	Russian	5.36	14.55	23.06	14.32	3.06	9.97	15.22	9.42
	Arabic	9.86	25.65	36.17	23.89	6.89	20.21	30.41	19.17
	Hindi	7.94	23.44	34.93	22.11	7.48	21.36	31.39	20.08
	German	10.43	<u>28.52</u>	<u>38.18</u>	<u>25.71</u>	8.59	24.48	36.15	23.07
	Chinese(zh)	5.36	13.40	19.52	12.76	2.03	5.86	9.47	5.79
	Swahili	7.75	20.77	32.15	20.22	5.91	16.42	25.72	16.02
	Japanese	12.44	26.03	35.41	24.63	7.85	22.30	32.48	20.87
	Turkish	8.04	20.48	30.62	19.71	5.91	18.76	28.31	17.66

Table 5.37: Complete GLAP retrieval results across evaluation languages on AudioCaps and ClothoV2. A2T and T2A correspond to Audio-to-Text and Text-to-Audio retrieval, respectively. **Bold** values indicate the best language for each metric and dataset, while underlined values indicate the second-best language.

to 40.76%, but more noticeable in Text-to-Audio retrieval, increasing from 33.53% to 36.08%. These results indicate that multilingual supervision does not simply recover non-English capabilities, but can also improve the quality of the English audio-text embedding space.

A similar trend is observed in zero-shot classification. PALMA Random ML improves the English ESC-50 accuracy from 83.50% to 90.12%, and the UrbanSound8K accuracy from 71.44% to 74.92%. This shows that the benefits of multilingual supervision transfer beyond retrieval and also improve the alignment between audio representations and English class prompts. Therefore, the multilingual training signal appears to act as

Model	Language	ESC-50	UrbanSound8K
PALMA Random Sampling	English	90.12 $\pm$ 1.35	74.92 $\pm$ 4.60
	Portuguese	<u>89.85</u> $\pm$ 0.94	71.87 $\pm$ 5.95
	Spanish	89.15 $\pm$ 1.51	74.61 $\pm$ 6.29
	French	86.70 $\pm$ 1.11	76.01 $\pm$ 6.05
	Russian	80.88 $\pm$ 2.53	79.30 $\pm$ 1.45
	Arabic	85.28 $\pm$ 1.80	72.16 $\pm$ 3.35
	Hindi	83.03 $\pm$ 0.55	69.38 $\pm$ 3.33
	German	88.23 $\pm$ 1.19	<u>79.23</u> $\pm$ 2.27
	Chinese(zh)	86.65 $\pm$ 1.53	67.98 $\pm$ 2.58
	Swahili	85.17 $\pm$ 1.98	75.37 $\pm$ 4.13
	Japanese	88.22 $\pm$ 0.52	78.34 $\pm$ 5.21
	Turkish	86.87 $\pm$ 1.55	66.45 $\pm$ 2.45
ML-CLAP	English	95.70 $\pm$ 0.66	83.84 $\pm$ 3.80
	Spanish	89.20 $\pm$ 0.43	81.68 $\pm$ 3.98
	French	87.00 $\pm$ 0.81	79.04 $\pm$ 3.10
	German	<u>91.20</u> $\pm$ 0.75	<u>82.25</u> $\pm$ 3.90
	Chinese(zh)	81.15 $\pm$ 1.30	69.71 $\pm$ 3.49
	Japanese	77.30 $\pm$ 0.90	73.52 $\pm$ 3.17
GLAP	English	81.10 $\pm$ 1.91	51.40 $\pm$ 1.39
	Portuguese	69.65 $\pm$ 0.73	45.93 $\pm$ 3.24
	Spanish	<u>80.25</u> $\pm$ 0.96	59.71 $\pm$ 3.35
	French	<u>75.70</u> $\pm$ 0.48	60.38 $\pm$ 5.13
	Russian	30.40 $\pm$ 1.03	41.33 $\pm$ 2.94
	Arabic	56.00 $\pm$ 0.94	66.33 $\pm$ 3.56
	Hindi	63.10 $\pm$ 1.15	55.96 $\pm$ 6.32
	German	69.90 $\pm$ 1.40	63.75 $\pm$ 3.30
	Chinese(zh)	58.50 $\pm$ 0.61	62.33 $\pm$ 1.72
	Swahili	52.75 $\pm$ 1.47	39.81 $\pm$ 2.51
	Japanese	62.85 $\pm$ 1.85	<u>66.17</u> $\pm$ 2.21
	Turkish	51.60 $\pm$ 1.07	56.77 $\pm$ 3.57

Table 5.38: Complete zero-shot classification performance for PALMA Random Sampling, ML-CLAP, and GLAP across evaluation languages on ESC-50 and UrbanSound8K. Values are reported as mean $\pm$ standard deviation. PALMA values are computed across three runs, while ML-CLAP and GLAP values are computed across folds. **Bold** values indicate the best language for each dataset within each model, while underlined values indicate the second-best language within each model.

a form of semantic regularization, encouraging the model to learn more robust audio-language correspondences instead of overfitting to the lexical structure of the original English captions.

When compared with external English-only baselines, PALMA Random ML is not always the strongest model in absolute terms. For example, ML-CLAP achieves

the best English zero-shot classification results, and some retrieval baselines remain stronger on specific datasets. However, the key observation is that the selected multilingual PALMA configuration outperforms its English-only counterpart while remaining competitive with several stronger, more costly baselines. This is relevant because PALMA achieves these gains by fine-tuning only the text encoder while keeping the audio encoder frozen.

English Retrieval Evaluation								
Model	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
AudioCaps								
MS-CLAP V2 (HTS-AT + GPT2) [50]	30.03	64.34	78.18	57.52	14.08	44.77	60.83	39.90
LAION-CLAP (HTS-AT + RoBERTa) [201]	46.68	79.75	90.55	72.33	34.58	70.80	83.69	63.02
LAION-CLAP (HTS-AT + RoBERTa + Fusion) [201]	44.66	78.52	90.44	71.20	34.60	71.32	84.39	63.43
WavCaps (HTS-AT + BERT) [127]	52.87	82.23	89.76	74.95	52.87	82.23	<u>89.76</u>	74.95
Auto-ACD (HTS-AT + RoBERTa) [183]	53.77	83.24	92.24	76.42	40.54	75.16	85.85	67.18
BLAT (CNN14 + BERT) [204]	43.64	74.58	85.49	67.90	33.72	68.84	83.06	61.87
Cacophony (AudioMAE + RoBERTa) [225]	48.26	79.64	90.44	72.78	37.30	71.45	84.27	64.34
AudioSetCaps (HTS-AT + RoBERTa) [11]	59.28	86.61	93.48	79.79	<u>46.59</u>	<u>80.31</u>	90.30	<u>72.40</u>
ML-CLAP (CED + SONAR-TE) [207]	<u>56.13</u>	83.13	91.68	<u>76.98</u>	40.70	75.77	87.22	67.90
GLAP (Dasheng [41] + SONAR-TE) [40]	46.34	78.85	88.53	71.24	26.86	61.12	76.81	54.93
PALMA English-only (CED + mGTE)	49.31 ± 1.58	81.14 ± 1.50	91.08 ± 1.11	73.84 ± 1.03	38.58 ± 0.23	73.85 ± 0.77	85.54 ± 0.39	65.99 ± 0.44
PALMA Random ML (CED + mGTE)	53.81 ± 0.81	83.61 ± 0.51	93.25 ± 1.08	76.89 ± 0.43	39.59 ± 0.75	75.33 ± 0.75	87.16 ± 0.24	67.36 ± 0.45
ClothoV2								
Model	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
MS-CLAP V2 (HTS-AT + GPT2) [50]	24.78	48.13	61.44	44.78	15.67	38.78	51.44	35.30
LAION-CLAP (HTS-AT + RoBERTa) [201]	21.34	47.66	61.53	43.51	16.50	39.44	51.75	35.90
LAION-CLAP (HTS-AT + RoBERTa + Fusion) [201]	21.82	48.23	62.30	44.11	16.44	39.79	52.46	36.23
WavCaps (HTS-AT + BERT) [127]	21.91	49.47	63.35	44.91	18.78	45.15	57.72	40.55
Auto-ACD (HTS-AT + RoBERTa) [183]	17.61	39.23	51.29	36.04	15.62	38.30	51.92	35.28
BLAT (CNN14 + BERT) [204]	17.61	40.00	52.73	36.78	12.73	34.81	48.19	31.91
Cacophony (AudioMAE + RoBERTa) [225]	24.02	51.87	64.69	46.86	19.50	45.32	57.86	40.89
AudioSetCaps (HTS-AT + RoBERTa) [11]	17.99	35.98	49.19	34.39	15.08	37.09	50.12	34.10
ML-CLAP (CED + SONAR-TE) [207]	29.38	54.55	67.56	50.49	23.67	51.10	64.75	46.51
GLAP (Dasheng+ SONAR-TE) [40]	11.29	29.86	42.49	27.88	10.09	27.89	40.13	26.04
PALMA English-only (CED + mGTE)	19.68 ± 0.80	43.92 ± 0.51	57.38 ± 0.64	40.33 ± 0.60	14.76 ± 0.21	36.42 ± 0.36	49.42 ± 0.25	33.53 ± 0.19
PALMA Random ML (CED + mGTE)	21.05 ± 1.23	43.68 ± 0.48	57.57 ± 1.07	40.76 ± 0.92	16.31 ± 0.20	39.39 ± 0.35	53.56 ± 0.25	36.08 ± 0.17

Table 5.39: *English retrieval performance on AudioCaps and ClothoV2. A2T and T2A denote Audio-to-Text and Text-to-Audio retrieval, respectively. Values are reported as Recall@1, Recall@5, Recall@10, and Recall@Avg. PALMA values are reported as mean ± standard deviation across three runs. The best value for each metric within each dataset is shown in **bold**, while the second-best value is underlined. PALMA Random ML results are also shown in **bold** to highlight the selected multilingual PALMA configuration.*

Summary

Overall, the baseline comparison shows that PALMA provides a strong multilingual trade-off. ML-CLAP remains the strongest model on ClothoV2 retrieval and English classification, while PALMA with Random Sampling achieves the best multilingual retrieval performance on AudioCaps and the best multilingual classification performance on ESC-50. Compared with GLAP, PALMA also shows substantially stronger retrieval and classification performance across the evaluated multilingual settings.

Model	ESC-50	UrbanSound8K
MS-CLAP V2 (HTS-AT + GPT2) [50]	<u>94.35</u>	<u>82.38</u>
LAION-CLAP (HTS-AT + RoBERTa) [201]	83.10	80.99
LAION-CLAP (HTS-AT + RoBERTa + Fusion) [201]	79.05	71.24
WavCaps (HTS-AT + BERT) [127]	94.25	82.28
Auto-ACD (HTS-AT + RoBERTa) [183]	82.35	75.46
BLAT (CNN14 + BERT) [204]	77.45	73.91
Cacophony (AudioMAE + RoBERTa) [225]	92.75	80.72
AudioSetCaps (HTS-AT + RoBERTa) [11]	52.00	51.65
ML-CLAP (CED + SONAR-TE) [207]	95.70	83.84
GLAP (Dasheng + SONAR-TE) [40]	81.10	51.40
PALMA English-only (CED + mGTE)	83.50 $\pm$ 2.85	71.44 $\pm$ 5.10
PALMA Random ML (CED + mGTE)	90.12 $\pm$ 1.35	74.92 $\pm$ 4.60

Table 5.40: *English zero-shot classification performance on ESC-50 and UrbanSound8K. Values are reported as mean accuracy. PALMA values are reported as mean $\pm$ standard deviation across three runs. The best value for each dataset is shown in **bold**, while the second-best value is underlined. PALMA Random ML results are also shown in **bold** to highlight the selected multilingual PALMA configuration.*

An important difference between PALMA and the multilingual baselines is the training cost. Both ML-CLAP and GLAP fine-tune their audio and text encoders jointly and are trained for 20 epochs, whereas the PALMA configurations are trained for only 10 epochs. In addition, PALMA follows a more parameter-efficient adaptation strategy, since it fine-tunes only the text encoder while keeping the audio encoder frozen. Despite this more constrained and computationally cheaper setup, PALMA achieves competitive, and in some cases superior, performance compared with more costly multilingual baselines.

These results reinforce one of the main hypotheses of this dissertation: carefully designed architectural choices, combined with simple and efficient training strategies, can produce strong multimodal alignment without requiring full model fine-tuning or substantially larger training budgets. In this sense, PALMA demonstrates that multilingual audio-language alignment can be improved through a lightweight adaptation strategy, while still remaining competitive with more expensive multilingual audio-language models.

5.3.9 Final Remarks

The results presented in this section show that PALMA provides a systematic view of the main factors that influence multilingual audio-language alignment. Across the different experimental phases, the analysis showed that performance depends not only on the isolated strength of the audio and text encoders, but also on the compatibility between their representation spaces. In this sense, PALMA highlights the importance of encoder synergy: stronger multilingual text encoders can enrich the shared embedding space and improve retrieval performance when paired with suitable acoustic encoders. The exper-

iments also showed that efficient training strategies, such as Locked Audio Pretraining, can preserve useful acoustic prior knowledge while reducing the cost of adaptation.

A central finding of PALMA is that pretrained acoustic encoders organize information hierarchically across their layers. The Mixture-of-Layers analysis showed that final layers remain highly important for audio-language alignment, but intermediate layers still contain relevant information that can contribute to the shared representation space. Therefore, the main relevance of MoL is not only the moderate performance gains observed in some configurations, but also the evidence that audio-language models can benefit from mechanisms that exploit the internal structure of pretrained encoders. This suggests that future progress in this domain may come from methods that use intermediate acoustic representations more effectively, instead of relying exclusively on the final encoder layer.

Finally, the multilingual supervision experiments showed that translated captions are an effective strategy for extending audio-language alignment beyond English. English-only training produced strong English results, but it led to substantial degradation under multilingual evaluation, indicating that a multilingual text encoder alone is not sufficient after audio-text fine-tuning. In contrast, explicit multilingual supervision substantially improved retrieval and zero-shot classification across languages. Random Sampling emerged as the strongest overall strategy, showing that a simple multilingual supervision method can match or surpass more complex and computationally costly alternatives. Overall, PALMA reinforces the idea that scalable multilingual audio-language learning depends on careful encoder selection, efficient adaptation, effective use of pre-trained representations, and simple but well-designed multilingual training strategies.

Conclusion

In this research, we investigated how multilingual multimodal representations of audio and text can be learned under data and computational constraints. The central motivation of this work was the strong dependence of audio-language models on English-centric resources and the high computational cost of large-scale training pipelines. Together, these factors limit the accessibility of audio-language models and reduce their applicability in multilingual real-world scenarios. To address these problems, we explored two complementary directions: CACARA, which investigates the efficient integration of audio into an existing multilingual image-text representation space, and PALMA, which studies the systematic design of bimodal multilingual audio-language models.

The results obtained with CACARA show that audio can be efficiently added to an existing frozen multilingual multimodal representation space. By aligning the audio encoder with a frozen textual anchor, CACARA enabled emergent multimodal alignment between audio and vision without requiring explicit audio-image training pairs or full tri-modal retraining. Moreover, the same mechanism also supported the emergence of multilingual audio-text alignment, showing that the newly added audio modality can inherit part of the multilingual structure already encoded in the pretrained representation space. These results support the hypothesis that a new modality can benefit from both the multimodal and multilingual organization of a pretrained multimodal model. At the same time, we showed that this emergent transfer is not uniform across languages. To describe this behavior, we introduced the concept of the cross-lingual gap, which captures the differences in alignment quality between English and non-English languages after the addition of the new modality. This finding indicates that the semantic organization of languages before the modality expansion is an important factor in determining how well the newly added modality aligns with non-English language representations.

The results obtained with PALMA complement this analysis by showing that audio-language alignment depends strongly on the choice of encoder, training strategy, data composition, batch size, and multilingual supervision. Among the evaluated configurations, the results show a clear synergy between acoustic and textual representations: the quality of audio-language alignment depends not only on the strength of each encoder in

isolation, but also on how compatible their representation spaces are. In particular, more recent and powerful multilingual text encoders have enriched the shared embedding space and improved alignment performance when paired with suitable acoustic encoders. At the same time, Locked Audio Pretraining (LaT) provided an effective performance-efficiency trade-off by freezing the audio encoder while preserving strong zero-shot classification performance. These results reinforce the view that the design of audio-language models should be understood not only as a matter of scaling data or model size, but also as a problem of selecting compatible pretrained components and choosing training strategies that preserve useful prior knowledge while still allowing effective cross-modal adaptation.

The proposed Mixture-of-Layers (MoL) mechanism further showed that pretrained acoustic encoders organize their representations hierarchically, with different encoder depths contributing differently to audio-language alignment. Although the final layers remained highly important, the learned layer weights indicated that intermediate representations still contain relevant information for the alignment process. Therefore, one of the main insights from MoL is not only its modest performance gain but also the evidence that audio-language models can benefit from mechanisms that make better use of the internal structure of pretrained acoustic encoders, suggesting a promising path forward for the development of more efficient methods that exploit intermediate representations instead of relying exclusively on the final encoder layer.

Finally, the multilingual supervision experiments showed that translated captions are an effective strategy for extending audio-language alignment beyond English. While English-only training produced strong English results, it led to severe degradation on multilingual evaluation, demonstrating that a multilingual text encoder alone is insufficient after audio-text fine-tuning. In contrast, explicit multilingual supervision substantially improved retrieval and zero-shot classification across languages. Random Sampling emerged as the strongest overall strategy, demonstrating that a simpler multilingual supervision method can match or surpass more complex, computationally costly multilingual mixture strategies, highlighting the importance of simple yet effective design choices for scalable multilingual audio-language learning.

Taken together, the findings of this work support the initial hypothesis that multilingual multimodal alignment can be achieved through efficient and modular design. CACARA and PALMA show that competitive multilingual audio-language systems do not necessarily require full end-to-end retraining of all modalities or prohibitively expensive large-scale training pipelines. Instead, strong alignment can be obtained by reusing pretrained encoders, preserving useful multilingual and acoustic prior knowledge, applying efficient adaptation strategies, and carefully designing the representation-learning process. In this sense, this research reinforces the idea that accessibility and scalability in multilingual multimodal learning depend not only on increasing model and dataset size,

but also on developing methods that make better use of the representation spaces already learned by existing models.

6.1 Answering Research Questions

This section revisits the research questions introduced in Section 1.3 and summarizes how they were addressed throughout this dissertation. The methodological answers are mainly developed in Chapter 4, while the empirical evidence is presented in Chapter 5. Overall, the results support the central hypothesis of this work: multilingual audio-text and multimodal alignment can be learned under data and computational constraints by reusing pretrained models, carefully selecting encoders, employing efficient training strategies, and using representation-level aggregation mechanisms. At the same time, the experiments also reveal important limitations, especially regarding the consistency of multilingual transfer across languages, datasets, and downstream tasks.

6.1.1 RQ1: How can multilingual audio-text alignment be achieved under English-centric data constraints?

The results show that multilingual audio-text alignment under English-centric data constraints can be achieved through two complementary mechanisms: emergent transfer from a frozen multilingual representation space and explicit multilingual supervision using translated captions.

CACARA demonstrates the first mechanism by aligning an audio encoder with English text representations produced by a frozen pretrained multilingual text encoder, as described in Sections 4.1.2 and 5.2.4. Although only English audio-text pairs are used during explicit alignment training, the audio encoder is connected to a textual representation space that already organizes multiple languages according to their semantic relationships. Consequently, part of this cross-lingual structure is transferred to the newly aligned audio modality. The multilingual evaluation presented in Section 5.2.5 confirms that this process enables audio-text retrieval and zero-shot audio classification in languages that were not observed during training.

This transfer, however, is not equally effective across all evaluation conditions. As shown by the cross-lingual gap analysis in Section 5.2.5, performance varies across languages, datasets, retrieval directions, and downstream tasks. These differences indicate that the multilingual structure inherited from the frozen text encoder provides a viable basis for cross-lingual audio-text interaction, but does not ensure the same degree of alignment for every language. The effectiveness of emergent multilingual transfer depends on

the linguistic structure already encoded by the language anchor and on how successfully that structure can be propagated to the audio modality.

PALMA complements this emergent strategy by investigating explicit multilingual supervision based on translated captions. The experiments presented in Section 5.3.8 show that machine-translated captions provide a practical means of extending audio-text supervision beyond English when large native multilingual audio-text datasets are unavailable. The retrieval results in Section 5.3.8 and the zero-shot classification results in Section 5.3.8 demonstrate that translated multilingual supervision strengthens cross-lingual alignment and improves generalization across languages.

Therefore, RQ1 is answered by showing that multilingual audio-text alignment can be achieved through two complementary strategies: emergent alignment with a frozen multilingual textual anchor and explicit multilingual supervision using translated captions. As demonstrated by CACARA, emergent alignment provides a low-cost mechanism for extending audio-text capabilities to languages not observed during training. As demonstrated by PALMA, translated multilingual supervision offers a more direct and generally more robust means of strengthening cross-lingual alignment.

6.1.2 RQ2: Can a new modality be efficiently integrated into an existing multilingual multimodal representation space?

The results demonstrate that a new modality can be efficiently integrated into an existing multilingual multimodal representation space without jointly retraining all participating modality encoders. CACARA achieves this integration by preserving the pretrained image and text encoders and optimizing only the newly introduced audio encoder, as described in Section 4.1.2. The frozen multilingual text encoder serves as the alignment anchor through which the audio representations are mapped into the pre-established image-text semantic space.

Because the image and text encoders are already aligned, bringing audio representations closer to the textual representations also establishes an indirect relationship between audio and images. This relationship emerges even though paired audio-image examples are not used during training. The results discussed in Section 5.2.4 show that the aligned audio encoder can interact with both textual and visual representations, supporting the central premise that previously learned cross-modal relationships can be reused when incorporating a new modality.

The same alignment process also transfers part of the multilingual structure of the frozen text encoder to the audio modality. The multilingual results in Section 5.2.5 demonstrate that modality integration does not produce only an English audio-text connection. Instead, the newly aligned audio encoder inherits cross-lingual relationships that

were already represented in the pretrained image-text space. CACARA therefore supports both forms of emergence investigated in this dissertation: multilingual transfer from text to audio and cross-modal transfer between audio and images.

The efficiency analysis presented in Section 5.2.6 further shows that this sequential strategy reduces the computational requirements of multimodal expansion. Rather than jointly realigning the complete architecture, CACARA reuses the existing multilingual image-text space and concentrates optimization on the new audio encoder tower. This lowers memory and computational requirements, preserves the previously learned image-text relationships, and makes the integration of new modalities more modular.

These efficiency gains are accompanied by performance trade-offs. Emergent alignment does not ensure that every cross-modal or cross-lingual relationship will be transferred with the same strength. The results presented in Sections 5.2.4 and 5.2.5 show that performance remains dependent on the target language, modality combination, dataset, and downstream task.

Therefore, RQ2 is answered positively. A new audio modality can be efficiently integrated into an existing multilingual multimodal representation space through frozen pretrained encoders and text-anchored sequential alignment. CACARA demonstrates that the audio encoder can inherit both multilingual and multimodal relationships from the pre-established semantic space without retraining the original image and text towers. This provides a more modular, computationally accessible, and environmentally favorable alternative to full multimodal retraining, while retaining the limitation that indirectly transferred relationships may remain weaker or less consistent than relationships learned through direct supervision.

6.1.3 RQ3: How do encoder choice and training strategy affect the performance and efficiency of multilingual audio-language representation learning?

The PALMA experiments show that encoder choice and training strategy jointly determine the performance and efficiency of multilingual audio-language representation learning. The strongest configuration cannot be identified solely by selecting the audio encoder or text encoder with the best isolated unimodal performance. Instead, successful audio-text alignment depends on the compatibility between the representation spaces produced by both encoders and on how these spaces respond to the adopted training and optimization strategy.

The audio-encoder consolidation experiments presented in Section 5.3.2 demonstrate that pretrained audio backbones differ substantially in their ability to support audio-text retrieval and zero-shot classification. These differences reflect not only model size

or unimodal performance, but also the type of information acquired during pretraining and its suitability for semantic alignment with text. Similarly, the text-encoder analysis in Section 5.3.3 shows that multilingual text encoders produce representation spaces with different degrees of compatibility with the evaluated audio encoders.

These findings establish representational synergy as an important factor in audio-language model design. The performance of an encoder pair emerges from the interaction between both towers rather than from a simple combination of their individual capabilities. Consequently, an encoder that performs strongly in isolation does not necessarily produce the strongest multimodal model when paired with another modality. Audio and text encoders must therefore be evaluated jointly under the intended cross-modal objective.

Training strategy further influences this interaction. The experiments on locked-encoder training and alternative contrastive learning objectives in Section 5.3.4 show that full fine-tuning, freezing, and efficient adaptation produce different balances between retrieval performance, zero-shot transfer, and computational cost. Full fine-tuning provides greater flexibility for adapting both representation spaces to the audio-text objective, but it requires more memory and computation and may modify useful pretrained structures. Locked-encoder strategies reduce the number of trainable parameters and preserve the representation space of the frozen tower, although this restriction can limit the degree of cross-modal adaptation. The experiments also show that alternative contrastive objectives yield configuration-dependent benefits rather than consistent improvements across all encoders and tasks. This variability may arise from differences in supervision scale and data characteristics between the domains in which these objectives were originally proposed and the settings considered in this work.

The broader PALMA optimization pipeline demonstrates that encoder and training-strategy selection cannot be separated from the remaining experimental configuration. The analyses presented in Sections 5.3.2, 5.3.6, and 5.3.7 show that learning rate, regularization, training-data composition, and batch size substantially affect the final model. Larger datasets, batches, or model configurations do not automatically produce better performance when they introduce noisy pairs, alter the distribution of contrastive negatives, or are not accompanied by appropriate optimization settings. Conversely, carefully selected configurations can achieve strong results without relying on the most computationally expensive alternative.

Explicit multilingual supervision introduces an additional design dimension. The results in Section 5.3.8 show that incorporating translated captions modifies the cross-lingual structure learned by the model and improves performance beyond English. However, the resulting gains depend on how languages are sampled and combined during training. Multilingual supervision must therefore be treated as part of the model-design

and optimization process rather than as a simple increase in the quantity of data.

Therefore, RQ3 is answered by showing that multilingual audio-language representation learning depends on the joint design of the audio encoder, text encoder, training objective, adaptation strategy, optimization settings, data composition, batch size, and multilingual supervision procedure. Encoder compatibility is more informative than isolated encoder quality, and efficient training strategies involve trade-offs among adaptability, preservation of pretrained knowledge, predictive performance, and computational cost. PALMA therefore demonstrates that robust and efficient audio-language models are best achieved through the systematic co-design of their encoders, training strategies, optimization settings, and data composition.

6.1.4 RQ4: What is the impact of cross-layer aggregation on multilingual audio-text representations?

The results show that cross-layer aggregation can improve multilingual audio-text representations by exploiting complementary information distributed across the hierarchy of pretrained audio encoders. Although the final encoder layer provides a strong semantic representation, it does not necessarily retain all relevant information that may remain useful for cross-modal alignment.

The Mixture-of-Layers (MoL) mechanism introduced in PALMA and evaluated in Section 5.3.5 addresses this limitation by learning to combine representations from multiple encoder layers. Rather than assuming that the final layer is universally optimal, MoL allows information from different layers to contribute to the audio representation used in the shared audio-text space. The main results presented in Section 5.3.5 show that cross-layer aggregation can yield modest improvements across downstream tasks relative to representations derived exclusively from the final layer.

The layer-weight importance analysis presented in Section 5.3.5 provides further evidence supporting the use of this approach. More importantly than isolated performance gains, the learned layer contributions reveal that intermediate representations remain relevant to audio-language alignment even when deeper layers receive substantial importance. This finding indicates that useful information is distributed across different levels of the encoder hierarchy and that early layers can retain residual relevant information that can complement the more abstract semantic information encoded in the final layers.

The dual-encoder and projection-only ablations presented in Section 5.3.5 further demonstrate that the effectiveness of cross-layer aggregation depends on where and how the aggregated information is incorporated into the model. Increasing architectural complexity alone is not sufficient; the aggregation mechanism must enable intermediate features to contribute meaningfully to the final aligned representation. These analyses also

suggest that some of the observed limitations may arise from the design of the aggregation and projection mechanisms, rather than from an absence of useful information within the pretrained encoder itself.

Therefore, RQ4 is answered by showing that cross-layer aggregation is a promising and empirically supported strategy for improving multilingual audio-text representation learning. The MoL mechanism demonstrates that early to intermediate layers can retain complementary information relevant to retrieval and zero-shot classification and that relying exclusively on the final layer may overlook useful aspects of the pretrained representation hierarchy. Nevertheless, its benefits remain configuration-dependent, and cross-layer aggregation should be interpreted as a representation-aware design strategy rather than as a universally performance-improving solution. More importantly, the findings establish a foundation for future research on aggregation mechanisms that can better exploit the complementary information distributed across encoder layers.

6.1.5 Summary

Taken together, the answers to the four research questions show that efficient multilingual multimodal learning can be achieved through the reuse and careful adaptation of pretrained representation spaces. CACARA demonstrates that aligning a new audio encoder with a frozen multilingual textual anchor enables multilingual audio-text capabilities and emergent audio-image interactions without requiring full multimodal retraining. PALMA demonstrates that the quality and efficiency of audio-language alignment depend on representational synergy between encoders, appropriate training and optimization strategies, carefully composed multilingual supervision, and the effective use of the internal hierarchy of pretrained audio representations.

The findings also establish important boundaries for these conclusions. Emergent multilingual and multimodal transfer remains uneven across languages, modality combinations, datasets, and tasks. Translated captions provide an effective and practical source of multilingual supervision, but do not completely replace native multilingual resources. Efficient adaptation strategies reduce computational requirements, but may introduce performance trade-offs relative to full fine-tuning. Finally, intermediate encoder layers preserve complementary information that remains valuable for audio-language alignment. However, the effectiveness of cross-layer aggregation varies across model configurations.

Overall, the combined evidence from CACARA and PALMA supports the central premise of this dissertation: multilingual audio-text and multimodal representation learning do not depend exclusively on large native multilingual datasets or computationally expensive joint retraining. Instead, competitive and more accessible models can be developed by reusing pretrained semantic spaces, selecting compatible encoder combi-

nations, balancing predictive performance and efficiency during training, and exploiting complementary information distributed throughout pretrained representation hierarchies. The remaining limitations and the research opportunities derived from these findings are discussed in Sections 6.2 and 6.3.

6.2 Limitations

Despite the positive results obtained, this research has limitations that must be acknowledged. The limitations presented below do not invalidate the main findings of the work. Instead, they define the scope within which the results should be interpreted and establish important boundaries on the conclusions that can be drawn.

Dependence on Translated Text. Although translated captions proved effective for multilingual supervision in PALMA and enabled multilingual evaluation in both CACARA and PALMA, they are not equivalent to native multilingual audio descriptions. Machine translation may introduce linguistic artifacts, reduce cultural specificity, and generate captions that preserve the meaning of the English source while failing to reflect how speakers of different languages would naturally describe the same acoustic event. This can, in turn, introduce language-specific biases into both training and evaluation. Consequently, the multilingual evaluation conducted in this work should be understood as an assessment of cross-lingual transfer under translated-text conditions, rather than as a full substitute for evaluation on native multilingual audio-text datasets.

Language Coverage. This work evaluates twelve languages, including English and eleven translated languages, which provides a broader perspective than studies restricted to English-only evaluation. However, this set still captures only a limited portion of the world’s linguistic diversity. Moreover, the evaluated languages are not equally represented in the pretraining data of the multilingual text encoders adopted in this dissertation. As a result, the observed performance differences may reflect not only the quality of the proposed alignment methods, but also disparities in the underlying multilingual capacity of the pretrained text encoders, as well as variation in the quality of the translated captions.

Computational Scale. One of the goals of this work was to investigate efficient alternatives under realistic resource constraints. Consequently, some experiments were intentionally limited in scale. Not all combinations of encoders, datasets, batch sizes, loss functions, and aggregation strategies could be explored exhaustively. Broader large-scale experiments may reveal different behaviors, particularly for objectives such as SigLIP, which may benefit from larger datasets, larger batch sizes, and longer training schedules.

Insights from Mixture-of-Layers. The analysis based on Mixture-of-Layers provides useful insight into how different encoder layers contribute to alignment, but it does not fully explain the internal mechanisms of the learned representation space. The learned weights indicate which layers the model emphasizes, but they do not reveal which acoustic or semantic properties are encoded at each layer. A deeper understanding of these mechanisms would require more targeted probing experiments, controlled synthetic data, or layer-wise evaluations across additional tasks.

6.3 Future Work

Future studies should evaluate CACARA and PALMA on native multilingual audio-text datasets, as such resources become available, and also contribute to the construction of new benchmarks with human-written captions in multiple languages. This would enable a more realistic assessment of multilingual audio-language alignment beyond translated-text conditions and would help determine how well the proposed methods generalize to naturally occurring cross-lingual variation.

Next, future work should investigate multilingual supervision strategies beyond Random Sampling and LE-Mixture. Possible directions include curriculum-based language scheduling, language-balanced batching, quality-aware filtering of translated captions, and adaptive sampling strategies guided by language difficulty or performance disparities. These alternatives may improve multilingual alignment further, particularly for languages that are less represented in the underlying pretrained text encoders.

Also, the evaluation should be extended to additional downstream tasks, such as audio captioning, audio question answering, sound event localization, and long-form audio retrieval. Exploring these tasks would provide a broader understanding of whether the learned representations support not only retrieval and zero-shot classification, but also more complex forms of audio-language interaction and reasoning.

The MoL mechanism can be extended through more expressive aggregation strategies. Future developments may combine layer aggregation with token-level or temporal aggregation, allowing the model to determine not only which encoder layers are most informative, but also which temporal regions of the audio are most relevant to a given textual query. Such extensions may improve fine-grained retrieval and help address cases in which multiple sound events occur at different times within the same recording.

Future work should continue exploring efficient multimodal expansion. The results obtained with CACARA suggest that new modalities can be integrated into frozen multilingual multimodal spaces through sequential alignment strategies. This idea could be extended beyond audio to modalities such as video, motion, sensor data, or symbolic music representations. Investigating these directions would help assess the

generality of emergent alignment as a scalable strategy for building broader multilingual multimodal systems.

6.4 Generative AI Use Disclosure

The author used generative artificial intelligence tools, specifically Large Language Models, only to polish the presentation of portions of this dissertation. The author assumes full responsibility for the integrity of this dissertation.

6.5 Published and Under-Review Papers

We list the publications produced during the development of this work, presented in chronological order. The name of the author of this dissertation is highlighted in **bold** and underlined. An asterisk (*) denotes equal contribution (co-first authorship).

6.5.1 Related to the Dissertation

- Moreira, D. A., **Ferreira, A. I.**, Silva, J., dos Santos, G. O., Pereira, L., Gondim, J. M., ... & Avila, S. (2024). *FairPIVARA: Reducing and Assessing Biases in CLIP-Based Multimodal Models*. In: *Workshop on Privacy, Fairness, Accountability and Transparency in Computer Vision, British Machine Vision Conference (BMVC) 2024*.
- Moreira, D. A.*, **Ferreira, A. I.***, Maia, H. A., da Silva, N. F. F., Scarton, C., Pedrini, H., and Avila, S. *CACARA: Cross-Modal Alignment Leveraging a Text-Centric Approach for Cost-Effective Multimodal and Multilingual Learning*. Under review at the *Symmetry and Geometry in Neural Representations (NeurReps) Workshop at NeurIPS 2026*.
- **Ferreira, A. I. S.***, Moreira, D. A. B.*, da Silva, N. F. F., & Avila, S. *PALMA: Pretrained Audio–Language Multilingual Alignment via Locked-Audio Tuning with Mixture of Layers*. Under review at *ICASSP 2027 – IEEE International Conference on Acoustics, Speech and Signal Processing*.

6.5.2 Collaborations

- Silva, J., **Ferreira, A.**, Moreira, D., Santos, G., Bonil, G., Gondim, J., Pereira, L., Maia, H., Silva, N., Hashiguti, S., Avila, S., & Pedrini, H. (2024). *Avaliação de Ferramentas de Ética no Levantamento de Considerações Éticas de Modelos de*

Linguagem em Português. In: *Anais da 1ª Conferência Latino-Americana de Ética em Inteligência Artificial*, (pp. 61-64). Porto Alegre.

- **Alef I. S. Ferreira**, José Mesquita-Neto, Nádia Félix Felipe da Silva, and Thierson Couto Rosa. *Beyond Species-Based Assessments: Automatic Acoustic Recognition of Pollinator Function*. In: *NeurIPS 2025 Workshop on AI for Non-Human Animal Communication*, 2025.
- **Alef I. S. Ferreira**, Luiz Fernando de Araújo Vidal, Lucas Rafael Stefanel Gris, and Arlindo Rodrigues Galvão Filho. *Multi-scale Transformer Language Modeling for Music Classification Tasks*. In: *1st Workshop on Large Language Models for Music & Audio (LLM4MA)*, an ISMIR 2025 Satellite Workshop, 2025.
- Marcelo Henrique Lopes Ferreira, **Alef I. S. Ferreira**, Lucas Rafael Stefanel Gris, Rodrigo Tanaka, et al. *Multimodal Disfluency Detection via Acoustic-Text Fusion: Comparing Intended and Verbatim ASR Transcripts*. In: *12th Workshop on Disfluency in Spontaneous Speech (DiSS 2025)*, an Interspeech 2025 Satellite Workshop, 2025.
- **Alef I. S. Ferreira**, Lucas Rafael Gris, Alexandre Ferro Filho, Lucas Ólives, et al. *Enhancing Speech Emotion Recognition with Graph-Based Multimodal Fusion and Prosodic Features for the Speech Emotion Recognition in Naturalistic Conditions Challenge at Interspeech 2025*, 2025.
- **Alef I. S. Ferreira**, Nádia Félix Felipe da Silva, Fernanda Neiva Mesquita, Thierson Couto Rosa, et al. *Transformer Models Improve the Acoustic Recognition of Buzz-Pollinating Bee Species*. *Ecological Informatics*, 2025.
- **Alef I. S. Ferreira**, Lucas Rafael Gris, Augusto Seben da Rosa, Frederico Santos de Oliveira, et al. *FreeSVC: Towards Zero-shot Multilingual Singing Voice Conversion*. In: *ICASSP 2025 – IEEE International Conference on Acoustics, Speech and Signal Processing*, 2025.
- Jhessica Silva, Diego A. B. Moreira, Gabriel O. dos Santos, **Alef I. S. Ferreira**, Helena Maia, Sandra Avila, and Helio Pedrini. *Evaluation of AI Ethics Tools in Language Models: A Developers' Perspective Case Study*. *AI and Ethics*, 2026.
- Frederico Santos de Oliveira, Lucas Rafael Stefanel Gris, **Alef I. S. Ferreira**, Augusto Seben da Rosa, et al. *Tagarela - A Portuguese Speech Dataset from Podcasts*. In: *ICASSP 2026 – IEEE International Conference on Acoustics, Speech and Signal Processing*, 2026.
- **Alef I. S. Ferreira**, Nádia Félix Felipe da Silva, Thierson Couto Rosa, and José

Mesquita-Neto. *Beyond Taxonomic-Based Assessments: AI-Based Acoustic Assessment of Bee Pollination Function*. *Smart Agricultural Technology*, 2026.

- Roberto Ahumada-García, David Zabala-Blanco, Víctor Hugo Monzón, Nádia Félix Felipe da Silva, Thierson Couto Rosa, **Alef I. S. Ferreira**, et al. *Machine Learning for Predicting Bee Honey Production and Quality: A Systematic Review*. *Computers and Electronics in Agriculture*, 2026.
- **Alef I. S. Ferreira**, Nádia Felix Felipe da Silva, Thierson Couto Rosa, and José Neiva Mesquita-Neto. *Beyond Taxonomic-Based Assessments: AI-Based Acoustic Assessment of Bee Pollination Function*. *Smart Agricultural Technology*, 2026.
- **Alef I. S. Ferreira**, Pedro Botelho, Fernanda Silva, Daniel Casanova, Rafael Faustino, Frederico Oliveira, Arlindo Galvão Filho, and Anderson da Silva Soares. *CAL-MOS: Bridging Layers with Adapters for Robust MOS Prediction Across Speech Foundation Models*. In: *Interspeech 2026*.
- **Alef I. S. Ferreira**, Lucas Hideki Ueda, Lucas Rafael Stefanel Gris, Frederico Santos de Oliveira, Augusto Seben da Rosa, Alexandre Costa Ferro Filho, and Victor Moreno. *Emotion-Aware Speaker Anonymization with Zero-Shot TTS Foundation Models for VoicePrivacy 2026*. In: *6th Symposium on Security and Privacy in Speech Communication (SPSC 2026) and VoicePrivacy Challenge Workshop*, co-located with Interspeech 2026.
- Lucas Rafael Stefanel Gris, **Alef I. S. Ferreira**, Frederico Santos de Oliveira, Augusto Seben da Rosa, Alexandre Costa Ferro Filho, Arlindo Rodrigues Galvão Filho, and Anderson da Silva Soares. *CVSS-X: A Multilingual Speech-to-Speech Translation Corpus for 28 Languages*. In: *Speech and Audio Language Models Workshop (SALMA 2026)*, at EMNLP 2026.
- **Alef I. S. Ferreira**, Lucas Rafael Stefanel Gris, Frederico Santos de Oliveira, Arlindo Rodrigues Galvão Filho, and Anderson da Silva Soares. *Echoes in Plato's Cave: Measuring Global and Local Alignment Between Speech and Language Representations*. In: *Speech and Audio Language Models Workshop (SALMA 2026)*, at EMNLP 2026.
- **Alef I. S. Ferreira**, Lucas Rafael Stefanel Gris, Frederico Santos de Oliveira, Christopher Dane Shulby, Luiz Fernando de Araújo Vidal, Anderson da Silva Soares, and Arlindo Rodrigues Galvão Filho. *Mask, Sample, Revise: A Revisable CTMC Inference Stack for Guided Discrete Flow Matching Text-to-Speech*. In: *Spoken Language Technologies (SLT) 2026*.
- Lucas Rafael Stefanel Gris, Daniel Casanova, Frederico Santos de Oliveira,

Alef I. S. Ferreira, Beatriz Almeida Felício, Raul César Reis Mata, Anderson da Silva Soares. *Cloned Voices, Real Consequences: Evaluating Bias in Political Deepfake Detection for Electoral Integrity in Brazil*. Under review at the *Trustworthy AI for Good (AI4GOOD) Workshop at NeurIPS 2026*.

Bibliography

- [1] AGOSTINELLI, A.; DENK, T. I.; BORSOS, Z.; ENGEL, J.; VERZETTI, M.; CAILLON, A.; HUANG, Q.; JANSEN, A.; ROBERTS, A.; TAGLIASACCHI, M.; OTHERS. **Musiclm: Generating music from text.** *arXiv preprint arXiv:2301.11325*, 2023.
- [2] ALAYRAC, J.-B.; RECASENS, A.; SCHNEIDER, R.; ARANDJELOVIĆ, R.; RAMAPURAM, J.; DE FAUW, J.; SMAIRA, L.; DIELEMAN, S.; ZISSERMAN, A. **Self-Supervised Multimodal Versatile Networks.** *Advances in Neural Information Processing Systems*, 33:25–37, 2020.
- [3] ALETTA, F.; MITCHELL, A.; OBERMAN, T.; KANG, J.; KHELIL, S.; BOUZIR, T. A. K.; BERKOUK, D.; XIE, H.; ZHANG, Y.; ZHANG, R.; OTHERS. **Soundscape descriptors in eighteen languages: Translation and validation through listening experiments.** *Applied acoustics*, 224:110109, 2024.
- [4] ALMUDÉVAR, A.; HERNÁNDEZ-LOBATO, J. M.; KHURANA, S.; MARXER, R.; ORTEGA, A. **Aligning multimodal representations through an information bottleneck.** *arXiv preprint arXiv:2506.04870*, 2025.
- [5] ARIK, S. Ö.; PFISTER, T. **Tabnet: Attentive interpretable tabular learning.** In: *AAAI Conference on Artificial Intelligence*, volume 35, p. 6679–6687, 2021.
- [6] BA, J. L.; KIROS, J. R.; HINTON, G. E. **Layer normalization.** *arXiv preprint arXiv:1607.06450*, 2016.
- [7] BAADE, A.; PENG, P.; HARWATH, D. **MAE-AST: Masked Autoencoding Audio Spectrogram Transformer.** In: *Interspeech 2022*, p. 2438–2442, 2022.
- [8] BADSHAH, A.; DAUD, A.; ALHARBAY, R.; BANJAR, A.; BUKHARI, A.; ALSHEMAIMRI, B. **Big data applications: Overview, challenges and future.** *Artificial Intelligence Review*, 57(11):290, 2024.
- [9] BAHDANAU, D.; CHO, K.; BENGIO, Y. **Neural machine translation by jointly learning to align and translate.** In: Bengio, Y.; LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

- [10] BAI, J.; LIU, H.; WANG, M.; SHI, D.; WANG, W.; PLUMBLEY, M. D.; GAN, W.-S.; CHEN, J. **Audiosetcaps: An enriched audio-caption dataset using automated generation pipeline with large audio and language models.** *arXiv preprint arXiv:2411.18953*, 2024.
- [11] BAI, J.; LIU, H.; WANG, M.; SHI, D.; WANG, W.; PLUMBLEY, M. D.; GAN, W.-S.; CHEN, J. **Audiosetcaps: An enriched audio-caption dataset using automated generation pipeline with large audio and language models.** *IEEE Transactions on Audio, Speech and Language Processing*, 2025.
- [12] BALTRUŠAITIS, T.; AHUJA, C.; MORENCY, L.-P. **Multimodal Nachine learning: A Survey and Taxonomy.** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2018.
- [13] BAYNE, T.; BRAINARD, D.; BYRNE, R. W.; CHITTKA, L.; CLAYTON, N.; HEYES, C.; MATHER, J.; ÖLVEČKY, B.; SHADLEN, M.; SUDDENDORF, T.; WEBB, B. **What is cognition?** *Current Biology*, 29(13):R608–R615, 2019.
- [14] BENGIO, Y.; COURVILLE, A.; VINCENT, P. **Representation learning: A review and new perspectives.** *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- [15] BISHOP, C. M. **Pattern Recognition and Machine Learning.** Springer, New York, 2006.
- [16] BOLYA, D.; HUANG, P.-Y.; SUN, P.; CHO, J. H.; MADOTTO, A.; WEI, C.; MA, T.; ZHI, J.; RAJASEGARAN, J.; BANGALATH, H.; OTHERS. **Perception encoder: The best visual embeddings are not at the output of the network.** *Advances in Neural Information Processing Systems*, 38:60884–60937, 2026.
- [17] BROKOWSKI, T.; SALLINEN, A.; HARTLEY, M.-A. **Ssca: Siglip-2 sonar concept alignment.** In: *Second Workshop on Visual Concepts*, 2025.
- [18] BROOKS, R. A. **Intelligence without representation.** *Artificial Intelligence*, 47(1–3):139–159, 1991.
- [19] CHEN, H.; XIE, W.; AFOURAS, T.; NAGRANI, A.; VEDALDI, A.; ZISSERMAN, A. **Localizing visual sounds the hard way.** In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, p. 16867–16876, 2021.
- [20] CHEN, H.; XIE, W.; VEDALDI, A.; ZISSERMAN, A. **VGGSound: A Large-Scale Audio-Visual Dataset.** In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 721–725, 2020.

- [21] CHEN, K.; DU, X.; ZHU, B.; MA, Z.; BERG-KIRKPATRICK, T.; DUBNOV, S. **HTS-AT: A Hierarchical Token-Semantic Audio Transformer for Sound Classification and Detection**. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 646–650. IEEE, 2022.
- [22] CHEN, S.; WU, Y.; WANG, C.; LIU, S.; TOMPKINS, D.; CHEN, Z.; CHE, W.; YU, X.; WEI, F. **BEATs: Audio Pre-Training with Acoustic Tokenizers**. In: *40th International Conference on Machine Learning*, p. 5178–5193, 2023.
- [23] CHEN, S.; LI, H.; WANG, Q.; ZHAO, Z.; SUN, M.; ZHU, X.; LIU, J. **VAST: A Vision-Audio-Subtitle-Text Omni-Modality Foundation Model and Dataset**. *Advances in Neural Information Processing Systems*, 36:72842–72866, 2023.
- [24] CHEN, T.; KORNBLITH, S.; NOROUZI, M.; HINTON, G. **A simple framework for contrastive learning of visual representations**. In: *International conference on machine learning*, p. 1597–1607. PmLR, 2020.
- [25] CHO, K.; VAN MERRIËNBOER, B.; BAHDANAU, D.; BENGIO, Y. **On the properties of neural machine translation: Encoder–decoder approaches**. In: *Proceedings of SSST-8, eighth workshop on syntax, semantics and structure in statistical translation*, p. 103–111, 2014.
- [26] CHOMSKY, N. **Aspects of the Theory of Syntax**. MIT Press, Cambridge, MA, 1965.
- [27] CHU, Y.; XU, J.; ZHOU, X.; YANG, Q.; ZHANG, S.; YAN, Z.; ZHOU, C.; ZHOU, J. **Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models**. *arXiv preprint arXiv:2311.07919*, 2023.
- [28] CHUNG, J.; GULCEHRE, C.; CHO, K.; BENGIO, Y. **Empirical evaluation of gated recurrent neural networks on sequence modeling**. *arXiv preprint arXiv:1412.3555*, 2014.
- [29] CHUNG, J.; GULCEHRE, C.; CHO, K.; BENGIO, Y. **Gated feedback recurrent neural networks**. In: *International conference on machine learning*, p. 2067–2075. PMLR, 2015.
- [30] COHN, N.; SCHILPEROORD, J. **A Multimodal Language Faculty: A Cognitive Framework for Human Communication**. Bloomsbury Academic, 1 edition, 2024.
- [31] CONNEAU, A.; KHANDLWAL, K.; GOYAL, N.; CHAUDHARY, V.; WENZKE, G.; GUZMÁN, F.; GRAVE, E.; OTT, M.; ZETTMLOYER, L.; STOYANOV, V. **Unsupervised Cross-lingual Representation Learning at Scale**. In: Jurafsky, D.; Chai, J.;

- Schluter, N.; Tetreault, J., editors, *58th Annual Meeting of the Association for Computational Linguistics*, p. 8440–8451, Online, July 2020. Association for Computational Linguistics.
- [32] COOLEY, J. W.; TUKEY, J. W. **An algorithm for the machine calculation of complex fourier series.** *Mathematics of computation*, 19(90):297–301, 1965.
- [33] COSTA-JUSSÀ, M. R.; CROSS, J.; ÇELEBI, O.; ELBAYAD, M.; HEAFIELD, K.; HEFFERNAN, K.; KALBASSI, E.; LAM, J.; LIGHT, D.; MAILLARD, J.; OTHERS. **No language left behind: Scaling human-centered machine translation.** *arXiv preprint arXiv:2207.04672*, 2022.
- [34] CYBENKO, G. **Approximation by superpositions of a sigmoidal function.** *Mathematics of control, signals and systems*, 2(4):303–314, 1989.
- [35] DAI, A. M.; LE, Q. V. **Semi-supervised sequence learning.** In: Cortes, C.; Lawrence, N.; Lee, D.; Sugiyama, M.; Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.
- [36] DEHAENE, S. **How We Learn: Why Brains Learn Better Than Any Machine... for Now.** Viking, New York, 2020.
- [37] DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. **BERT: Pre-training of deep bidirectional transformers for language understanding.** In: Burstein, J.; Doran, C.; Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [38] DIMENT, A.; MESAROS, A.; HEITTOLA, T.; VIRTANEN, T. **TUT Rare Sound Events, Development Dataset**, Mar. 2017.
- [39] DINKEL, H.; WANG, Y.; YAN, Z.; ZHANG, J.; WANG, Y. **Ced: Consistent ensemble distillation for audio tagging.** In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 291–295. IEEE, 2024.
- [40] DINKEL, H.; YAN, Z.; WANG, T.; WANG, Y.; SUN, X.; NIU, Y.; LIU, J.; LI, G.; ZHANG, J.; LUAN, J. **Glap: General contrastive audio-text pretraining across domains and languages.** In: *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 14737–14741. IEEE, 2026.

- [41] DINKEL, H.; YAN, Z.; WANG, Y.; ZHANG, J.; WANG, Y.; WANG, B. **Scaling up masked audio encoder learning for general audio classification.** *arXiv preprint arXiv:2406.06992*, 2024.
- [42] DOSOVITSKIY, A.; BEYER, L.; KOLESNIKOV, A.; WEISSENBORN, D.; ZHAI, X.; UNTERTHINER, T.; DEGHANI, M.; MINDERER, M.; HEIGOLD, G.; GELLY, S.; USZKOREIT, J.; HOULSBY, N. **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.** In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.
- [43] DOSOVITSKIY, A.; BEYER, L.; KOLESNIKOV, A.; WEISSENBORN, D.; ZHAI, X.; UNTERTHINER, T.; DEGHANI, M.; MINDERER, M.; HEIGOLD, G.; GELLY, S.; USZKOREIT, J.; HOULSBY, N. **An image is worth 16x16 words: Transformers for image recognition at scale.** In: *International Conference on Learning Representations*, 2021.
- [44] DOSOVITSKIY, A.; BEYER, L.; KOLESNIKOV, A.; WEISSENBORN, D.; ZHAI, X.; UNTERTHINER, T.; DEGHANI, M.; MINDERER, M.; HEIGOLD, G.; GELLY, S.; OTHERS. **An image is worth 16x16 words: Transformers for image recognition at scale.** *arXiv preprint arXiv:2010.11929*, 2020.
- [45] DROSSOS, K.; LIPPING, S.; VIRTANEN, T. **Clotho: An Audio Captioning Dataset.** In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 736–740. IEEE, 2020.
- [46] DUFUMIER, B.; NAVARRO, J. C.; TUIA, D.; THIRAN, J.-P. **What to align in multi-modal contrastive learning?** In: *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, May 2025. Poster, published 22 Jan 2025; last revised 15 Mar 2025.
- [47] DUQUENNE, P.-A.; SCHWENK, H.; SAGOT, B. **Sonar: sentence-level multimodal and language-agnostic representations.** *arXiv preprint arXiv:2308.11466*, 2023.
- [48] ELIZALDE, B.; DESHMUKH, S.; AL ISMAIL, M.; WANG, H. **Clap learning audio concepts from natural language supervision.** In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1–5. IEEE, 2023.
- [49] ELIZALDE, B.; DESHMUKH, S.; WANG, H. **Natural Language Supervision for General-Purpose Audio Representations**, 2023.

- [50] ELIZALDE, B.; DESHMUKH, S.; WANG, H. **Natural language supervision for general-purpose audio representations.** In: *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 336–340. IEEE, 2024.
- [51] ELNOKRASHY, M.; ALKHAMISSI, B.; DIAB, M. **Depth-wise attention (dwatt): A layer fusion method for data-efficient classification.** *arXiv preprint arXiv:2209.15168*, 2022.
- [52] ERNST, M. O.; BANKS, M. S. **Humans integrate visual and haptic information in a statistically optimal fashion.** *Nature*, 415(6870):429–433, Jan 2002.
- [53] ESLAMI, S.; DE MELO, G. **Mitigate the gap: Improving cross-modal alignment in clip.** In: *The Thirteenth International Conference on Learning Representations*, 2025.
- [54] EVERETT, C. **A myriad of tongues: How languages reveal differences in how we think.** In: *A Myriad of Tongues*. Harvard University Press, 2023.
- [55] FANG, W.; ZHANG, T.; CHAN, A. **To align or not to align: Strategic multimodal representation alignment for optimal performance.** In: *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, p. 21056–21064, 2026.
- [56] FENG, F.; YANG, Y.; CER, D.; ARIVAZHAGAN, N.; WANG, W. **Language-agnostic bert sentence embedding.** *arXiv preprint arXiv:2007.01852*, 2020.
- [57] FONSECA, E.; FAVORY, X.; PONS, J.; FONT, F.; SERRA, X. **Fsd50k: an open dataset of human-labeled sound events.** *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:829–852, 2021.
- [58] FONSECA, E.; PONS, J.; FAVORY, X.; FONT, F.; BOGDANOV, D.; FERRARO, A.; ORAMAS, S.; PORTER, A.; SERRA, X. **Freesound datasets: A platform for the creation of open audio datasets.** In: *ISMIR*, p. 486–493, 2017.
- [59] FONSECA, N. **Introdução à engenharia de som.** FCA - Editora de Informática, Lisboa, 2007.
- [60] FORET, P.; KLEINER, A.; MOBAHI, H.; NEYSHABUR, B. **Sharpness-aware minimization for efficiently improving generalization.** *arXiv preprint arXiv:2010.01412*, 2020.
- [61] GEIGLE, G.; TIMOFTE, R.; GLAVAŠ, G. **Babel-imagenet: Massively multilingual evaluation of vision-and-language representations**, 2024.

- [62] GEMMEKE, J. F.; ELLIS, D. P.; FREEDMAN, D.; JANSEN, A.; LAWRENCE, W.; MOORE, R. C.; PLAKAL, M.; RITTER, M. **Audio Set: An Ontology and Human-Labeled Dataset for Audio Events**. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 776–780. IEEE, 2017.
- [63] GHOSH, S.; KUMAR, S.; SETH, A.; EVURU, C. K. R.; TYAGI, U.; SAKSHI, S.; NIETO, O.; DURAISWAMI, R.; MANOCHA, D. **Gama: A large audio-language model with advanced audio understanding and complex reasoning abilities**. In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, p. 6288–6313, 2024.
- [64] GHOSH, S.; SETH, A.; KUMAR, S.; TYAGI, U.; EVURU, C. K.; SAKSHI, S.; NIETO, O.; DURAISWAMI, R.; MANOCHA, D.; OTHERS. **Compa: Addressing the gap in compositional reasoning in audio-language models**. In: *International Conference on Learning Representations*, volume 2024, p. 15486–15511, 2024.
- [65] GIRDHAR, R.; EL-NOUBY, A.; LIU, Z.; SINGH, M.; ALWALA, K. V.; JOULIN, A. R.; MISRA, I. **ImageBind: One Embedding Space to Bind them All**. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 15180–15190, 2023.
- [66] GOECKE, R. **Current Trends in Joint Audio-Video Signal Processing: A Review**. In: *Eighth International Symposium on Signal Processing and its Applications*, p. 70–73. IEEE, 2005.
- [67] GOLDSMITH, J. A.; LAKS, B. **Battle in the Mind Fields**. University of Chicago Press, Chicago and London, 2019.
- [68] GONG, Y.; CHUNG, Y.-A.; GLASS, J. **AST: Audio Spectrogram Transformer**. In: *Proc. Interspeech 2021*, p. 571–575, 2021.
- [69] GONG, Y.; CHUNG, Y.-A.; GLASS, J. **Ast: Audio spectrogram transformer**. *arXiv preprint arXiv:2104.01778*, 2021.
- [70] GONG, Y.; LAI, C.-I.; CHUNG, Y.-A.; GLASS, J. **Ssast: Self-supervised audio spectrogram transformer**. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, p. 10699–10709, 2022.
- [71] GOODFELLOW, I. **Deep learning-ian goodfellow, yoshua bengio, aaron courville-google books**, 2016.
- [72] GRASSUCCI, E.; CICHETTI, G.; FRASCA, E.; UNCINI, A.; COMMINELO, D. **Closing the modality gap aligns group-wise semantics**. In: *International Conference on Learning Representations*, 2026.

- [73] GRAVES, A. **Generating sequences with recurrent neural networks.** *arXiv preprint arXiv:1308.0850*, 2013.
- [74] GRÖGER, F.; WEN, S.; BRBIĆ, M. **Revisiting the platonic representation hypothesis: An aristotelian view.** *arXiv preprint arXiv:2602.14486*, 2026.
- [75] GUO, W.; WANG, J.; WANG, S. **Deep multimodal representation learning: A survey.** *Ieee Access*, 7:63373–63394, 2019.
- [76] GUZHOV, A.; RAUE, F.; HEES, J.; DENGEL, A. **Audioclip: Extending clip to image, text and audio.** In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 976–980. IEEE, 2022.
- [77] HAUSER, M.; CHOMSKY, N.; FITCH, W. **The faculty of language: What is it, who has it, and how did it evolve?**, p. 14–42. Cambridge University Press, Jan. 2010. Publisher Copyright: © Cambridge University Press 2010.
- [78] HAYKIN, S. S. **Neural networks and learning machines.** Pearson Education, Upper Saddle River, NJ, third edition, 2009.
- [79] HE, K.; CHEN, X.; XIE, S.; LI, Y.; DOLLÁR, P.; GIRSHICK, R. **Masked autoencoders are scalable vision learners.** In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, p. 16000–16009, 2022.
- [80] HE, K.; FAN, H.; WU, Y.; XIE, S.; GIRSHICK, R. **Momentum contrast for unsupervised visual representation learning.** In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, p. 9729–9738, 2020.
- [81] HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep residual learning for image recognition.** In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 770–778, 2016.
- [82] HEBB, D. **The Organization of Behavior: A Neuropsychological Theory.** A Wiley book in clinical psychology. Wiley, 1949.
- [83] HERSHEY, S.; CHAUDHURI, S.; ELLIS, D. P.; GEMMEKE, J. F.; JANSEN, A.; MOORE, R. C.; PLAKAL, M.; PLATT, D.; SAUROUS, R. A.; SEYBOLD, B.; OTHERS. **Cnn architectures for large-scale audio classification.** In: *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, p. 131–135. IEEE, 2017.
- [84] HESSEL, J.; HOLTZMAN, A.; FORBES, M.; LE BRAS, R.; CHOI, Y. **ClipScore: A reference-free evaluation metric for image captioning.** In: *Proceedings of the*

- 2021 conference on empirical methods in natural language processing*, p. 7514–7528, 2021.
- [85] HOCHREITER, S.; SCHMIDHUBER, J. **Long short-term memory**. *Neural computation*, 9(8):1735–1780, 1997.
- [86] HOCKETT, C. F. **Logical considerations in the study of animal communication**. In: Lanyon, W. E.; Tavolga, W. N., editors, *Animal Sounds and Communication*, número 7 em AIBS Symposium Series, p. 392–430. American Institute of Biological Sciences, Washington, DC, 1960.
- [87] HORNIK, K.; STINCHCOMBE, M.; WHITE, H. **Multilayer feedforward networks are universal approximators**. *Neural networks*, 2(5):359–366, 1989.
- [88] HU, E. J.; YELONG SHEN.; WALLIS, P.; ALLEN-ZHU, Z.; LI, Y.; WANG, S.; WANG, L.; CHEN, W. **LoRA: Low-Rank Adaptation of Large Language Models**. In: *International Conference on Learning Representations*, 2022.
- [89] HUANG, P.-Y.; XU, H.; LI, J.; BAEVSKI, A.; AULI, M.; GALUBA, W.; METZE, F.; FEICHTENHOFER, C. **Masked Autoencoders that Listen**. *Advances in Neural Information Processing Systems*, 35:28708–28720, 2022.
- [90] HUH, M.; CHEUNG, B.; WANG, T.; ISOLA, P. **The platonic representation hypothesis**. *arXiv preprint arXiv:2405.07987*, 2024.
- [91] ILHARCO, G.; WORTSMAN, M.; WIGHTMAN, R.; GORDON, C.; CARLINI, N.; TAORI, R.; DAVE, A.; SHANKAR, V.; NAMKOONG, H.; MILLER, J.; HAJISHIRZI, H.; FARHADI, A.; SCHMIDT, L. **OpenCLIP**, July 2021.
- [92] JAISWAL, A.; BABU, A. R.; ZADEH, M. Z.; BANERJEE, D.; MAKEDON, F. **A survey on contrastive self-supervised learning**. *Technologies*, 9(1):2, 2020.
- [93] JIA, C.; YANG, Y.; XIA, Y.; CHEN, Y.-T.; PAREKH, Z.; PHAM, H.; LE, Q.; SUNG, Y.-H.; LI, Z.; DUERIG, T. **Scaling up visual and vision-language representation learning with noisy text supervision**. In: *International conference on machine learning*, p. 4904–4916. PMLR, 2021.
- [94] JOSHI, P.; SANTY, S.; BUDHIRAJA, A.; BALI, K.; CHOUDHURY, M. **The State and Fate of Linguistic Diversity and Inclusion in the NLP World**. In: Jurafsky, D.; Chai, J.; Schluter, N.; Tetreault, J., editors, *58th Annual Meeting of the Association for Computational Linguistics*, p. 6282–6293, Online, July 2020. Association for Computational Linguistics.

- [95] JOZEFOWICZ, R.; ZAREMBA, W.; SUTSKEVER, I. **An empirical exploration of recurrent network architectures.** In: *International conference on machine learning*, p. 2342–2350. PMLR, 2015.
- [96] JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models.** Online manuscript, 3rd edition, 2026. Online manuscript released January 6, 2026.
- [97] KBAR, G.; BHATIA, A.; ABIDI, M. H.; AND, I. A. **Assistive technologies for hearing, and speaking impaired people: a survey.** *Disability and Rehabilitation: Assistive Technology*, 12(1):3–20, 2017. PMID: 26882961.
- [98] KIM, C. D.; KIM, B.; LEE, H.; KIM, G. **AudioCaps: Generating Captions for Audios in The Wild.** In: Burstein, J.; Doran, C.; Solorio, T., editors, *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 119–132, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [99] KIM, C. D.; KIM, B.; LEE, H.; KIM, G. **AudioCaps: Generating captions for Audios in the Wild.** In: *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, p. 119–132, 2019.
- [100] KINGMA, D. P.; BA, J. **Adam: A method for stochastic optimization.** *arXiv preprint arXiv:1412.6980*, 2014.
- [101] KOEPKE, A.; ZVEREV, D.; GINOSAR, S.; EFROS, A. A. **Back into plato’s cave: examining cross-modal representational convergence at scale.** *arXiv preprint arXiv:2604.18572*, 2026.
- [102] KOISHIGARINA, D.; USELIS, A.; OH, S. J. **Clip behaves like a bag-of-words model cross-modally but not uni-modally.** *arXiv preprint arXiv:2502.03566*, 2025.
- [103] KONG, Q.; CAO, Y.; IQBAL, T.; WANG, Y.; WANG, W.; PLUMBLEY, M. D. **PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition.** *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.
- [104] KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. **ImageNet Classification with Deep Convolutional Neural Networks.** *Advances in Neural Information Processing Systems*, 25, 2012.

- [105] LAN, Z.; CHEN, M.; GOODMAN, S.; GIMPEL, K.; SHARMA, P.; SORICUT, R. **Al-bert: A lite bert for self-supervised learning of language representations**. In: *International Conference on Learning Representations*, 2020.
- [106] LE-KHAC, P. H.; HEALY, G.; SMEATON, A. F. **Contrastive representation learning: A framework and review**. *Ieee Access*, 8:193907–193934, 2020.
- [107] LECUN, Y. **Generalization and network design strategies**. In: Pfeifer, R.; Schreter, Z.; Fogelman, F.; Steels, L., editors, *Connectionism in Perspective*, Zurich, Switzerland, 1989. Elsevier. an extended version was published as a technical report of the University of Toronto.
- [108] LECUN, Y.; BOSER, B.; DENKER, J. S.; HENDERSON, D.; HOWARD, R. E.; HUBBARD, W.; JACKEL, L. D. **Backpropagation applied to handwritten zip code recognition**. *Neural computation*, 1(4):541–551, 1989.
- [109] LEVELT, W. J. M. **A History of Psycholinguistics: The Pre-Chomskyan Era**. Oxford University Press, Oxford, 2013.
- [110] LI, F. F.; COX, T. J. **Digital Signal Processing in Audio and Acoustical Engineering**. CRC Press, 1st edition, 2019.
- [111] LI, J.; LI, D.; SAVARESE, S.; HOI, S. **Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models**. In: *International conference on machine learning*, p. 19730–19742. PMLR, 2023.
- [112] LI, Y.; GUO, Z.; WANG, X.; LIU, H. **Advancing multi-grained alignment for contrastive language-audio pre-training**. In: *Proceedings of the 32nd ACM International Conference on Multimedia*, p. 7356–7365, 2024.
- [113] LIANG, V. W.; ZHANG, Y.; KWON, Y.; YEUNG, S.; ZOU, J. Y. **Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning**. *Advances in Neural Information Processing Systems*, 35:17612–17625, 2022.
- [114] LIU, F.; REN, X.; ZHAO, G.; YOU, C.; MA, X.; WU, X.; SUN, X. **Rethinking and improving natural language generation with layer-wise multi-view decoding**. *arXiv preprint arXiv:2005.08081*, 2020.
- [115] LIU, J.; CHEN, S.; HE, X.; GUO, L.; ZHU, X.; WANG, W.; TANG, J. **VALOR: Vision-Audio-Language Omni-Perception Pretraining Model and Dataset**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–18, 2024.

- [116] LIU, S.; ZENG, Z.; REN, T.; LI, F.; ZHANG, H.; YANG, J.; JIANG, Q.; LI, C.; YANG, J.; SU, H.; OTHERS. **Grounding dino: Marrying dino with grounded pre-training for open-set object detection**. In: *European Conference on Computer Vision*, p. 38–55. Springer, 2024.
- [117] LIU, Y.; OTT, M.; GOYAL, N.; DU, J.; JOSHI, M.; CHEN, D.; LEVY, O.; LEWIS, M.; ZETTLEMOYER, L.; STOYANOV, V. **Roberta: A robustly optimized bert pretraining approach**. *arXiv preprint arXiv:1907.11692*, 2019.
- [118] LIU, Z.; LIN, Y.; CAO, Y.; HU, H.; WEI, Y.; ZHANG, Z.; LIN, S.; GUO, B. **Swin transformer: Hierarchical vision transformer using shifted windows**. In: *Proceedings of the IEEE/CVF international conference on computer vision*, p. 10012–10022, 2021.
- [119] LUGER, G. F. **Artificial intelligence: Structure and Strategies for Complex Problem Solving**. Pearson, 6 edition, Feb. 2008.
- [120] LUONG, M.-T.; PHAM, H.; MANNING, C. D. **Effective approaches to attention-based neural machine translation**. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, p. 1412–1421, 2015.
- [121] LUPAȘCU, M.; ROGOZ, A.-C.; STUPARIU, M. S.; IONESCU, R. T. **Large multimodal models for low-resource languages: a survey**. *Information Fusion*, p. 104189, 2026.
- [122] MANZOOR, M. A.; ALBARRI, S.; XIAN, Z.; MENG, Z.; NAKOV, P.; LIANG, S. **Multimodality representation learning: A survey on evolution, pretraining and its applications**. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20(3):1–34, 2023.
- [123] MATAMALA, A.; BOIX, C. O. **Accesibility and multilingualism: an exploratory study on the machine translation of audio descriptions**. *TRANS: revista de traductología*, (20):11–24, 2016.
- [124] MCCULLOCH, W. S.; PITTS, W. **A logical calculus of the ideas immanent in nervous activity**. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [125] MCGURK, H.; MACDONALD, J. **Hearing lips and seeing voices**. *Nature*, 264(5588):746–748, Dec 1976.
- [126] MEI, X.; LE LAN, G.; LIU, H.; NI, Z.; NAGARAJA, V.; LIU, Y.; SHI, Y.; CHANDRA, V. **Slap: Scalable language-audio pretraining with variable-duration audio and**

- multi-objective training.** In: *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 14517–14521. IEEE, 2026.
- [127] MEI, X.; MENG, C.; LIU, H.; KONG, Q.; KO, T.; ZHAO, C.; PLUMBLEY, M. D.; ZOU, Y.; WANG, W. **Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research.** *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3339–3354, 2024.
- [128] MEI, X.; MENG, C.; LIU, H.; KONG, Q.; KO, T.; ZHAO, C.; PLUMBLEY, M. D.; ZOU, Y.; WANG, W. **WavCaps: A ChatGPT-Assisted Weakly-Labelled Audio Captioning Dataset for Audio-Language Multimodal Research.** *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, p. 1–15, 2024.
- [129] MINSKY, M.; PAPERT, S. **Perceptrons: An Introduction to Computational Geometry.** MIT Press, Cambridge, MA, expanded edition, 1988.
- [130] MISTRETTA, M.; BALDRATI, A.; AGNOLUCCI, L.; BERTINI, M.; BAGDANOV, A. **Cross the gap: Exposing the intra-modal misalignment in clip via modality inversion.** In: *International Conference on Learning Representations*, volume 2025, p. 77750–77771, 2025.
- [131] MITCHELL, T. M. **Machine learning**, volume 1. McGraw-hill New York, 1997.
- [132] MORATO, I. M.; MESAROS, A. **MACS - Multi-Annotator Captioned Soundscapes**, July 2021.
- [133] MORATO, I. M.; MESAROS, A. **Macs-multi-annotator captioned soundscapes.** *MACS-Multi-Annotator Captioned Soundscapes*, 2021.
- [134] MOREIRA, D. A.; FERREIRA, A. I.; SILVA, J.; SANTOS, G. O. D.; PEREIRA, L.; GONDIM, J. M.; BONIL, G.; MAIA, H.; DA SILVA, N.; HASHIGUTI, S. T.; OTHERS. **Fairpivara: Reducing and assessing biases in clip-based multimodal models.** *arXiv preprint arXiv:2409.19474*, 2024.
- [135] MUENNIGHOFF, N.; TAZI, N.; MAGNE, L.; REIMERS, N. **Mteb: Massive text embedding benchmark.** In: *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, p. 2014–2037, 2023.
- [136] MURPHY, K. P. **Probabilistic Machine Learning: An introduction.** MIT Press, 2022.
- [137] NAGRANI, A.; YANG, S.; ARNAB, A.; JANSEN, A.; SCHMID, C.; SUN, C. **Attention Bottlenecks for Multimodal Fusion.** *Advances in Neural Information Processing Systems*, 34:14200–14213, 2021.

- [138] NIIZUMI, D.; TAKEUCHI, D.; OHISHI, Y.; HARADA, N.; YASUDA, M.; TSUBAKI, S.; IMOTO, K. **M2d-clap: masked modeling duo meets clap for learning general-purpose audio-language representation.** *arXiv preprint arXiv:2406.02032*, 2024.
- [139] NYQUIST, H. **Certain topics in telegraph transmission theory.** *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.
- [140] OORD, A. V. D.; LI, Y.; VINYALS, O. **Representation learning with contrastive predictive coding.** *arXiv preprint arXiv:1807.03748*, 2018.
- [141] PARCALABESCU, L.; TROST, N.; FRANK, A. **What is multimodality?** In: Donatelli, L.; Krishnaswamy, N.; Lai, K.; Pustejovsky, J., editors, *Proceedings of the 1st Workshop on Multimodal Semantic Representations (MMSR)*, p. 1–10, Groningen, Netherlands (Online), June 2021. Association for Computational Linguistics.
- [142] PARK, D. S.; CHAN, W.; ZHANG, Y.; CHIU, C.-C.; ZOPH, B.; CUBUK, E. D.; LE, Q. V. **SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition.** *Interspeech 2019*, Sept. 2019.
- [143] PARK, D. S.; CHAN, W.; ZHANG, Y.; CHIU, C.-C.; ZOPH, B.; CUBUK, E. D.; LE, Q. V. **SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition.** *Interspeech 2019*, p. 2613, 2019.
- [144] PASAD, A. **What do speech foundation models learn? analysis and applications.** *arXiv preprint arXiv:2508.12255*, 2025.
- [145] PASAD, A.; CHOU, J.-C.; LIVESCU, K. **Layer-wise analysis of a self-supervised speech representation model.** In: *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, p. 914–921. IEEE, 2021.
- [146] PASAD, A.; SHI, B.; LIVESCU, K. **Comparative layer-wise analysis of self-supervised speech models.** In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1–5. IEEE, 2023.
- [147] PATHAK, D.; KRAHENBUHL, P.; DONAHUE, J.; DARRELL, T.; EFROS, A. A. **Context encoders: Feature learning by inpainting.** In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, p. 2536–2544, 2016.
- [148] PHAN, H.; KOCH, P.; KATZBERG, F.; MAASS, M.; MAZUR, R.; MERTINS, A. **Audio scene classification with deep recurrent neural networks.** *Interspeech 2017*, p. 3043–3047, 2017.

- [149] PICZAK, K. J. **ESC: Dataset for Environmental Sound Classification**. In: *Proceedings of the 23rd Annual ACM Conference on Multimedia*, p. 1015–1018. ACM Press.
- [150] PICZAK, K. J. **Environmental sound classification with convolutional neural networks**. In: *2015 IEEE 25th international workshop on machine learning for signal processing (MLSP)*, p. 1–6. IEEE, 2015.
- [151] PICZAK, K. J. **ESC: Dataset for Environmental Sound Classification**. In: *23rd ACM international conference on Multimedia*, p. 1015–1018, 2015.
- [152] PRINCE, S. J. **Understanding Deep Learning**. The MIT Press, 2023.
- [153] PROAKIS, J. G.; MANOLAKIS, D. G. **Digital Signal Processing: Principles, Algorithms, and Applications**. Pearson Prentice Hall, 4th edition, 2006.
- [154] RADFORD, A.; KIM, J. W.; HALLACY, C.; RAMESH, A.; GOH, G.; AGARWAL, S.; SASTRY, G.; ASKELL, A.; MISHKIN, P.; CLARK, J.; KRUEGER, G.; SUTSKEVER, I. **Learning Transferable Visual Models From Natural Language Supervision**. *CoRR*, abs/2103.00020, 2021.
- [155] RADFORD, A.; NARASIMHAN, K.; SALIMANS, T.; SUTSKEVER, I.; OTHERS. **Improving language understanding by generative pre-training**. Technical report, 2018.
- [156] RADFORD, A.; WU, J.; CHILD, R.; LUAN, D.; AMODEI, D.; SUTSKEVER, I.; OTHERS. **Language models are unsupervised multitask learners**. *OpenAI blog*, 1(8):9, 2019.
- [157] RAFFEL, C.; SHAZEER, N.; ROBERTS, A.; LEE, K.; NARANG, S.; MATENA, M.; ZHOU, Y.; LI, W.; LIU, P. J. **Exploring the limits of transfer learning with a unified text-to-text transformer**. *Journal of machine learning research*, 21(140):1–67, 2020.
- [158] RAHMAN, M. S.; REZA, H. **A systematic review towards big data analytics in social media**. *Big Data Mining and Analytics*, 5(3):228–244, 2022.
- [159] RIHOM, N. **Exploring the intersection of perception and cognition: The role of sensory experience in mental processes**. *Journal of Psychology and Cognition*, 9(6), 2024. Opinion Article.
- [160] ROSENBLATT, F. **The perceptron: a probabilistic model for information storage and organization in the brain**. *Psychological review*, 65(6):386, 1958.

- [161] ROSENBLATT, F.; OTHERS. **Principles of neurodynamics: Perceptrons and the theory of brain mechanisms**, volume 55. Spartan books Washington, DC, 1962.
- [162] RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. **Learning representations by back-propagating errors**. *nature*, 323(6088):533–536, 1986.
- [163] RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Always learning. Pearson, 2016.
- [164] SALAMON, J.; JACOBY, C.; BELLO, J. P. **A Dataset and Taxonomy for Urban Sound Research**. In: *22nd ACM International Conference on Multimedia*, p. 1041–1044, 2014.
- [165] SANTOS, G. O. D.; MOREIRA, D. A. B.; FERREIRA, A. I.; SILVA, J.; PEREIRA, L.; BUENO, P.; SOUSA, T.; MAIA, H.; DA SILVA, N.; COLOMBINI, E.; PEDRINI, H.; AVILA, S. **CAPIVARA: Cost-Efficient Approach for Improving Multilingual CLIP Performance on Low-Resource Languages**. In: *Workshop on Multi-lingual Representation Learning (MRL), Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 184–207, 2023.
- [166] SCHRODI, S.; HOFFMANN, D. T.; ARGUS, M.; FISCHER, V.; BROX, T. **Two effects, one trigger: On the modality gap, object bias, and information imbalance in contrastive vision-language models**. In: *International Conference on Learning Representations*, volume 2025, p. 27836–27864, 2025.
- [167] SCHUHMANN, C.; BEAUMONT, R.; VENCU, R.; GORDON, C.; WIGHTMAN, R.; CHERTI, M.; COOMBES, T.; KATTA, A.; MULLIS, C.; WORTSMAN, M.; OTHERS. **Laion-5b: An open large-scale dataset for training next generation image-text models**. *Advances in neural information processing systems*, 35:25278–25294, 2022.
- [168] SEJNOWSKI, T. J. **The Deep Learning Revolution**. The MIT Press. MIT Press, London, England, Sept. 2018.
- [169] SEKI, K.; OKAMOTO, Y.; YAMAOKA, K.; SAITO, Y.; TAKAMICHI, S.; SARUWATARI, H. **Spatial-clap: Learning spatially-aware audio–text embeddings for multi-source conditions**. In: *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 14742–14746. IEEE, 2026.
- [170] SHAH, J.; SINGLA, Y. K.; CHEN, C.; SHAH, R. R. **What all do audio transformer models hear? probing acoustic representations for language delivery and its structure**. *arXiv preprint arXiv:2101.00387*, 2021.

- [171] SHANNON, C. E. **Communication in the presence of noise.** *Proceedings of the IRE*, 37(1):10–21, 1949.
- [172] SHARMA, G.; UMAPATHY, K.; KRISHNAN, S. **Trends in audio signal feature extraction methods.** *Applied Acoustics*, 158:107020, 2020.
- [173] SHI, P.; WELLE, M.; BJØRKMAN, M.; KRAGIC, D. **Understanding the modality gap in clip.** *ICLR, Stockholm, Sweden*, p. 8–10, 2023.
- [174] SILVA, J.; FERREIRA, A.; MOREIRA, D.; SANTOS, G.; BONIL, G.; GONDIM, J.; PEREIRA, L.; MAIA, H.; SILVA, N.; HASHIGUTI, S.; AVILA, S.; PEDRINI, H. **Avaliação de ferramentas de Ética no levantamento de considerações Éticas de modelos de linguagem em português.** In: *Anais da 1ª Conferência Latino-Americana de Ética em Inteligência Artificial*, p. 61–64, Porto Alegre, RS, Brasil, 2024. SBC.
- [175] SKEAN, O.; AREFIN, M. R.; ZHAO, D.; PATEL, N.; NAGHIYEV, J.; LECUN, Y.; SHWARTZ-ZIV, R. **Layer by layer: Uncovering hidden representations in language models.** *arXiv preprint arXiv:2502.02013*, 2025.
- [176] SMITH, L. N. **Cyclical learning rates for training neural networks.** In: *2017 IEEE winter conference on applications of computer vision (WACV)*, p. 464–472. IEEE, 2017.
- [177] STEIN, B. E.; STANFORD, T. R. **Multisensory integration: current issues from the perspective of the single neuron.** *Nature Reviews Neuroscience*, 9(4):255–266, Apr 2008.
- [178] STEVENS, S. S.; VOLKMANN, J.; NEWMAN, E. B. **A scale for the measurement of the psychological magnitude pitch.** *The journal of the acoustical society of america*, 8(3):185–190, 1937.
- [179] STRUBELL, E.; GANESH, A.; MCCALLUM, A. **Energy and Policy Considerations for Deep Learning in NLP.** In: Korhonen, A.; Traum, D.; Màrquez, L., editors, *57th Annual Meeting of the Association for Computational Linguistics*, p. 3645–3650, Florence, Italy, July 2019. Association for Computational Linguistics.
- [180] STRUBELL, E.; GANESH, A.; MCCALLUM, A. **Energy and Policy Considerations for Modern Deep Learning Research.** In: *AAAI Conference on Artificial Intelligence*, volume 34, p. 13693–13696, 2020.
- [181] SU, J.; AHMED, M.; LU, Y.; PAN, S.; BO, W.; LIU, Y. **Roformer: Enhanced transformer with rotary position embedding.** *Neurocomputing*, 568:127063, 2024.

- [182] SU, Y.; BAI, J.; XU, Q.; XU, K.; DOU, Y. **Audio-language models for audio-centric tasks: A survey.** *arXiv preprint arXiv:2501.15177*, 2025.
- [183] SUN, L.; XU, X.; WU, M.; XIE, W. **Auto-acd: A large-scale dataset for audio-language representation learning.** In: *Proceedings of the 32nd ACM International Conference on Multimedia*, p. 5025–5034, 2024.
- [184] SZELISKI, R. **Computer Vision: Algorithms and Applications.** Texts in Computer Science. Springer Cham, 2 edition, 2022.
- [185] TJANDRASUWITA, M.; EKBOTE, C.; ZIYIN, L.; LIANG, P. P. **Understanding the emergence of multimodal representation alignment.** *arXiv preprint arXiv:2502.16282*, 2025.
- [186] TORRALBA, A.; ISOLA, P.; FREEMAN, W. **Foundations of Computer Vision.** Adaptive Computation and Machine Learning series. MIT Press, 2024.
- [187] TSCHANNEN, M.; GRITSENKO, A.; WANG, X.; NAEEM, M. F.; ALABDULMOHSIN, I.; PARTHASARATHY, N.; EVANS, T.; BEYER, L.; XIA, Y.; MUSTAFA, B.; OTHERS. **Siglip 2: Multilingual vision-language encoders with improved semantic understanding.** *Localization, and Dense Features*, 6, 2025.
- [188] VAN AKEN, B.; WINTER, B.; LÖSER, A.; GERS, F. A. **How does bert answer questions? a layer-wise analysis of transformer representations.** In: *Proceedings of the 28th ACM international conference on information and knowledge management*, p. 1823–1832, 2019.
- [189] VAN DEN OORD, A.; LI, Y.; VINYALS, O. **Representation Learning with Contrastive Predictive Coding.** *CoRR*, abs/1807.03748, 2018.
- [190] VAN DEN OORD, A.; DIELEMAN, S.; ZEN, H.; SIMONYAN, K.; VINYALS, O.; GRAVES, A.; KALCHBRENNER, N.; SENIOR, A.; KAVUKCUOGLU, K. **WaveNet: A Generative Model for Raw Audio.** In: *9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, p. 125, 2016.
- [191] VAN DER MAATEN, L.; HINTON, G. **Visualizing high-dimensional data using t-sne.** *journal of machine learning research.* *J Mach Learn Res*, 9(26):5, 2008.
- [192] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. **Attention is All You Need.** In: *31st International Conference on Neural Information Processing Systems, NIPS'17*, p. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

- [193] WANG, J.; REYNAUD, H.; HAMAMCI, I. E.; ER, S.; SHIT, S.; MENZE, B.; KAINZ, B. **Sigvlp: Sigmoid volume-language pre-training for self-supervised ct-volume adaptive representation learning**. *arXiv preprint arXiv:2602.21735*, 2026.
- [194] WANG, J.; CHEN, Y.; HAO, S.; PENG, X.; HU, L. **Deep Learning for Sensor-Based Activity Recognition: A Survey**. *Pattern Recognition Letters*, 119:3–11, 2019. Deep Learning for Pattern Recognition.
- [195] WANG, L.; YANG, N.; HUANG, X.; YANG, L.; MAJUMDER, R.; WEI, F. **Multilingual e5 text embeddings: A technical report**. *arXiv preprint arXiv:2402.05672*, 2024.
- [196] WANG, T.; ISOLA, P. **Understanding contrastive representation learning through alignment and uniformity on the hypersphere**. In: *International conference on machine learning*, p. 9929–9939. PMLR, 2020.
- [197] WANG, Z.; ZHAO, Y.; HUANG, H.; LIU, J.; YIN, A.; TANG, L.; LI, L.; WANG, Y.; ZHANG, Z.; ZHAO, Z. **Connecting Multi-Modal Contrastive Representations**. *Advances in Neural Information Processing Systems*, 36:22099–22114, 2023.
- [198] WIDROW, B.; HOFF, M. **Adaptive switching circuits**. *Western Electronic Show and Convention*, 4:96–104, 1960.
- [199] WU, S.; YU, D.; TAN, X.; SUN, M. **Clamp: Contrastive language-music pre-training for cross-modal symbolic music information retrieval**. In: Sarti, A.; Antonacci, F.; Sandler, M.; Bestagini, P.; Dixon, S.; Liang, B.; Richard, G.; Pauwels, J., editors, *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023, Milan, Italy, November 5-9, 2023*, p. 157–165, 2023.
- [200] WU, Y.; LIU, L.; BAE, J.; CHOW, K.-H.; IYENGAR, A.; PU, C.; WEI, W.; YU, L.; ZHANG, Q. **Demystifying learning rate policies for high accuracy training of deep neural networks**. In: *2019 IEEE International conference on big data (Big Data)*, p. 1971–1980. IEEE, 2019.
- [201] WU, Y.; CHEN, K.; ZHANG, T.; HUI, Y.; BERG-KIRKPATRICK, T.; DUBNOV, S. **Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation**. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1–5. IEEE, 2023.
- [202] WU, Y.; CHEN, K.; ZHANG, T.; HUI, Y.; BERG-KIRKPATRICK, T.; DUBNOV, S. **Large-scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation**. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

- [203] XU, K.; BA, J.; KIROS, R.; CHO, K.; COURVILLE, A.; SALAKHUDINOV, R.; ZEMEL, R.; BENGIO, Y. **Show, attend and tell: Neural image caption generation with visual attention**. In: *International conference on machine learning*, p. 2048–2057. PMLR, 2015.
- [204] XU, X.; ZHANG, Z.; ZHOU, Z.; ZHANG, P.; XIE, Z.; WU, M.; ZHU, K. Q. **Blat: Bootstrapping language-audio pre-training based on audioset tag-guided synthetic data**. In: *Proceedings of the 31st ACM International Conference on Multimedia*, p. 2756–2764, 2023.
- [205] XUE, L.; CONSTANT, N.; ROBERTS, A.; KALE, M.; AL-RFOU, R.; SIDDHANT, A.; BARUA, A.; RAFFEL, C. **mt5: A massively multilingual pre-trained text-to-text transformer**. In: *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies*, p. 483–498, 2021.
- [206] YAN, Z.; DINKEL, H.; WANG, Y.; LIU, J.; ZHANG, J.; WANG, Y.; WANG, B. **Bridging language gaps in audio-text retrieval**. In: *Interspeech 2024*, p. 1675–1679, 2024.
- [207] YAN, Z.; DINKEL, H.; WANG, Y.; LIU, J.; ZHANG, J.; WANG, Y.; WANG, B. **Bridging language gaps in audio-text retrieval**. *arXiv preprint arXiv:2406.07012*, 2024.
- [208] YANG, Y.; ABREGO, G. H.; YUAN, S.; GUO, M.; SHEN, Q.; CER, D.; SUNG, Y.-H.; STROPE, B.; KURZWEIL, R. **Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax**. *arXiv preprint arXiv:1902.08564*, 2019.
- [209] YANG, Z.; DAI, Z.; YANG, Y.; CARBONELL, J.; SALAKHUTDINOV, R. R.; LE, Q. V. **Xlnet: Generalized autoregressive pretraining for language understanding**. *Advances in neural information processing systems*, 32, 2019.
- [210] YANG, Z.; YANG, D.; DYER, C.; HE, X.; SMOLA, A.; HOVY, E. **Hierarchical attention networks for document classification**. In: Knight, K.; Nenkova, A.; Rambow, O., editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.
- [211] YARAS, C.; CHEN, S.; WANG, P.; QU, Q. **Explaining and mitigating the modality gap in contrastive multimodal learning**. *arXiv preprint arXiv:2412.07909*, 2024.

- [212] YEH, C.-F.; HUANG, P.-Y.; SHARMA, V.; LI, S.-W.; GOSH, G. **Flap: Fast language-audio pre-training**. In: *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, p. 1–8. IEEE, 2023.
- [213] YIN, Y.; XIE, Y.; YANG, W.; YANG, D.; RU, J.; ZHUANG, X.; LIANG, L.; ZOU, Y. **ATRI: Mitigating multilingual audio text retrieval inconsistencies by reducing data distribution errors**. In: Che, W.; Nabende, J.; Shutova, E.; Pilehvar, M. T., editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, p. 5491–5504, Vienna, Austria, July 2025. Association for Computational Linguistics.
- [214] YUAN, Y.; CHEN, Z.; LIU, X.; LIU, H.; XU, X.; JIA, D.; CHEN, Y.; PLUMBLEY, M. D.; WANG, W. **T-clap: Temporal-enhanced contrastive language-audio pretraining**. In: *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, p. 1–6. IEEE, 2024.
- [215] ZAMAN, K.; SAH, M.; DIREKOGLU, C.; UNOKI, M. **A survey of audio classification using deep learning**. *IEEE access*, 11:106620–106649, 2023.
- [216] ZHAI, X.; MUSTAFA, B.; KOLESNIKOV, A.; BEYER, L. **Sigmoid loss for language image pre-training**. In: *Proceedings of the IEEE/CVF international conference on computer vision*, p. 11975–11986, 2023.
- [217] ZHAI, X.; WANG, X.; MUSTAFA, B.; STEINER, A.; KEYSERS, D.; KOLESNIKOV, A.; BEYER, L. **LiT: Zero-Shot Transfer with Locked-Image Text Tuning**. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 18123–18133, 2022.
- [218] ZHANG, L.; RUAN, L.; HU, A.; JIN, Q. **Multimodal Pretraining from Monolingual to Multilingual**. *Machine Intelligence Research*, 20(2):220–232, 2023.
- [219] ZHANG, X.; ZHANG, Y.; LONG, D.; XIE, W.; DAI, Z.; TANG, J.; LIN, H.; YANG, B.; XIE, P.; HUANG, F.; OTHERS. **mgte: Generalized long-context text representation and reranking models for multilingual text retrieval**. *arXiv preprint arXiv:2407.19669*, 2024.
- [220] ZHANG, Z.; HAN, X.; LIU, Z.; JIANG, X.; SUN, M.; LIU, Q. **Ernie: Enhanced language representation with informative entities**. In: *Proceedings of the 57th annual meeting of the association for computational linguistics*, p. 1441–1451, 2019.
- [221] ZHANG, Z.; ZHOU, Z.; TANG, H.; LI, G.; WU, M.; ZHU, K. Q. **Enriching ontology with temporal commonsense for low-resource audio tagging**. In: *Proceedings of*

- the 30th ACM International Conference on Information & Knowledge Management*, p. 3652–3656, 2021.
- [222] ZHAO, Y.; HESSEL, J.; YU, Y.; LU, X.; ZELLERS, R.; CHOI, Y. **Connecting the dots between audio and text without parallel data through visual knowledge transfer**. In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 4492–4507, 2022.
- [223] ZHOU, M.; ZHOU, L.; WANG, S.; CHENG, Y.; LI, L.; YU, Z.; LIU, J. **UC2: Universal Cross-lingual Cross-modal Vision-and-Language Pre-training**. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 4155–4165, 2021.
- [224] ZHU, B.; LIN, B.; NING, M.; YAN, Y.; CUI, J.; HONGFA, W.; PANG, Y.; JIANG, W.; ZHANG, J.; LI, Z.; OTHERS. **LanguageBind: Extending Video-Language Pretraining to N -modality by Language-based Semantic Alignment**. In: *The Twelfth International Conference on Learning Representations*, p. 1–22, 2023.
- [225] ZHU, G.; DAREFSKY, J.; DUAN, Z. **Cacophony: An improved contrastive audio-text model**. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.
- [226] ZÖLZER, U. **Digital Audio Signal Processing**. John Wiley & Sons, 2nd edition, 2008.
- [227] ZONG, Y.; MAC AODHA, O.; HOSPEDALES, T. **Self-supervised multimodal learning: A survey**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Resources Description

This chapter describes the main resources used in the experimental pipeline, including the data augmentation strategies and the datasets adopted for training, validation, and evaluation.

A.1 Data Augmentation

Two augmentation strategies were considered in this work: Random Truncation (RT) and SpecAugment. These techniques were evaluated in a preliminary stage in order to assess their impact on retrieval and classification performance.

Random Truncation (RT) standardizes the temporal length of the audio input by either truncating or padding each clip to a fixed duration. In this work, the target duration was set to 10 seconds. For audio clips shorter than the target length, silence padding is applied to reach the required duration. For clips longer than the target duration, a random segment of the appropriate length is extracted. This procedure exposes the model to different temporal regions of the same audio signal during training, introducing temporal variability and reducing overfitting while maintaining consistent input dimensions across the dataset.

SpecAugment (SpecAug) [143] is a data augmentation method applied directly to the time-frequency representation of the audio signal. Given a log-mel spectrogram $\mathbf{X} \in \mathbb{R}^{v \times \tau}$, where v is the number of mel-frequency channels and τ is the number of time steps, the spectrogram is treated analogously to an image whose vertical axis corresponds to frequency and whose horizontal axis corresponds to time. The original SpecAugment policy is composed of three transformations: time warping, frequency masking, and time masking. Time warping slightly deforms the spectrogram along the temporal axis by selecting a random point on the time axis and moving it to the left or right by a randomly sampled displacement. This operation encourages the model to become less sensitive to small temporal distortions in the input. Frequency masking removes a contiguous block of mel-frequency channels from the spectrogram. More formally, a mask width f is sampled from a uniform distribution between 0 and a maximum frequency-mask parameter F ,

and a starting frequency index f_0 is sampled such that the interval $[f_0, f_0 + f)$ lies within the v mel channels. The selected frequency band is then masked for all time steps. This operation simulates the partial loss or corruption of spectral information and encourages the model to avoid relying excessively on narrow frequency regions. Time masking follows the same principle along the temporal axis: a mask width t is sampled from a uniform distribution between 0 and a maximum time-mask parameter T , and a starting time index t_0 is sampled such that the interval $[t_0, t_0 + t)$ lies within the τ time steps. The selected temporal segment is then masked across all frequency channels, forcing the model to infer the relevant semantics from the remaining acoustic evidence. Multiple frequency and time masks can be applied to the same spectrogram, and these masks are allowed to overlap. Since the log-mel spectrograms are normalized to have zero mean, the masked regions are set to zero, which is equivalent to replacing them with the mean value of the normalized input. In this work, SpecAugment was applied using a maximum frequency-mask width of $F = 48$ and a maximum time-mask width of $T = 96$, controlling the largest possible masked regions along the frequency and temporal axes, respectively.

A.2 Datasets

AudioCaps [99] is a human-annotated audio-caption dataset built on top of AudioSet [62]. It contains audio clips from YouTube videos and natural language captions collected through crowdsourcing. The dataset was designed to describe audio content in the wild, emphasizing auditory information rather than visual scene descriptions. In its commonly used splits, the training set contains audio clips with a single caption, while the validation and test sets contain multiple captions per clip, enabling more reliable evaluation. AudioCaps is used in this work both as a high-quality audio-text training resource and as a standard benchmark for audio-text retrieval evaluation. Its human-written captions provide semantically precise descriptions that are useful for aligning audio and language representations.

Clotho [45] is a human-annotated audio-caption dataset designed specifically for general audio captioning. The original release contains 4,981 audio samples and 24,905 captions, with each audio sample associated with five captions. The clips are between 15 and 30 seconds long and were collected from the Freesound platform. In this work, Clotho is used as a high-quality retrieval benchmark and as a complementary training resource to AudioCaps.

MACS [132] or Multi-Annotator Captioned Soundscapes, is a captioned soundscape dataset containing approximately 4,000 audio samples with multiple human annotations per clip. Each clip is limited to approximately 10 seconds, and the annotations are focused on the auditory content of the recordings. Since MACS provides multiple descrip-

tions for the same audio sample, it captures some of the natural variability in how humans describe sound events and acoustic scenes. In this work, MACS is useful as a validation resource because it provides carefully annotated captions that can help assess the quality of audio-text alignment.

WavCaps [127] is a weakly labelled audio-caption dataset containing approximately 400K audio-caption pairs. It was constructed from multiple sources, including FreeSound, BBC Sound Effects, SoundBible, and AudioSet. Since raw descriptions collected from web sources are often noisy, incomplete, or not directly suitable as captions, WavCaps applies a structured three-stage processing pipeline to filter noisy samples and transform raw descriptions into more caption-like sentences. Although WavCaps is weakly labelled because its captions are automatically refined rather than manually written from scratch, it has been shown to be effective for audio-language pretraining. In this work, WavCaps is used as a large-scale audio-text resource that complements human-annotated datasets by increasing the diversity and volume of training data.

Auto-ACD [183] is a large-scale audio-language dataset containing approximately 1.5 million audio-text pairs. The dataset was created through an automatic pipeline that uses multimodal cues, including audio, visual, and textual information, to generate rich descriptions of audio content. A key characteristic of Auto-ACD is the inclusion of detailed environmental and contextual information in the generated captions. Its captions are generally longer and more descriptive than those found in many previous audio-caption datasets, with an average length of approximately 18 words and a vocabulary of around 20K-23K words. In the context of this work, Auto-ACD is used as a large-scale training resource because its detailed captions can help the model learn richer correspondences between acoustic events and natural language descriptions.

AudioSetCaps [10] is a large-scale audio-caption dataset derived from AudioSet recordings and constructed through an automated generation pipeline. The dataset contains approximately 1.9 million audio-caption pairs and was designed to provide enriched audio-language supervision at a scale larger than previous audio-caption resources. Its pipeline integrates large audio-language models for fine-grained content extraction, large language models for caption generation, and CLAP-based refinement to filter low-quality captions and reduce hallucinated content. Unlike datasets that rely only on raw metadata or simple tag-to-caption conversion, AudioSetCaps aims to include more detailed acoustic information, such as speech characteristics, music genre, musical instruments, and environmental context. In this work, AudioSetCaps is used as a large-scale resource for audio-language training because it provides broad acoustic coverage and caption-level supervision suitable for contrastive representation learning.

ESC-50 [151] is an environmental sound classification dataset containing 2,000 labelled audio clips distributed across 50 balanced classes. Each class contains 40 clips,

and each clip has a duration of 5 seconds. The 50 classes are organized into five broader groups: animal sounds, natural soundscapes and water sounds, human non-speech sounds, domestic sounds, and exterior or urban noises. ESC-50 provides predefined 5-fold cross-validation splits, making it a widely used benchmark for environmental sound classification. In this work, ESC-50 is used to evaluate the zero-shot classification capability of the learned audio-language representation space.

UrbanSound8K [164] is an urban sound classification dataset containing 8,732 labelled audio excerpts from 10 urban sound classes. The classes include air conditioner, car horn, children playing, dog bark, drilling, engine idling, gun shot, jackhammer, siren, and street music. The clips are at most 4 seconds long and are organized into 10 predefined folds for cross-validation. UrbanSound8K is particularly useful for evaluating the recognition of sound events in real-world urban environments, where background noise and acoustic variability are common. In this work, it is used as a zero-shot classification benchmark to assess whether the learned representations generalize to short urban sound events.

VGGSound [20] is a large-scale audio-visual dataset containing approximately 200K 10-second video clips distributed across 309 sound classes. The dataset was collected from YouTube and curated to ensure audio-visual correspondence, meaning that the object or event producing the sound is expected to be visible in the video. Its classes cover a broad range of categories, including human sounds, animal vocalizations, music, environmental sounds, and mechanical noises. Although VGGSound is audio-visual by construction, it is also widely used for audio recognition and audio classification experiments. In this work, VGGSound is used to help evaluate the audio-image alignment of CACARA.

VGG-Sound Source (VGG-SS) [19] is an audio-visual sound-source localization benchmark derived from the VGG-Sound test set. While VGG-Sound provides video-level sound-category labels, VGG-SS extends it with spatial annotations indicating where the visible sound-producing objects are located in the image. The dataset contains 5,158 video clips spanning 220 sound categories, making it substantially larger and more diverse than earlier sound-source localization benchmarks such as Flickr SoundNet. Its construction followed a multi-stage annotation process. First, the authors considered the VGG-Sound test set, composed of 15K 10-second clips from 309 classes, and extracted the center frame of each video. Candidate bounding boxes were generated automatically using a Faster R-CNN object detector pretrained on OpenImages, and semantic matching between visual and audio categories was performed using word2vec. The remaining samples were manually annotated, while classes that were difficult or impossible to localize spatially, such as diffuse environmental sounds, were removed. Finally, the annotations were manually verified by watching a 5-second video segment around each annotated frame

to ensure that the annotated object actually corresponded to the audible sound source. The final dataset provides bounding-box annotations for visible sounding objects and includes cases with one or multiple sound sources in the same frame. Therefore, VGG-SS is particularly useful for evaluating whether audio-visual models can localize the object or region responsible for a sound, rather than only recognizing the sound category. In this work, VGG-SS is used as a source to measure the alignment between audio and image modalities.

Dataset	Main use	Audio samples	Captions	Avg. Audios Duration
AudioSetCaps [10]	Training	1.9M audio-caption pairs	1.9M captions	10 s
Auto-ACD [183]	Training	1.5M audio-text pairs	1.5M captions	Variable
WavCaps [127]	Training	~400K audio-caption pairs	~400K captions	Variable
AudioCaps [99]	Training / Evaluation	~46K audio clips	~57K captions	10 s
Clotho v2 [45]	Training / Evaluation	4,981 samples in the original release	24,905 captions in the original release	15–30 s
MACS [132]	Validation	~4K audio samples	Multiple captions per clip	10 s

Table A.1: Summary of the audio-text datasets used for retrieval-oriented training and evaluation.

Dataset	Task	Main use	# Samples	# Classes	Duration
ESC-50 [151]	Environmental sound classification	Evaluation	2,000	50	5 s
UrbanSound8K [164]	Urban sound classification	Evaluation	8,732	10	≤ 4 s
VGGSound [20]	Audio classification	Evaluation / Validation	~200K	309	10 s
VGG-SS [19]	Audio-visual sound-source localization	Evaluation	5,158	220	10 s

Table A.2: Summary of datasets used for validation and evaluation.

A.3 Audio Encoders

This section describes the pretrained audio encoders evaluated in the CACARA and PALMA optimization pipelines. All encoders operate over time-frequency audio representations, such as log-mel spectrograms or filterbank features, but differ in their architectural design, pretraining objective, and supervision strategy. The evaluated models include supervised audio-tagging encoders, self-supervised masked-modeling encoders, and distillation-based encoders.

HTS-AT (The Hierarchical Token-Semantic Audio Transformer [21]) is a Transformer-based audio encoder designed for efficient sound classification and sound event detection. Unlike standard audio transformers that process the spectrogram as a long sequence of fixed-length tokens, HTS-AT adopts a hierarchical structure inspired by the Swin Transformer [118]. The input mel-spectrogram is first divided into local time-frequency patches, which are embedded into token representations. These tokens are then processed through groups of transformer blocks using local window attention, reducing the computational cost associated with global self-attention. Between transformer groups, patch merging progressively reduces the temporal and frequency resolution while increasing the channel dimension, allowing the model to build higher-level audio representations in a hierarchical manner. HTS-AT also introduces a token-semantic module that maps latent tokens into class-specific activation maps, enabling weakly supervised temporal localization of sound events in addition to clip-level classification.

AudioMAE [89] extends the masked autoencoder framework [79] from computer vision to audio representation learning. The model follows an encoder-decoder architecture based on Vision Transformers [42]. First, an audio signal is converted into a spectrogram and divided into non-overlapping patches. During self-supervised pretraining, a large proportion of these patches is masked, and only the visible patches are passed through the Transformer encoder. The decoder then receives the encoded visible patches together with learnable mask tokens and attempts to reconstruct the original spectrogram. The reconstruction objective is typically defined as a mean squared error over the masked regions. A central motivation of AudioMAE is that spectrograms contain substantial local redundancy in both time and frequency, making it possible to learn useful contextual representations from a small subset of visible patches. The authors also explore local attention mechanisms in the decoder to better model the local structure of audio spectrograms. After pretraining, the decoder is discarded, and the encoder is fine-tuned or reused as a general-purpose audio representation model.

BEATs (Bidirectional Encoder representation from Audio Transformers) [22] is a self-supervised audio pretraining framework based on masked acoustic-token prediction. Unlike reconstruction-based methods such as AudioMAE, which require the model to reconstruct continuous spectrogram values, BEATs trains the model to predict discrete acoustic labels for masked regions. These labels are produced by an acoustic tokenizer. In the first training iteration, the tokenizer is initialized using random projection to provide a cold-start set of discrete targets. In later iterations, the tokenizer is improved by distilling semantic knowledge from a pretrained or fine-tuned audio model. This iterative process allows the tokenizer and the audio encoder to improve each other, in which the tokenizer produces more semantically meaningful targets, while the encoder learns higher-level audio representations from masked-token prediction. BEATs uses a

ViT-based [42] backbone and was designed to encourage the model to focus on semantic acoustic patterns rather than low-level reconstruction details, making it particularly effective for downstream audio classification tasks. We use BEATs in its third training iteration.

CED (Consistent Ensemble Distillation) [39] is a distillation-based framework for audio tagging that trains compact Transformer-based student models from strong teacher ensembles. The core idea is to combine knowledge distillation with consistent data augmentation. Instead of only storing teacher predictions for the original input samples, CED stores both the teacher logits and the random seeds used to generate the augmentations. This makes it possible to reproduce the same augmented input for the student while using the corresponding teacher prediction, enabling consistent teaching without storing large augmented datasets. The method is also label-free during student optimization, since the student is trained directly from the stored teacher logits. CED includes multiple ViT-based [42] model sizes. We use CED in its Base version in this work.

Encoder	Architecture	Pretraining method	Training strategy	Parameters
HTS-AT [21]	Transformer (Swin-based [118])	Supervised pretraining	Audio tagging	~30M
AudioMAE [89]	Transformer (ViT-based [42])	Self-supervised pretraining	Mask reconstruction	~86M
BEATs [22]	Transformer (ViT-based [42])	Self-supervised pretraining	Masked acoustic-token prediction	~90M
CED [39]	Transformer (ViT-based [42])	MAE + knowledge distillation	Consistent ensemble distillation	~86M

Table A.3: Comparison of the pretrained audio encoders evaluated in the CACARA optimization pipeline. The table reports the architecture family, pretraining method, training signal, and approximate number of parameters for each encoder. Parameter counts may vary depending on the specific checkpoint or implementation variant used.

A.4 Text Encoders

This section describes the pretrained text encoders considered in the Text Encoder Consolidation phase of PALMA. Since the proposed framework aims to align audio with natural language descriptions across multiple languages, the choice of text encoder is central to the quality and transferability of the shared latent space. The evaluated models differ in their pretraining objectives, multilingual coverage, sentence-level alignment strategies, and suitability for retrieval-oriented representation learning.

XLM-RoBERTa (XLM-R) [31] is a multilingual Transformer encoder-only model pretrained with a masked language modeling objective [37] over large-scale CommonCrawl Corpus (CC)¹ data covering 100 languages. It follows the RoBERTa-

¹<https://commoncrawl.org>

style [117] training recipe, removing the next-sentence prediction objective and relying on large-scale unsupervised pretraining to learn contextualized multilingual representations. XLM-R is primarily designed as a general-purpose multilingual language encoder for cross-lingual transfer tasks such as question answering, sequence labeling, and classification. Nevertheless, its broad language coverage and strong contextual representations make it a useful text encoder for multilingual audio-language experiments.

LaBSE (Language-agnostic BERT Sentence Embedding) [56] is a multilingual sentence embedding model designed explicitly for cross-lingual semantic alignment. LaBSE is optimized to produce language-agnostic sentence-level representations in which translations are close in the embedding space. Its training combines multilingual language-model pretraining objectives, such as masked language modeling and translation language modeling, with a dual-encoder translation-ranking objective. In the dual-encoder setup, source and target sentences are encoded independently, and the model is trained to assign higher similarity to true translation pairs than to in-batch negative examples. LaBSE also incorporates additive margin softmax [208], which encourages stronger separation between true translations and semantically close non-translations, making LaBSE suitable for multilingual retrieval settings, since its embeddings are already structured around cross-lingual sentence similarity.

Multilingual E5 (mE5) [195] is a family of multilingual text embedding models designed for retrieval, semantic similarity, clustering, and related embedding-based applications. The training procedure follows a two-stage recipe. First, the models are trained with weakly supervised contrastive pretraining on approximately one billion multilingual text pairs collected from diverse sources, including Wikipedia², mC4 [205], multilingual news, NLLB [33], Reddit³, StackExchange⁴, and other sentence-pair resources. Second, the models are fine-tuned on a smaller set of high-quality labeled datasets, including retrieval, question answering, natural language inference, and multilingual supervision. During fine-tuning, the model uses contrastive learning with in-batch negatives, mined hard negatives, and knowledge distillation from a stronger cross-encoder.

mGTE (The multilingual General Text Embedding model) [219] is a long-context multilingual text encoder and retrieval model designed for multilingual text representation and reranking. mGTE is pretrained with a native long-context curriculum, reaching up to 8,192 tokens. Architecturally, it builds on a BERT-like encoder but introduces several efficiency and modeling modifications, including rotary position embeddings [181], gated linear units, and an unpadding strategy to reduce redundant computation over padding tokens. The model is first pretrained with masked language modeling

²<https://www.wikipedia.org/>

³<https://www.reddit.com>

⁴<https://stackexchange.com>

over multilingual data and then adapted for retrieval through contrastive pretraining and fine-tuning. In the retrieval setting, mGTE can produce dense representations and, in some variants, sparse token-level representations, making it useful for both semantic retrieval and long-document matching.

Encoder	Architecture	Pretraining / Training Method	Main Objective	Parameters
XLM-R - Base [31]	Transformer	Multilingual MLM	Token-level language modeling	~270M
LaBSE [56]	Transformer	MLM + TLM + translation ranking	Sentence-level cross-lingual alignment	~470M
mE5 - Base [195]	Transformer	Contrastive pretraining + supervised fine-tuning	Text-pair retrieval alignment	~278M
mGTE - Base [219]	Transformer	MLM + contrastive retrieval training	Long-context multilingual retrieval	~305M

Table A.4: *Comparison of the multilingual text encoders. The table summarizes the architecture family, training strategy, main learning objective, and approximate number of parameters of each encoder.*

Evaluation Protocol

This chapter details the multilingual evaluation protocol used for PALMA and CACARA, complementing the metric definitions of Section 4.3. It first presents the language-specific prompt templates used for zero-shot classification, and then details the retrieval evaluation algorithm used to compute Recall@1, Recall@5, Recall@10, and Recall@Avg.

B.1 Multilingual Zero-Shot Classification Templates

For zero-shot classification, each class name is converted into a text prompt before being encoded by the text encoder, and the predicted class is the one whose prompt embedding maximizes the cosine similarity with the audio embedding. Given a class label y , the prompt is constructed by concatenating a language-specific template with the translated class label. For a language ℓ , the final prompt is defined as:

$$p_\ell(y) = \text{Template}_\ell \parallel y, \quad (\text{B-1})$$

where $\parallel$ denotes string concatenation and Template_ℓ is the language-specific template.

Table B.1 shows the templates used in the multilingual zero-shot classification experiments. They cover English and the eleven additional languages used in the multilingual evaluation (see Section 4.3).

B.2 Retrieval Evaluation Protocol

For retrieval, the evaluation follows a grouped multi-caption protocol that generalizes the single-caption definition of Recall@ K given in Section 4.3. Each audio item has one audio embedding, while the text side may contain several captions associated with the same audio, as in AudioCaps [99] and ClothoV2 [45]. Let N be the number of unique audio samples and M the number of captions. The audio encoder produces normalized embeddings $\mathbf{A} \in \mathbb{R}^{N \times d}$ with rows $\mathbf{A}_i = \mathbf{z}_i^a$, and the text encoder produces normalized

Table B.1: *Language-specific prompt templates used for multilingual zero-shot classification. Each class label is appended to the corresponding template prefix before being encoded by the text encoder.*

embeddings $\mathbf{T} \in \mathbb{R}^{M \times d}$ with rows $\mathbf{T}_j = \mathbf{z}_j^t$. The similarity matrix is

$$\mathbf{S} = \mathbf{A}\mathbf{T}^\top, \quad (\text{B-2})$$

so that $\mathbf{S}_{i,j} = \text{sim}(\mathbf{z}_i^a, \mathbf{z}_j^t) = s_{ij}$ is the cosine similarity between audio i and caption j , matching the score s_{ij} of Section 4.3.

In audio-to-text retrieval ($a \rightarrow t$), an audio query is correct at rank K if at least one of its paired captions appears among the top- K retrieved captions. In text-to-audio retrieval ($t \rightarrow a$), each caption is a query and is correct at rank K if its paired audio appears among the top- K retrieved audios. When every audio has exactly one caption ($M = N$, $\mathcal{C}_i = \{i\}$, $a(i) = i$), the protocol reduces to the $\text{rank}_{a \rightarrow t}(i) \leq K$ and $\text{rank}_{t \rightarrow a}(i) \leq K$ conditions of the main text. Recall@Avg is the arithmetic mean of Recall@1, Recall@5, and Recall@10, computed separately for each direction.

Symbol	Meaning
N	Number of unique audio samples
M	Number of captions
d	Embedding dimension
$\mathbf{A} \in \mathbb{R}^{N \times d}$	Normalized audio embeddings, $\mathbf{A}_i = \mathbf{z}_i^a$
$\mathbf{T} \in \mathbb{R}^{M \times d}$	Normalized text embeddings, $\mathbf{T}_j = \mathbf{z}_j^t$
$\mathbf{S} \in \mathbb{R}^{N \times M}$	Similarity matrix, $\mathbf{S}_{i,j} = s_{ij} = \text{sim}(\mathbf{z}_i^a, \mathbf{z}_j^t)$
$\mathcal{C}_i$	Set of caption indices paired with audio i
$a(j)$	Audio index paired with caption j
$\text{TopK}(\mathbf{v}, K)$	Indices of the K largest entries of vector $\mathbf{v}$
$\mathbb{I}[\cdot]$	Indicator function

Algorithm 1 Grouped Audio–Text Retrieval Evaluation

Input: audio embeddings $\mathbf{A} \in \mathbb{R}^{N \times d}$, text embeddings $\mathbf{T} \in \mathbb{R}^{M \times d}$
Input: caption groups $\{\mathcal{C}_i\}_{i=1}^N$, caption-to-audio map $a(\cdot)$, ranks $\mathcal{K} = \{1, 5, 10\}$
Output: $\text{R@}K_{\text{a2t}}$ and $\text{R@}K_{\text{t2a}}$ for all $K \in \mathcal{K}$
Output: $\text{R@Avg}_{\text{a2t}}$ and $\text{R@Avg}_{\text{t2a}}$

```

    // 1. normalize embeddings
    1:  $\mathbf{A}_i \leftarrow \mathbf{A}_i / \|\mathbf{A}_i\|_2, \quad \forall i \in \{1, \dots, N\}$ 
    2:  $\mathbf{T}_j \leftarrow \mathbf{T}_j / \|\mathbf{T}_j\|_2, \quad \forall j \in \{1, \dots, M\}$ 
    // 2. cosine similarity matrix
    3:  $\mathbf{S} \leftarrow \mathbf{A}\mathbf{T}^\top$   $\triangleright \mathbf{S} \in \mathbb{R}^{N \times M}$ 
    // 3. audio-to-text retrieval
    4: for  $K \in \mathcal{K}$  do
    5:    $h \leftarrow 0$ 
    6:   for  $i \leftarrow 1$  to  $N$  do
    7:      $\mathcal{R} \leftarrow \text{TopK}(\mathbf{S}_{i,:}, K)$   $\triangleright$  top- $K$  captions
    8:      $h \leftarrow h + \mathbb{I}[\mathcal{R} \cap \mathcal{C}_i \neq \emptyset]$ 
    9:   end for
    10:   $\text{R@}K_{\text{a2t}} \leftarrow 100 \cdot h/N$ 
    11: end for
    // 4. text-to-audio retrieval
    12: for  $K \in \mathcal{K}$  do
    13:    $h \leftarrow 0$ 
    14:   for  $j \leftarrow 1$  to  $M$  do
    15:      $\mathcal{R} \leftarrow \text{TopK}(\mathbf{S}_{:,j}, K)$   $\triangleright$  top- $K$  audios
    16:      $h \leftarrow h + \mathbb{I}[a(j) \in \mathcal{R}]$ 
    17:   end for
    18:   $\text{R@}K_{\text{t2a}} \leftarrow 100 \cdot h/M$ 
    19: end for
    // 5. average recall across ranks
    20:  $\text{R@Avg}_{\text{a2t}} \leftarrow \frac{\text{R@}1_{\text{a2t}} + \text{R@}5_{\text{a2t}} + \text{R@}10_{\text{a2t}}}{3}$ 
    21:  $\text{R@Avg}_{\text{t2a}} \leftarrow \frac{\text{R@}1_{\text{t2a}} + \text{R@}5_{\text{t2a}} + \text{R@}10_{\text{t2a}}}{3}$ 
    22: return  $\{\text{R@}K_{\text{a2t}}\}_{K \in \mathcal{K}}, \{\text{R@}K_{\text{t2a}}\}_{K \in \mathcal{K}}, \text{R@Avg}_{\text{a2t}}, \text{R@Avg}_{\text{t2a}}$ 

```

Hyperparameters and Computing Resources

This chapter summarizes the hyperparameters and computational resources used in the CACARA and PALMA optimization pipelines. Since both frameworks involve multiple optimization stages, the configurations are organized into shared hyperparameters and phase-specific settings. The shared hyperparameter tables report the preprocessing and optimization values that remain fixed across most experiments, while the phase-specific tables describe the batch sizes, training lengths, hardware, evaluated configurations, and consolidated settings used in each phase.

C.1 CACARA

The CACARA optimization pipeline was organized around audio encoder consolidation, augmentation consolidation, basic training, and expanded-resource training. Table C.1 reports the shared hyperparameters used across the main CACARA training phases, while Table C.2 summarizes the phase-specific batch sizes, training lengths, and hardware configurations.

C.2 PALMA

The PALMA optimization pipeline follows a staged consolidation strategy in which each major modeling component is evaluated before moving to the next phase. This procedure includes audio encoder consolidation, hyperparameter tuning, text encoder consolidation, efficiency method exploration, layer aggregation, dataset consolidation, batch size consolidation, and multilingual supervision. Table C.3 reports the shared PALMA hyperparameters, while Table C.4 summarizes the phase-specific configurations evaluated during the optimization pipeline.

Hyperparameter	Value
Maximum text token length	77
Maximum audio length	10 seconds
Optimizer	Adam
Adam β	(0.9, 0.98)
Adam ϵ	1×10^{-8}
Weight decay	1×10^{-6}
Learning rate schedule	CosineWarmupLR / cosine decay with warmup
Warmup steps	500
Maximum learning rate	5×10^{-5} for the selected configurations
Minimum learning rate	1×10^{-5}
Learning-rate grid for encoder tuning	$\{5 \times 10^{-4}, 5 \times 10^{-5}, 5 \times 10^{-6}\}$
Final augmentation strategy	SpecAugment + Random Truncation

Table C.1: *Shared hyperparameters used across the main CACARA training phases.*

Phase	Batch size	Training length	Hardware
Phase 1	HTS-AT: 256 AudioMAE: 96 BEATs: 64	100k steps	RTX 8000, 48GB
Phase 2	64	100k steps	RTX 8000, 48GB
Phase 3	64	3 epochs	RTX 8000, 48GB
Phase 4	110	10 epochs	A100, 80GB

Table C.2: *Phase-specific batch sizes, training lengths, and hardware used in the CACARA optimization pipeline. Phase 1 corresponds to audio encoder consolidation and learning-rate tuning; Phase 2 corresponds to augmentation consolidation; Phase 3 corresponds to basic CACARA training; and Phase 4 corresponds to expanded-resource CACARA training.*

Hyperparameter	PALMA Configuration
Maximum text token length	77
Maximum audio length	10 seconds
Optimizer	Adam
Adam β	(0.9, 0.98)
Adam ϵ	1×10^{-8}
Learning rate schedule	CosineWarmupLR / cosine decay with warmup
Warmup steps	500
Batch size before batch-size consolidation	48
Learning rate candidates	$\{10^{-4}, 10^{-5}, 10^{-6}\}$
Weight decay candidates	$\{10^{-6}, 10^{-4}, 10^{-2}\}$
Batch size candidates	$\{24, 48, 64, 96, 128, 192, 256, 384, 512\}$

Table C.3: Shared hyperparameters used across the PALMA optimization pipeline. The learning rate and weight decay are selected from predefined candidate sets during the hyperparameter tuning phase, while the final batch size is selected during the batch size consolidation phase.

Phase	Batch size	Training length	Hardware
Phase 1	48	3 epochs	RTX A5000, 24GB
Phase 2	48	3 epochs	RTX A5000, 24GB
Phase 3	48	3 epochs	RTX A5000, 24GB
Phase 4	48	3 epochs	RTX A5000, 24GB
Phase 5	48	3 epochs	RTX A5000, 24GB
Phase 6	48	3 epochs	RTX A5000, 24GB
Phase 7	$\{24, 48, 64, 96, 128, 192, 256, 384, 512\}$; final: 128	3 epochs	A100, 80GB
Phase 8	128	10 epochs	A100, 80GB

Table C.4: Phase-specific batch sizes, training lengths, and hardware used in the PALMA optimization pipeline. Phase 1 corresponds to audio encoder consolidation; Phase 2 to hyperparameter tuning; Phase 3 to text encoder consolidation; Phase 4 to efficiency methods exploration; Phase 5 to layer aggregation; Phase 6 to dataset consolidation; Phase 7 to batch size consolidation; and Phase 8 to multilingual supervision. Before batch size consolidation, the batch size was fixed at 48. The final batch size selected during Phase 7 was 128, which was also used during the multilingual supervision experiments. All phases were trained for 3 epochs, except the multilingual supervision phase, which was trained for 10 epochs.

CACARA - Expanded Resources Result

This chapter presents the complete comparison between the expanded resources (ER) CACARA model configuration with the Optimized Basic Model (OBM).

Language	Optimized Basic Model						Expanded Resources Model					
	Audio to Text			Text to Audio			Audio to Text			Text to Audio		
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
AudioCaps												
English	33.98 ± 0.64	68.30 ± 0.43	81.81 ± 0.21	61.36 ± 0.27	7.30 ± 0.15	27.87 ± 0.32	41.21 ± 0.29	25.46 ± 0.16	31.45 ± 0.14	66.09 ± 0.31	79.61 ± 0.45	59.05 ± 0.28
Portuguese	21.62 ± 0.70	51.27 ± 1.31	66.34 ± 1.23	46.41 ± 1.00	5.74 ± 0.18	21.46 ± 0.18	33.47 ± 0.35	20.22 ± 0.11	18.54 ± 0.98	46.57 ± 0.75	60.96 ± 0.53	42.02 ± 0.74
Spanish	24.06 ± 0.48	53.94 ± 0.87	68.05 ± 0.35	48.68 ± 0.51	5.85 ± 0.16	22.58 ± 0.29	34.31 ± 0.25	20.91 ± 0.08	21.03 ± 0.38	50.85 ± 0.26	65.32 ± 0.80	45.73 ± 0.45
French	21.87 ± 0.70	51.20 ± 1.51	66.31 ± 1.29	46.46 ± 1.17	5.94 ± 0.05	22.97 ± 0.27	35.19 ± 0.47	21.37 ± 0.16	20.33 ± 0.42	48.61 ± 0.36	62.67 ± 0.38	43.87 ± 0.27
Russian	20.73 ± 0.47	48.96 ± 0.60	63.03 ± 0.74	44.24 ± 0.58	5.55 ± 0.31	19.48 ± 0.47	30.70 ± 0.39	18.58 ± 0.39	17.58 ± 0.35	44.19 ± 1.45	58.16 ± 0.89	39.97 ± 0.89
Arabic	15.25 ± 0.73	39.70 ± 1.05	53.90 ± 1.15	6.28 ± 0.97	4.71 ± 0.15	18.45 ± 0.23	28.27 ± 0.31	17.15 ± 0.10	11.18 ± 0.47	31.96 ± 0.86	45.17 ± 0.51	29.44 ± 0.58
Hindi	14.08 ± 0.41	37.47 ± 0.30	51.51 ± 0.80	34.35 ± 0.42	3.71 ± 0.08	14.97 ± 0.09	23.56 ± 0.07	14.08 ± 0.02	11.70 ± 0.30	33.33 ± 0.71	46.76 ± 0.76	30.60 ± 0.38
German	23.95 ± 0.41	54.36 ± 0.06	68.26 ± 0.14	48.86 ± 0.12	6.00 ± 0.09	23.09 ± 0.37	35.04 ± 0.45	21.38 ± 0.25	15.33 ± 0.13	50.33 ± 0.90	64.41 ± 0.90	45.36 ± 0.62
Chinese(zh)	18.24 ± 0.28	46.59 ± 1.11	62.04 ± 0.42	42.29 ± 0.47	5.45 ± 0.12	20.52 ± 0.20	31.70 ± 0.23	19.23 ± 0.09	21.27 ± 0.74	41.17 ± 1.37	56.36 ± 0.43	37.60 ± 0.80
Swahili	1.11 ± 0.23	3.96 ± 0.11	6.48 ± 0.24	3.85 ± 0.19	0.65 ± 0.07	2.29 ± 0.06	3.67 ± 0.08	2.20 ± 0.02	0.61 ± 0.18	2.89 ± 0.32	5.10 ± 0.18	2.87 ± 0.21
Japanese	21.36 ± 0.49	51.46 ± 0.24	65.64 ± 0.38	46.15 ± 0.22	5.59 ± 0.24	21.87 ± 0.48	33.82 ± 0.20	20.42 ± 0.30	18.35 ± 0.52	45.76 ± 0.96	59.78 ± 0.80	41.30 ± 0.68
Turkish	17.04 ± 0.40	42.65 ± 0.63	56.80 ± 0.45	38.84 ± 0.47	4.42 ± 0.23	17.22 ± 0.32	27.15 ± 0.51	16.27 ± 0.33	12.76 ± 0.15	35.07 ± 0.38	47.96 ± 0.51	31.93 ± 0.34
ClothoV2												
English	13.21 ± 3.16	33.63 ± 6.91	45.67 ± 8.51	30.84 ± 6.19	2.59 ± 0.74	10.07 ± 2.87	16.62 ± 4.68	9.76 ± 2.76	15.67 ± 0.50	38.46 ± 0.44	51.25 ± 0.06	35.13 ± 0.31
Portuguese	11.16 ± 0.08	30.27 ± 0.56	41.59 ± 0.48	27.68 ± 0.33	2.26 ± 0.17	9.47 ± 0.43	15.82 ± 0.50	9.18 ± 0.31	11.17 ± 0.20	30.45 ± 0.28	41.94 ± 0.72	27.85 ± 0.40
Spanish	11.67 ± 0.24	30.95 ± 0.33	43.00 ± 0.63	28.54 ± 0.21	2.30 ± 0.11	9.72 ± 0.37	15.67 ± 0.16	9.23 ± 0.19	11.67 ± 0.40	31.33 ± 0.25	42.58 ± 0.39	28.53 ± 0.23
French	10.30 ± 0.14	28.72 ± 0.06	40.27 ± 0.13	26.43 ± 0.06	2.28 ± 0.18	9.63 ± 0.39	15.67 ± 0.15	9.20 ± 0.16	10.45 ± 0.74	28.07 ± 1.20	39.78 ± 0.82	26.10 ± 0.80
Russian	9.53 ± 0.21	26.88 ± 0.35	38.57 ± 0.23	25.00 ± 0.15	2.11 ± 0.10	8.29 ± 0.24	13.70 ± 0.76	8.03 ± 0.35	10.09 ± 0.52	27.71 ± 0.76	39.13 ± 0.77	25.64 ± 0.66
Arabic	7.90 ± 0.19	23.31 ± 0.52	34.68 ± 0.03	21.96 ± 0.20	1.71 ± 0.26	7.06 ± 0.29	12.02 ± 0.34	6.93 ± 0.29	7.99 ± 0.10	23.64 ± 0.12	34.04 ± 0.28	21.89 ± 0.03
Hindi	6.55 ± 0.27	19.04 ± 0.26	28.74 ± 0.44	18.11 ± 0.29	1.56 ± 0.12	6.07 ± 0.20	10.26 ± 0.32	5.97 ± 0.10	6.81 ± 0.66	19.51 ± 0.89	28.32 ± 0.85	18.22 ± 0.79
German	10.95 ± 0.25	29.53 ± 0.63	40.91 ± 0.83	27.13 ± 0.57	2.62 ± 0.09	10.11 ± 0.22	16.54 ± 0.37	9.76 ± 0.21	11.22 ± 0.39	30.31 ± 0.74	41.44 ± 1.40	27.66 ± 0.81
Chinese(zh)	9.90 ± 0.35	27.41 ± 0.56	39.20 ± 0.55	25.51 ± 0.31	2.26 ± 0.05	8.76 ± 0.24	14.25 ± 0.20	8.43 ± 0.13	9.98 ± 0.15	27.28 ± 0.14	38.59 ± 0.80	25.28 ± 0.36
Swahili	0.78 ± 0.06	3.44 ± 0.26	5.61 ± 0.12	3.28 ± 0.11	0.42 ± 0.05	1.54 ± 0.07	2.43 ± 0.05	1.46 ± 0.02	0.75 ± 0.10	2.98 ± 0.07	5.20 ± 0.38	2.98 ± 0.10
Japanese	10.49 ± 0.35	28.92 ± 0.41	40.76 ± 0.76	26.72 ± 0.44	2.06 ± 0.14	8.86 ± 0.30	14.51 ± 0.43	8.48 ± 0.25	10.48 ± 0.23	28.87 ± 0.26	40.19 ± 0.44	26.51 ± 0.31
Turkish	8.60 ± 0.48	24.53 ± 0.82	35.75 ± 0.55	22.96 ± 0.60	1.74 ± 0.09	7.04 ± 0.09	12.13 ± 0.00	6.97 ± 0.06	8.45 ± 0.06	24.02 ± 0.50	34.61 ± 0.75	22.36 ± 0.41
English	13.21 ± 3.16	33.63 ± 6.91	45.67 ± 8.51	30.84 ± 6.19	2.59 ± 0.74	10.07 ± 2.87	16.62 ± 4.68	9.76 ± 2.76	15.67 ± 0.50	38.46 ± 0.44	51.25 ± 0.06	35.13 ± 0.31
Portuguese	11.16 ± 0.08	30.27 ± 0.56	41.59 ± 0.48	27.68 ± 0.33	2.26 ± 0.17	9.47 ± 0.43	15.82 ± 0.50	9.18 ± 0.31	11.17 ± 0.20	30.45 ± 0.28	41.94 ± 0.72	27.85 ± 0.40
Spanish	11.67 ± 0.24	30.95 ± 0.33	43.00 ± 0.63	28.54 ± 0.21	2.30 ± 0.11	9.72 ± 0.37	15.67 ± 0.16	9.23 ± 0.19	11.67 ± 0.40	31.33 ± 0.25	42.58 ± 0.39	28.53 ± 0.23
French	10.30 ± 0.14	28.72 ± 0.06	40.27 ± 0.13	26.43 ± 0.06	2.28 ± 0.18	9.63 ± 0.39	15.67 ± 0.15	9.20 ± 0.16	10.45 ± 0.74	28.07 ± 1.20	39.78 ± 0.82	26.10 ± 0.80
Russian	9.53 ± 0.21	26.88 ± 0.35	38.57 ± 0.23	25.00 ± 0.15	2.11 ± 0.10	8.29 ± 0.24	13.70 ± 0.76	8.03 ± 0.35	10.09 ± 0.52	27.71 ± 0.76	39.13 ± 0.77	25.64 ± 0.66
Arabic	7.90 ± 0.19	23.31 ± 0.52	34.68 ± 0.03	21.96 ± 0.20	1.71 ± 0.26	7.06 ± 0.29	12.02 ± 0.34	6.93 ± 0.29	7.99 ± 0.10	23.64 ± 0.12	34.04 ± 0.28	21.89 ± 0.03
Hindi	6.55 ± 0.27	19.04 ± 0.26	28.74 ± 0.44	18.11 ± 0.29	1.56 ± 0.12	6.07 ± 0.20	10.26 ± 0.32	5.97 ± 0.10	6.81 ± 0.66	19.51 ± 0.89	28.32 ± 0.85	18.22 ± 0.79
German	10.95 ± 0.25	29.53 ± 0.63	40.91 ± 0.83	27.13 ± 0.57	2.62 ± 0.09	10.11 ± 0.22	16.54 ± 0.37	9.76 ± 0.21	11.22 ± 0.39	30.31 ± 0.74	41.44 ± 1.40	27.66 ± 0.81
Chinese(zh)	9.90 ± 0.35	27.41 ± 0.56	39.20 ± 0.55	25.51 ± 0.31	2.26 ± 0.05	8.76 ± 0.24	14.25 ± 0.20	8.43 ± 0.13	9.98 ± 0.15	27.28 ± 0.14	38.59 ± 0.80	25.28 ± 0.36
Swahili	0.78 ± 0.06	3.44 ± 0.26	5.61 ± 0.12	3.28 ± 0.11	0.42 ± 0.05	1.54 ± 0.07	2.43 ± 0.05	1.46 ± 0.02	0.75 ± 0.10	2.98 ± 0.07	5.20 ± 0.38	2.98 ± 0.10
Japanese	10.49 ± 0.35	28.92 ± 0.41	40.76 ± 0.76	26.72 ± 0.44	2.06 ± 0.14	8.86 ± 0.30	14.51 ± 0.43	8.48 ± 0.25	10.48 ± 0.23	28.87 ± 0.26	40.19 ± 0.44	26.51 ± 0.31
Turkish	8.60 ± 0.48	24.53 ± 0.82	35.75 ± 0.55	22.96 ± 0.60	1.74 ± 0.09	7.04 ± 0.09	12.13 ± 0.00	6.97 ± 0.06	8.45 ± 0.06	24.02 ± 0.50	34.61 ± 0.75	22.36 ± 0.41
English	13.21 ± 3.16	33.63 ± 6.91	45.67 ± 8.51	30.84 ± 6.19	2.59 ± 0.74	10.07 ± 2.87	16.62 ± 4.68	9.76 ± 2.76	15.67 ± 0.50	38.46 ± 0.44	51.25 ± 0.06	35.13 ± 0.31
Portuguese	11.16 ± 0.08	30.27 ± 0.56	41.59 ± 0.48	27.68 ± 0.33	2.26 ± 0.17	9.47 ± 0.43	15.82 ± 0.50	9.18 ± 0.31	11.17 ± 0.20	30.45 ± 0.28	41.94 ± 0.72	27.85 ± 0.40
Spanish	11.67 ± 0.24	30.95 ± 0.33	43.00 ± 0.63	28.54 ± 0.21	2.30 ± 0.11	9.72 ± 0.37	15.67 ± 0.16	9.23 ± 0.19	11.67 ± 0.40	31.33 ± 0.25	42.58 ± 0.39	28.53 ± 0.23
French	10.30 ± 0.14	28.72 ± 0.06	40.27 ± 0.13	26.43 ± 0.06	2.28 ± 0.18	9.63 ± 0.39	15.67 ± 0.15	9.20 ± 0.16	10.45 ± 0.74	28.07 ± 1.20	39.78 ± 0.82	26.10 ± 0.80
Russian	9.53 ± 0.21	26.88 ± 0.35	38.57 ± 0.23	25.00 ± 0.15	2.11 ± 0.10	8.29 ± 0.24	13.70 ± 0.76	8.03 ± 0.35	10.09 ± 0.52	27.71 ± 0.76	39.13 ± 0.77	25.64 ± 0.66
Arabic	7.90 ± 0.19	23.31 ± 0.52	34.68 ± 0.03	21.96 ± 0.20	1.71 ± 0.26	7.06 ± 0.29	12.02 ± 0.34	6.93 ± 0.29	7.99 ± 0.10	23.64 ± 0.12	34.04 ± 0.28	21.89 ± 0.03
Hindi	6.55 ± 0.27	19.04 ± 0.26	28.74 ± 0.44	18.11 ± 0.29	1.56 ± 0.12	6.07 ± 0.20	10.26 ± 0.32	5.97 ± 0.10	6.81 ± 0.66	19.51 ± 0.89	28.32 ± 0.85	18.22 ± 0.79
German	10.95 ± 0.25	29.53 ± 0.63	40.91 ± 0.83	27.13 ± 0.57	2.62 ± 0.09	10.11 ± 0.22	16.54 ± 0.37	9.76 ± 0.21	11.22 ± 0.39	30.31 ± 0.74	41.44 ± 1.40	27.66 ± 0.81
Chinese(zh)	9.90 ± 0.35	27.41 ± 0.56	39.20 ± 0.55	25.51 ± 0.31	2.26 ± 0.05	8.76 ± 0.24	14.25 ± 0.20	8.43 ± 0.13	9.98 ± 0.15	27.28 ± 0.14	38.59 ± 0.80	25.28 ± 0.36
Swahili	0.78 ± 0.06	3.44 ± 0.26	5.61 ± 0.12	3.28 ± 0.11	0.42 ± 0.05	1.54 ± 0.07	2.43 ± 0.05	1.46 ± 0.02	0.75 ± 0.10	2.98 ± 0.07	5.20 ± 0.38	2.98 ± 0.10
Japanese	10.49 ± 0.35	28.92 ± 0.41	40.76 ± 0.76	26.72 ± 0.44	2.06 ± 0.14	8.86 ± 0.30	14.51 ± 0.43	8.48 ± 0.25	10.48 ± 0.23	28.87 ± 0.26	40.19 ± 0.44	26.51 ± 0.31
Turkish	8.60 ± 0.48	24.53 ± 0.82	35.75 ± 0.55	22.96 ± 0.60	1.74 ± 0.09	7.04 ± 0.09	12.13 ± 0.00	6.97 ± 0.06	8.45 ± 0.06	24.02 ± 0.50	34.61 ± 0.75	22.36 ± 0.41

Table D.1: Retrieval task results for expanded resources experiments, divided into two datasets: AudioCaps and ClothoV2, and two retrieval tasks: Audio-to-Text and Text-to-Audio across the twelve evaluated languages.

PALMA - Multilingual Results

This chapter presents complete and language-wise results of different configurations of PALMA considering audio-language retrieval considering audio-to-text and text-to-audio directions and zero-shot classification. Twelve languages are evaluated: English, Portuguese, Spanish, French, Russian, Arabic, Hindi, German, Chinese (Simplified), Swahili, Japanese and Turkish. The results are an average across three runs, reported alongside its standard deviation.

E.1 Retrieval

English-only								
AudioCaps								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	49.31 ± 1.58	81.14 ± 1.50	91.08 ± 1.11	73.84 ± 1.03	38.58 ± 0.23	73.85 ± 0.77	85.54 ± 0.39	65.99 ± 0.44
Portuguese	1.99 ± 0.34	6.97 ± 0.34	11.66 ± 0.36	6.87 ± 0.25	1.18 ± 0.15	4.53 ± 0.36	7.74 ± 0.59	4.48 ± 0.32
Spanish	1.87 ± 0.13	7.01 ± 1.14	10.61 ± 1.55	6.50 ± 0.88	1.30 ± 0.11	4.39 ± 0.68	7.31 ± 0.99	4.33 ± 0.59
French	<u>3.52</u> ± 0.47	<u>10.46</u> ± 0.59	<u>16.76</u> ± 1.37	<u>10.25</u> ± 0.62	<u>2.02</u> ± 0.08	<u>7.78</u> ± 0.53	<u>12.64</u> ± 0.99	<u>7.48</u> ± 0.47
Russian	0.07 ± 0.13	0.75 ± 0.32	1.24 ± 0.39	0.69 ± 0.21	0.08 ± 0.02	0.59 ± 0.09	1.18 ± 0.14	0.62 ± 0.07
Arabic	0.11 ± 0.11	0.45 ± 0.11	1.12 ± 0.34	0.56 ± 0.10	0.10 ± 0.02	0.57 ± 0.16	1.28 ± 0.18	0.65 ± 0.12
Hindi	0.11 ± 0.11	0.41 ± 0.17	1.16 ± 0.23	0.56 ± 0.17	0.09 ± 0.02	0.58 ± 0.10	1.15 ± 0.06	0.60 ± 0.05
German	1.99 ± 0.28	7.08 ± 0.98	11.10 ± 0.68	6.72 ± 0.58	1.17 ± 0.14	4.73 ± 0.52	8.18 ± 0.64	4.69 ± 0.40
Chinese(zh)	0.22 ± 0.12	0.67 ± 0.40	1.39 ± 0.53	0.76 ± 0.34	0.16 ± 0.12	0.62 ± 0.12	1.18 ± 0.31	0.65 ± 0.18
Swahili	2.89 ± 0.28	6.33 ± 0.23	9.15 ± 0.47	6.12 ± 0.17	0.55 ± 0.05	2.41 ± 0.26	3.73 ± 0.33	2.23 ± 0.21
Japanese	0.15 ± 0.06	0.38 ± 0.06	1.39 ± 0.29	0.64 ± 0.14	0.12 ± 0.01	0.65 ± 0.13	1.15 ± 0.20	0.64 ± 0.11
Turkish	0.94 ± 0.40	2.92 ± 0.34	4.87 ± 0.62	2.91 ± 0.34	0.32 ± 0.02	1.61 ± 0.26	3.26 ± 0.63	1.73 ± 0.30
ClothoV2								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	19.68 ± 0.80	43.92 ± 0.51	57.38 ± 0.64	40.33 ± 0.60	14.76 ± 0.21	36.42 ± 0.36	49.42 ± 0.25	33.53 ± 0.19
Portuguese	0.99 ± 0.24	3.41 ± 0.83	5.84 ± 1.04	3.41 ± 0.56	0.72 ± 0.05	2.24 ± 0.28	3.66 ± 0.46	2.21 ± 0.23
Spanish	1.56 ± 0.30	4.75 ± 0.43	7.98 ± 0.55	4.76 ± 0.41	0.70 ± 0.18	2.95 ± 0.51	4.84 ± 0.59	2.83 ± 0.42
French	<u>2.07</u> ± 0.39	<u>6.73</u> ± 0.69	<u>10.59</u> ± 1.25	<u>6.46</u> ± 0.74	<u>1.16</u> ± 0.21	<u>4.30</u> ± 0.27	<u>7.22</u> ± 0.34	<u>4.23</u> ± 0.22
Russian	0.13 ± 0.11	0.32 ± 0.20	0.89 ± 0.24	0.45 ± 0.18	0.11 ± 0.04	0.51 ± 0.06	0.87 ± 0.02	0.50 ± 0.02
Arabic	0.16 ± 0.05	0.54 ± 0.20	0.96 ± 0.33	0.55 ± 0.19	0.09 ± 0.02	0.47 ± 0.03	0.96 ± 0.08	0.51 ± 0.03
Hindi	0.13 ± 0.05	0.45 ± 0.20	0.93 ± 0.06	0.50 ± 0.05	0.13 ± 0.04	0.49 ± 0.06	0.98 ± 0.15	0.53 ± 0.07
German	1.11 ± 0.22	3.22 ± 0.34	5.20 ± 0.44	3.18 ± 0.27	0.56 ± 0.12	2.35 ± 0.15	3.90 ± 0.17	2.27 ± 0.14
Chinese(zh)	0.13 ± 0.05	0.51 ± 0.11	0.86 ± 0.10	0.50 ± 0.03	0.12 ± 0.04	0.56 ± 0.08	1.04 ± 0.13	0.57 ± 0.06
Swahili	1.25 ± 0.33	3.03 ± 0.98	4.37 ± 0.97	2.88 ± 0.76	0.34 ± 0.06	1.31 ± 0.24	2.21 ± 0.49	1.29 ± 0.24
Japanese	0.10 ± 0.00	0.41 ± 0.27	0.67 ± 0.38	0.39 ± 0.21	0.09 ± 0.02	0.47 ± 0.09	0.87 ± 0.07	0.47 ± 0.05
Turkish	0.48 ± 0.10	1.88 ± 0.15	3.29 ± 0.39	1.88 ± 0.19	0.28 ± 0.06	1.27 ± 0.09	2.15 ± 0.04	1.23 ± 0.06

Table E.1: Language-wise retrieval performance for the English-only strategy. AudioCaps and ClothoV2 are shown in separate blocks. A2T and T2A denote Audio-to-Text and Text-to-Audio retrieval, respectively. Values are reported as Recall@k mean ± standard deviation across three runs. The best value for each metric within each dataset block is shown in bold, while the second-best value is underlined.

Random Sampling								
AudioCaps								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	53.81 $\pm$ 0.81	83.62 $\pm$ 0.51	93.25 $\pm$ 1.08	76.89 $\pm$ 0.43	39.59 $\pm$ 0.75	75.33 $\pm$ 0.80	87.16 $\pm$ 0.24	67.36 $\pm$ 0.45
Portuguese	<u>53.39</u> $\pm$ 0.06	<u>83.24</u> $\pm$ 0.70	91.72 $\pm$ 0.06	<u>76.12</u> $\pm$ 0.23	37.80 $\pm$ 0.40	73.01 $\pm$ 0.26	85.07 $\pm$ 0.21	65.29 $\pm$ 0.29
Spanish	50.62 $\pm$ 1.19	81.40 $\pm$ 0.66	91.45 $\pm$ 0.60	74.49 $\pm$ 0.28	37.32 $\pm$ 1.08	72.55 $\pm$ 0.58	84.97 $\pm$ 0.29	64.94 $\pm$ 0.58
French	52.38 $\pm$ 0.28	82.04 $\pm$ 1.24	91.07 $\pm$ 1.62	75.16 $\pm$ 1.00	37.66 $\pm$ 0.50	72.93 $\pm$ 0.26	85.44 $\pm$ 0.27	65.34 $\pm$ 0.14
Russian	50.84 $\pm$ 0.88	81.66 $\pm$ 1.47	91.94 $\pm$ 0.40	74.82 $\pm$ 0.71	37.57 $\pm$ 0.20	72.30 $\pm$ 0.36	84.75 $\pm$ 0.55	64.87 $\pm$ 0.21
Arabic	51.52 $\pm$ 0.70	81.92 $\pm$ 0.88	92.05 $\pm$ 0.28	75.17 $\pm$ 0.55	37.23 $\pm$ 0.63	72.68 $\pm$ 0.41	84.94 $\pm$ 0.25	64.95 $\pm$ 0.39
Hindi	51.78 $\pm$ 1.77	81.07 $\pm$ 1.50	90.93 $\pm$ 0.94	74.59 $\pm$ 1.20	37.35 $\pm$ 0.52	72.82 $\pm$ 0.27	85.50 $\pm$ 0.30	65.22 $\pm$ 0.23
German	52.16 $\pm$ 0.17	82.30 $\pm$ 1.01	<u>92.20</u> $\pm$ 1.05	75.55 $\pm$ 0.71	<u>38.20</u> $\pm$ 0.70	<u>73.51</u> $\pm$ 0.43	<u>86.25</u> $\pm$ 0.17	<u>65.98</u> $\pm$ 0.29
Chinese(zh)	50.88 $\pm$ 1.24	81.48 $\pm$ 0.45	91.41 $\pm$ 0.26	74.59 $\pm$ 0.49	36.84 $\pm$ 0.35	72.21 $\pm$ 0.07	85.28 $\pm$ 0.43	64.78 $\pm$ 0.28
Swahili	49.12 $\pm$ 1.54	80.61 $\pm$ 1.80	91.34 $\pm$ 0.49	73.69 $\pm$ 0.75	35.69 $\pm$ 0.29	70.32 $\pm$ 0.32	83.19 $\pm$ 0.19	63.06 $\pm$ 0.25
Japanese	51.26 $\pm$ 0.28	81.96 $\pm$ 1.04	91.15 $\pm$ 0.72	74.79 $\pm$ 0.67	36.81 $\pm$ 0.77	72.05 $\pm$ 0.19	85.36 $\pm$ 0.49	64.74 $\pm$ 0.34
Turkish	51.74 $\pm$ 1.14	82.04 $\pm$ 0.13	91.60 $\pm$ 0.69	75.13 $\pm$ 0.53	37.39 $\pm$ 0.52	72.42 $\pm$ 0.28	85.83 $\pm$ 0.37	65.21 $\pm$ 0.35
ClothoV2								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	21.05 $\pm$ 1.23	43.67 $\pm$ 0.48	57.57 $\pm$ 1.07	40.76 $\pm$ 0.92	16.31 $\pm$ 0.20	39.39 $\pm$ 0.35	52.56 $\pm$ 0.25	36.08 $\pm$ 0.17
Portuguese	<u>20.38</u> $\pm$ 0.72	<u>43.64</u> $\pm$ 1.34	<u>56.46</u> $\pm$ 0.42	<u>40.16</u> $\pm$ 0.77	15.83 $\pm$ 0.28	38.98 $\pm$ 0.44	<u>52.15</u> $\pm$ 0.44	35.65 $\pm$ 0.26
Spanish	20.29 $\pm$ 0.82	42.87 $\pm$ 0.78	55.95 $\pm$ 0.38	39.70 $\pm$ 0.40	15.87 $\pm$ 0.45	<u>39.02</u> $\pm$ 0.18	52.10 $\pm$ 0.80	<u>35.67</u> $\pm$ 0.29
French	19.07 $\pm$ 0.72	41.98 $\pm$ 0.20	55.50 $\pm$ 1.11	38.85 $\pm$ 0.45	15.33 $\pm$ 0.57	38.02 $\pm$ 0.47	51.53 $\pm$ 0.06	34.96 $\pm$ 0.29
Russian	19.46 $\pm$ 1.25	41.57 $\pm$ 0.87	55.09 $\pm$ 1.21	38.71 $\pm$ 1.10	15.03 $\pm$ 0.56	37.69 $\pm$ 0.35	50.65 $\pm$ 0.16	34.46 $\pm$ 0.20
Arabic	19.17 $\pm$ 0.80	41.63 $\pm$ 1.08	55.22 $\pm$ 0.77	38.67 $\pm$ 0.80	14.81 $\pm$ 0.30	37.10 $\pm$ 0.47	50.42 $\pm$ 0.17	34.11 $\pm$ 0.12
Hindi	19.97 $\pm$ 0.69	42.71 $\pm$ 1.55	55.66 $\pm$ 1.14	39.45 $\pm$ 1.04	<u>15.97</u> $\pm$ 0.33	38.15 $\pm$ 0.76	51.13 $\pm$ 0.44	35.09 $\pm$ 0.46
German	19.20 $\pm$ 0.40	42.55 $\pm$ 1.45	55.85 $\pm$ 1.36	39.20 $\pm$ 1.03	15.96 $\pm$ 0.38	38.76 $\pm$ 0.44	51.62 $\pm$ 0.52	35.45 $\pm$ 0.20
Chinese(zh)	19.30 $\pm$ 1.30	41.88 $\pm$ 0.81	54.26 $\pm$ 1.63	38.48 $\pm$ 0.94	15.13 $\pm$ 0.42	38.06 $\pm$ 0.18	51.18 $\pm$ 0.22	34.79 $\pm$ 0.06
Swahili	18.41 $\pm$ 1.38	41.69 $\pm$ 0.45	54.45 $\pm$ 1.41	38.18 $\pm$ 1.02	14.17 $\pm$ 0.41	36.38 $\pm$ 0.21	48.92 $\pm$ 0.07	33.16 $\pm$ 0.19
Japanese	19.11 $\pm$ 0.14	42.17 $\pm$ 0.82	54.80 $\pm$ 0.58	38.69 $\pm$ 0.33	15.19 $\pm$ 0.18	38.48 $\pm$ 0.52	51.55 $\pm$ 0.29	35.07 $\pm$ 0.22
Turkish	19.55 $\pm$ 1.15	41.79 $\pm$ 0.48	54.71 $\pm$ 1.36	38.68 $\pm$ 0.32	15.25 $\pm$ 0.18	37.47 $\pm$ 0.26	50.01 $\pm$ 0.37	34.24 $\pm$ 0.18

Table E.2: Language-wise retrieval performance for the Random Sampling strategy. AudioCaps and ClothoV2 are shown in separate blocks. A2T and T2A denote Audio-to-Text and Text-to-Audio retrieval, respectively. Values are reported as Recall@k mean $\pm$ standard deviation across three runs. The best value for each metric within each dataset block is shown in bold, while the second-best value is underlined.

LE-Mixture 10%								
AudioCaps								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	51.03 ± 1.23	82.71 ± 0.23	92.20 ± 0.26	75.31 ± 0.34	39.01 ± 0.59	73.77 ± 0.51	86.29 ± 0.19	66.35 ± 0.37
Portuguese	45.63 ± 1.41	77.47 ± 0.75	88.71 ± 0.13	70.61 ± 0.55	31.76 ± 0.10	65.76 ± 0.14	79.41 ± 0.05	58.98 ± 0.05
Spanish	44.17 ± 0.94	76.08 ± 0.72	87.37 ± 0.55	69.21 ± 0.60	30.98 ± 0.44	64.51 ± 0.29	78.89 ± 0.27	58.13 ± 0.33
French	<u>45.90</u> ± 1.49	77.95 ± 0.40	88.64 ± 0.58	<u>70.83</u> ± 0.51	<u>32.59</u> ± 0.58	<u>67.55</u> ± 0.28	80.83 ± 0.36	<u>60.32</u> ± 0.34
Russian	40.01 ± 0.45	73.00 ± 0.89	85.26 ± 1.17	66.09 ± 0.55	27.34 ± 0.20	60.00 ± 0.45	74.11 ± 0.34	53.82 ± 0.28
Arabic	42.75 ± 2.08	74.62 ± 1.24	85.68 ± 0.66	67.68 ± 1.08	29.87 ± 0.18	62.85 ± 0.85	77.18 ± 0.75	56.63 ± 0.58
Hindi	45.56 ± 0.34	76.87 ± 0.74	87.85 ± 0.70	70.09 ± 0.35	31.59 ± 0.90	67.38 ± 0.53	<u>81.22</u> ± 0.70	60.06 ± 0.61
German	44.92 ± 1.15	<u>78.40</u> ± 0.68	<u>89.02</u> ± 0.51	70.78 ± 0.69	32.48 ± 0.72	66.31 ± 0.56	80.48 ± 0.68	59.76 ± 0.52
Chinese(zh)	42.67 ± 0.36	76.19 ± 1.02	87.66 ± 0.37	68.84 ± 0.55	30.89 ± 1.04	64.69 ± 0.16	79.37 ± 0.71	58.31 ± 0.38
Swahili	41.43 ± 0.79	74.20 ± 0.81	86.69 ± 0.53	67.44 ± 0.27	29.87 ± 0.29	63.01 ± 0.48	76.98 ± 0.75	56.62 ± 0.43
Japanese	42.89 ± 0.26	75.07 ± 0.79	87.10 ± 1.24	68.35 ± 0.37	31.16 ± 0.53	66.44 ± 0.18	80.52 ± 0.30	59.37 ± 0.33
Turkish	45.52 ± 1.32	77.36 ± 1.01	88.60 ± 1.35	70.49 ± 1.18	31.39 ± 0.13	66.12 ± 0.94	79.72 ± 0.45	59.08 ± 0.41
ClothoV2								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	18.02 ± 0.56	41.40 ± 0.15	54.96 ± 0.59	38.13 ± 0.40	14.72 ± 0.47	36.34 ± 0.62	49.18 ± 0.54	33.41 ± 0.50
Portuguese	15.76 ± 0.82	35.41 ± 1.01	47.91 ± 0.87	33.03 ± 0.76	12.11 ± 0.13	31.32 ± 0.68	42.68 ± 0.68	28.70 ± 0.47
Spanish	15.54 ± 0.97	36.08 ± 1.17	47.94 ± 0.53	33.19 ± 0.67	12.33 ± 0.30	31.50 ± 0.58	43.18 ± 0.51	29.00 ± 0.38
French	<u>16.01</u> ± 1.01	<u>36.30</u> ± 0.48	48.16 ± 0.06	<u>33.49</u> ± 0.43	<u>12.70</u> ± 0.46	<u>31.84</u> ± 0.03	<u>43.92</u> ± 0.31	<u>29.49</u> ± 0.24
Russian	13.14 ± 0.72	31.29 ± 1.18	41.79 ± 0.81	28.74 ± 0.75	9.81 ± 0.06	26.06 ± 0.39	36.72 ± 0.57	24.20 ± 0.24
Arabic	13.81 ± 1.40	33.04 ± 0.39	45.77 ± 0.20	30.87 ± 0.52	10.76 ± 0.19	28.10 ± 0.38	39.20 ± 0.40	26.02 ± 0.20
Hindi	15.37 ± 0.39	36.11 ± 1.94	<u>48.84</u> ± 0.99	33.44 ± 1.07	11.64 ± 0.54	30.11 ± 0.20	41.82 ± 0.63	27.86 ± 0.33
German	15.57 ± 0.78	35.38 ± 0.72	47.12 ± 0.77	32.69 ± 0.21	11.63 ± 0.23	29.59 ± 0.27	40.65 ± 0.30	27.29 ± 0.05
Chinese(zh)	13.01 ± 0.42	31.58 ± 1.26	44.85 ± 1.74	29.81 ± 0.99	10.69 ± 0.27	28.20 ± 0.20	39.90 ± 0.33	26.26 ± 0.17
Swahili	14.67 ± 0.14	34.00 ± 0.74	46.16 ± 1.25	31.61 ± 0.59	11.03 ± 0.42	28.84 ± 0.49	39.40 ± 0.32	26.43 ± 0.36
Japanese	14.67 ± 1.06	33.62 ± 2.19	47.18 ± 2.01	31.82 ± 1.75	11.60 ± 0.29	30.34 ± 0.12	41.83 ± 0.37	27.92 ± 0.20
Turkish	14.51 ± 0.77	35.18 ± 0.81	47.88 ± 0.44	32.52 ± 0.40	11.32 ± 0.28	28.93 ± 0.18	40.08 ± 0.19	26.78 ± 0.04

Table E.3: Language-wise retrieval performance for the LE-Mixture 10% strategy. AudioCaps and ClothoV2 are shown in separate blocks. A2T and T2A denote Audio-to-Text and Text-to-Audio retrieval, respectively. Values are reported as Recall@k mean ± standard deviation across three runs. The best value for each metric within each dataset block is shown in bold, while the second-best value is underlined.

LE-Mixture 20%								
AudioCaps								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	52.12 ± 1.82	83.43 ± 1.43	92.50 ± 0.66	76.02 ± 1.12	39.54 ± 0.39	74.49 ± 0.34	86.78 ± 0.41	66.94 ± 0.37
Portuguese	48.56 ± 1.13	80.69 ± 1.57	90.51 ± 0.51	73.25 ± 0.92	34.50 ± 0.90	69.22 ± 1.12	82.20 ± 0.72	61.98 ± 0.77
Spanish	47.13 ± 0.81	78.37 ± 0.37	89.01 ± 0.94	71.50 ± 0.13	33.83 ± 0.41	68.36 ± 0.78	82.04 ± 0.26	61.41 ± 0.42
French	<u>50.09</u> ± 1.89	80.80 ± 1.42	<u>91.08</u> ± 0.97	<u>73.99</u> ± 1.34	<u>35.11</u> ± 0.88	<u>70.06</u> ± 0.96	<u>83.34</u> ± 0.65	<u>62.83</u> ± 0.77
Russian	44.24 ± 1.50	76.01 ± 0.55	87.81 ± 1.40	69.35 ± 0.15	31.15 ± 0.06	65.19 ± 1.24	79.19 ± 0.63	58.51 ± 0.62
Arabic	47.85 ± 1.15	78.78 ± 0.26	89.17 ± 1.31	71.93 ± 0.15	33.08 ± 0.86	67.52 ± 1.27	81.16 ± 0.60	60.59 ± 0.89
Hindi	47.43 ± 1.02	78.82 ± 1.14	90.33 ± 1.07	72.19 ± 0.43	34.46 ± 0.70	69.94 ± 0.55	82.92 ± 0.56	62.44 ± 0.58
German	48.59 ± 0.30	<u>80.95</u> ± 0.94	90.89 ± 0.56	73.48 ± 0.28	34.40 ± 1.07	69.58 ± 0.62	83.19 ± 0.55	62.39 ± 0.70
Chinese(zh)	46.23 ± 1.14	79.23 ± 1.07	88.68 ± 1.05	71.38 ± 1.05	33.39 ± 0.78	68.42 ± 0.61	82.56 ± 0.58	61.46 ± 0.65
Swahili	46.53 ± 0.99	78.18 ± 0.20	88.41 ± 0.20	71.04 ± 0.22	32.54 ± 0.52	65.80 ± 0.91	79.93 ± 0.51	59.42 ± 0.59
Japanese	45.71 ± 1.25	78.33 ± 0.75	88.94 ± 1.10	70.99 ± 0.75	34.38 ± 0.39	68.99 ± 0.86	82.88 ± 0.57	62.08 ± 0.53
Turkish	48.22 ± 0.62	79.26 ± 1.01	90.66 ± 0.52	72.72 ± 0.54	34.05 ± 0.09	68.59 ± 0.43	82.29 ± 0.56	61.64 ± 0.32
ClothoV2								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	<u>17.86</u> ± 0.39	42.74 ± 1.52	55.28 ± 0.34	38.63 ± 0.66	15.02 ± 0.20	36.42 ± 0.85	49.46 ± 0.35	33.64 ± 0.45
Portuguese	16.87 ± 1.82	<u>38.88</u> ± 1.25	<u>52.72</u> ± 1.01	<u>36.16</u> ± 1.36	13.36 ± 0.37	33.78 ± 0.37	46.12 ± 0.39	31.09 ± 0.16
Spanish	17.07 ± 0.97	38.34 ± 0.53	50.94 ± 1.39	35.45 ± 0.73	13.57 ± 0.58	<u>34.12</u> ± 0.24	<u>47.07</u> ± 0.30	<u>31.59</u> ± 0.22
French	17.26 ± 0.49	37.99 ± 0.96	51.99 ± 0.52	35.75 ± 0.35	13.34 ± 0.37	33.45 ± 0.25	46.43 ± 0.43	31.07 ± 0.26
Russian	15.41 ± 0.63	34.64 ± 0.19	47.08 ± 0.50	32.38 ± 0.33	11.62 ± 0.27	29.58 ± 0.32	41.22 ± 0.13	27.48 ± 0.20
Arabic	15.92 ± 0.82	37.45 ± 0.53	50.37 ± 1.60	34.58 ± 0.78	12.26 ± 0.11	31.74 ± 0.19	43.55 ± 0.44	29.19 ± 0.17
Hindi	18.05 ± 1.01	38.76 ± 0.83	51.42 ± 1.07	36.08 ± 0.54	<u>13.63</u> ± 0.28	33.36 ± 0.86	45.58 ± 0.38	30.85 ± 0.48
German	16.40 ± 0.80	37.99 ± 0.50	50.49 ± 0.15	34.96 ± 0.37	13.29 ± 0.11	32.77 ± 0.83	44.33 ± 0.44	30.13 ± 0.45
Chinese(zh)	15.22 ± 0.29	35.73 ± 0.78	48.48 ± 0.34	33.14 ± 0.30	12.69 ± 0.11	31.42 ± 0.45	43.38 ± 0.54	29.17 ± 0.34
Swahili	15.76 ± 1.11	36.94 ± 0.63	49.66 ± 0.42	34.12 ± 0.37	11.99 ± 0.55	30.95 ± 0.28	42.31 ± 0.34	28.42 ± 0.37
Japanese	16.84 ± 0.19	38.25 ± 1.21	50.62 ± 2.45	35.24 ± 1.24	13.33 ± 0.53	33.35 ± 0.10	45.27 ± 0.35	30.65 ± 0.24
Turkish	16.49 ± 0.71	37.03 ± 1.09	49.64 ± 0.80	34.38 ± 0.11	12.62 ± 0.17	32.23 ± 0.66	43.77 ± 0.17	29.54 ± 0.22

Table E.4: Language-wise retrieval performance for the LE-Mixture 20% strategy. AudioCaps and ClothoV2 are shown in separate blocks. A2T and T2A denote Audio-to-Text and Text-to-Audio retrieval, respectively. Values are reported as Recall@k mean ± standard deviation across three runs. The best value for each metric within each dataset block is shown in bold, while the second-best value is underlined.

LE-Mixture 30%								
AudioCaps								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	51.29 ± 1.37	83.69 ± 0.41	92.77 ± 0.55	75.92 ± 0.67	39.17 ± 0.24	74.72 ± 0.67	87.03 ± 0.42	66.97 ± 0.39
Portuguese	50.36 ± 1.06	80.99 ± 1.09	91.08 ± 0.57	74.14 ± 0.72	35.48 ± 0.21	71.19 ± 0.21	84.08 ± 0.51	63.58 ± 0.31
Spanish	47.81 ± 1.19	80.58 ± 0.91	90.33 ± 0.41	72.90 ± 0.36	34.94 ± 0.57	70.25 ± 0.09	83.31 ± 0.21	62.83 ± 0.16
French	49.31 ± 1.25	80.99 ± 0.88	90.40 ± 0.91	73.57 ± 0.91	35.58 ± 0.51	71.05 ± 0.18	84.00 ± 0.39	63.54 ± 0.18
Russian	47.09 ± 0.43	79.38 ± 1.69	89.35 ± 1.45	71.94 ± 0.97	34.07 ± 0.44	68.17 ± 0.51	81.89 ± 0.32	61.37 ± 0.16
Arabic	49.68 ± 1.34	80.24 ± 0.92	90.33 ± 0.68	73.41 ± 0.68	34.96 ± 0.33	70.20 ± 0.46	83.18 ± 0.17	62.78 ± 0.08
Hindi	48.63 ± 0.83	80.24 ± 0.68	90.55 ± 0.30	73.14 ± 0.21	35.64 ± 0.87	71.09 ± 0.42	84.41 ± 0.33	63.71 ± 0.35
German	<u>51.14</u> ± 1.01	81.44 ± 1.19	<u>91.90</u> ± 0.81	<u>74.83</u> ± 0.28	<u>35.94</u> ± 0.58	<u>71.56</u> ± 0.49	<u>84.72</u> ± 0.06	<u>64.08</u> ± 0.34
Chinese(zh)	48.63 ± 1.81	80.54 ± 1.36	90.44 ± 0.49	73.21 ± 1.00	34.82 ± 0.35	70.85 ± 0.41	83.79 ± 0.36	63.15 ± 0.18
Swahili	46.91 ± 0.85	78.18 ± 0.74	89.17 ± 0.23	71.42 ± 0.38	33.00 ± 0.13	67.69 ± 0.87	81.82 ± 0.26	60.83 ± 0.42
Japanese	46.98 ± 1.39	80.28 ± 1.02	90.52 ± 0.98	72.59 ± 1.05	35.52 ± 0.30	70.54 ± 0.61	84.06 ± 0.26	63.37 ± 0.22
Turkish	49.76 ± 1.02	<u>81.85</u> ± 1.36	90.97 ± 0.96	74.19 ± 0.88	35.44 ± 0.35	70.66 ± 0.75	83.97 ± 0.36	63.35 ± 0.29
ClothoV2								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	19.17 ± 0.34	43.22 ± 0.77	56.24 ± 1.13	39.54 ± 0.67	15.37 ± 0.68	37.43 ± 0.61	49.78 ± 0.52	34.20 ± 0.34
Portuguese	17.99 ± 1.15	40.09 ± 0.51	53.14 ± 0.54	37.08 ± 0.40	14.23 ± 0.23	35.64 ± 0.40	48.40 ± 0.31	32.75 ± 0.27
Spanish	18.76 ± 1.23	40.29 ± 2.32	<u>53.72</u> ± 1.72	<u>37.59</u> ± 1.74	14.33 ± 0.46	<u>36.08</u> ± 0.35	<u>48.66</u> ± 0.15	<u>33.02</u> ± 0.28
French	18.47 ± 1.10	40.00 ± 0.35	52.18 ± 1.04	36.88 ± 0.27	13.99 ± 0.40	35.32 ± 0.29	48.13 ± 0.52	32.48 ± 0.35
Russian	16.33 ± 0.67	36.24 ± 1.52	49.95 ± 1.92	34.17 ± 1.02	12.61 ± 0.06	32.47 ± 0.40	44.51 ± 0.38	29.87 ± 0.19
Arabic	17.16 ± 0.72	39.14 ± 1.22	51.32 ± 1.27	35.88 ± 0.82	13.11 ± 0.46	33.97 ± 0.49	46.11 ± 0.22	31.07 ± 0.38
Hindi	<u>18.85</u> ± 0.50	<u>40.44</u> ± 2.02	52.60 ± 1.69	37.30 ± 1.30	14.28 ± 0.32	34.94 ± 0.61	47.55 ± 0.75	32.25 ± 0.52
German	17.48 ± 0.39	39.42 ± 1.10	52.54 ± 0.78	36.48 ± 0.65	13.85 ± 0.17	34.88 ± 0.19	47.01 ± 0.31	31.91 ± 0.08
Chinese(zh)	17.32 ± 0.58	37.51 ± 0.92	50.37 ± 1.82	35.07 ± 0.94	13.55 ± 0.34	33.24 ± 0.25	46.02 ± 0.26	30.94 ± 0.26
Swahili	17.86 ± 0.39	40.00 ± 0.26	51.48 ± 0.88	36.45 ± 0.18	12.86 ± 0.11	32.48 ± 0.56	44.41 ± 0.35	29.91 ± 0.32
Japanese	18.76 ± 0.29	39.55 ± 1.20	52.86 ± 0.87	37.05 ± 0.63	<u>14.37</u> ± 0.17	34.98 ± 0.61	47.26 ± 0.55	32.20 ± 0.41
Turkish	18.44 ± 1.05	40.13 ± 0.87	51.29 ± 0.19	36.62 ± 0.10	13.77 ± 0.28	34.07 ± 0.56	45.83 ± 0.71	31.22 ± 0.40

Table E.5: Language-wise retrieval performance for the LE-Mixture 30% strategy. AudioCaps and ClothoV2 are shown in separate blocks. A2T and T2A denote Audio-to-Text and Text-to-Audio retrieval, respectively. Values are reported as Recall@k mean ± standard deviation across three runs. The best value for each metric within each dataset block is shown in bold, while the second-best value is underlined.

LE-Mixture 40%								
AudioCaps								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	52.83 ± 2.02	84.51 ± 0.75	92.95 ± 0.57	76.77 ± 0.78	38.99 ± 1.28	74.32 ± 0.52	86.60 ± 0.26	66.64 ± 0.69
Portuguese	51.85 ± 0.98	82.38 ± 0.91	<u>91.94</u> ± 0.72	75.39 ± 0.76	36.28 ± 0.37	71.87 ± 0.18	83.78 ± 0.17	63.98 ± 0.15
Spanish	50.84 ± 2.17	80.80 ± 1.59	90.78 ± 0.23	74.14 ± 0.25	36.02 ± 0.89	70.63 ± 0.48	83.73 ± 0.25	63.46 ± 0.45
French	53.13 ± 0.23	<u>82.64</u> ± 1.24	91.60 ± 1.13	<u>75.79</u> ± 0.73	<u>36.75</u> ± 0.46	71.26 ± 0.40	84.04 ± 0.09	64.01 ± 0.23
Russian	48.11 ± 0.62	79.64 ± 1.01	90.06 ± 1.47	72.60 ± 1.03	34.11 ± 0.69	69.22 ± 0.48	82.36 ± 0.37	61.90 ± 0.40
Arabic	51.03 ± 0.28	81.55 ± 0.70	91.04 ± 1.36	74.54 ± 0.57	34.84 ± 0.23	70.24 ± 0.16	83.19 ± 0.57	62.76 ± 0.31
Hindi	48.63 ± 0.68	80.50 ± 0.17	90.66 ± 0.85	73.27 ± 0.20	35.62 ± 0.21	71.46 ± 0.24	<u>84.87</u> ± 0.13	63.98 ± 0.08
German	49.68 ± 0.45	80.99 ± 0.70	91.53 ± 0.34	74.07 ± 0.19	36.54 ± 1.06	<u>72.11</u> ± 0.18	84.77 ± 0.36	<u>64.47</u> ± 0.41
Chinese(zh)	49.68 ± 1.50	80.47 ± 0.75	89.65 ± 0.23	73.26 ± 0.75	34.43 ± 0.20	70.99 ± 0.42	83.76 ± 0.21	63.06 ± 0.15
Swahili	46.64 ± 1.72	78.67 ± 0.36	90.06 ± 1.14	71.79 ± 1.00	33.76 ± 0.29	68.93 ± 0.72	81.86 ± 0.43	61.52 ± 0.48
Japanese	49.83 ± 2.14	80.46 ± 0.28	89.58 ± 0.40	73.29 ± 0.69	34.98 ± 0.50	71.16 ± 0.62	84.45 ± 0.23	63.53 ± 0.37
Turkish	50.99 ± 0.86	81.55 ± 0.34	91.34 ± 0.41	74.63 ± 0.50	35.52 ± 0.65	71.33 ± 0.15	84.29 ± 0.27	63.71 ± 0.21
ClothoV2								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	19.40 ± 0.62	42.39 ± 1.07	55.95 ± 1.46	39.25 ± 0.87	15.51 ± 0.29	37.86 ± 0.27	50.46 ± 0.16	34.61 ± 0.03
Portuguese	18.60 ± 0.48	39.94 ± 0.87	53.56 ± 0.78	37.36 ± 0.43	14.72 ± 0.43	36.35 ± 0.08	49.10 ± 0.22	33.39 ± 0.22
Spanish	18.85 ± 0.54	40.38 ± 1.27	53.46 ± 1.34	<u>37.56</u> ± 1.00	<u>14.84</u> ± 0.41	<u>36.42</u> ± 0.30	<u>49.26</u> ± 0.35	<u>33.51</u> ± 0.14
French	18.47 ± 0.34	39.42 ± 0.83	51.96 ± 0.38	36.62 ± 0.42	14.55 ± 0.02	36.06 ± 0.87	48.80 ± 0.57	33.13 ± 0.46
Russian	16.37 ± 0.33	38.02 ± 0.38	51.64 ± 1.65	35.34 ± 0.61	13.00 ± 0.26	33.43 ± 0.45	45.56 ± 0.80	30.67 ± 0.46
Arabic	17.77 ± 0.48	39.93 ± 1.96	53.21 ± 0.58	36.97 ± 0.69	13.61 ± 0.34	34.72 ± 0.17	47.04 ± 0.34	31.79 ± 0.28
Hindi	17.80 ± 1.16	40.51 ± 1.18	<u>53.87</u> ± 0.88	37.40 ± 0.57	14.51 ± 0.35	35.58 ± 0.15	48.62 ± 0.72	32.91 ± 0.27
German	18.12 ± 0.24	<u>40.73</u> ± 0.92	52.67 ± 1.16	37.17 ± 0.44	14.70 ± 0.55	35.65 ± 0.42	48.08 ± 0.40	32.81 ± 0.42
Chinese(zh)	16.97 ± 0.65	39.20 ± 0.86	51.74 ± 0.36	35.97 ± 0.46	13.99 ± 0.25	35.04 ± 0.06	47.34 ± 0.08	32.12 ± 0.09
Swahili	17.22 ± 0.42	39.17 ± 0.56	52.12 ± 2.46	36.17 ± 1.01	13.15 ± 0.27	33.58 ± 0.44	45.54 ± 0.34	30.76 ± 0.18
Japanese	<u>19.23</u> ± 0.35	40.10 ± 1.08	52.18 ± 1.75	37.17 ± 0.96	14.25 ± 0.18	35.45 ± 0.40	48.19 ± 0.22	32.63 ± 0.16
Turkish	18.53 ± 1.46	39.55 ± 0.43	51.20 ± 0.39	36.43 ± 0.42	14.14 ± 0.27	34.77 ± 0.31	46.77 ± 0.29	31.89 ± 0.10

Table E.6: Language-wise retrieval performance for the LE-Mixture 40% strategy. AudioCaps and ClothoV2 are shown in separate blocks. A2T and T2A denote Audio-to-Text and Text-to-Audio retrieval, respectively. Values are reported as Recall@k mean ± standard deviation across three runs. The best value for each metric within each dataset block is shown in bold, while the second-best value is underlined.

LE-Mixture 50%								
AudioCaps								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	52.68 ± 0.85	82.90 ± 0.69	92.27 ± 0.23	75.96 ± 0.38	38.88 ± 0.67	74.52 ± 0.34	86.84 ± 0.28	66.75 ± 0.38
Portuguese	50.25 ± 0.88	<u>81.82</u> ± 1.69	91.45 ± 0.70	<u>74.50</u> ± 1.02	36.14 ± 0.28	71.40 ± 0.09	84.31 ± 0.25	63.95 ± 0.12
Spanish	49.79 ± 0.75	79.42 ± 1.18	90.47 ± 0.55	73.23 ± 0.47	35.72 ± 0.18	70.80 ± 0.32	83.82 ± 0.35	63.44 ± 0.21
French	50.17 ± 1.18	81.21 ± 0.79	91.34 ± 0.45	74.24 ± 0.66	<u>36.39</u> ± 0.62	71.13 ± 0.26	84.57 ± 0.18	64.03 ± 0.34
Russian	47.39 ± 2.82	79.60 ± 0.85	90.32 ± 1.27	72.44 ± 1.52	33.96 ± 0.42	69.57 ± 0.46	83.10 ± 0.29	62.20 ± 0.29
Arabic	50.47 ± 1.34	80.61 ± 1.24	90.37 ± 0.81	73.82 ± 1.13	35.33 ± 0.36	70.34 ± 0.32	83.68 ± 0.19	63.11 ± 0.16
Hindi	49.04 ± 1.14	80.46 ± 1.26	90.02 ± 1.08	73.18 ± 0.40	35.58 ± 0.48	71.51 ± 0.32	84.81 ± 0.26	63.96 ± 0.29
German	50.10 ± 1.61	81.21 ± 0.81	<u>91.49</u> ± 0.26	74.27 ± 0.56	36.28 ± 0.61	<u>72.08</u> ± 0.21	<u>85.37</u> ± 0.53	<u>64.58</u> ± 0.38
Chinese(zh)	50.21 ± 0.35	80.28 ± 0.86	90.25 ± 0.51	73.58 ± 0.32	34.41 ± 0.40	70.61 ± 0.94	83.96 ± 0.24	62.99 ± 0.37
Swahili	48.48 ± 2.03	79.38 ± 0.81	89.16 ± 0.77	72.34 ± 1.09	33.71 ± 0.18	68.61 ± 0.38	82.66 ± 0.20	61.66 ± 0.11
Japanese	48.97 ± 1.24	79.75 ± 0.98	90.03 ± 0.56	72.92 ± 0.75	35.07 ± 0.67	70.74 ± 0.11	84.30 ± 0.34	63.37 ± 0.20
Turkish	<u>50.65</u> ± 0.96	81.44 ± 0.81	91.07 ± 0.06	74.39 ± 0.17	35.07 ± 0.69	71.15 ± 0.07	84.72 ± 0.29	63.64 ± 0.34
ClothoV2								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	19.30 ± 0.64	44.40 ± 1.04	56.94 ± 0.25	40.21 ± 0.52	16.33 ± 0.51	38.68 ± 0.72	50.74 ± 0.38	35.25 ± 0.37
Portuguese	18.92 ± 1.01	41.37 ± 0.59	54.90 ± 1.20	38.39 ± 0.93	14.72 ± 0.18	36.43 ± 0.60	49.33 ± 0.42	33.49 ± 0.31
Spanish	<u>19.07</u> ± 0.65	40.83 ± 0.86	<u>54.96</u> ± 1.21	38.29 ± 0.52	14.87 ± 0.42	<u>36.63</u> ± 0.16	<u>49.49</u> ± 0.23	<u>33.66</u> ± 0.22
French	18.28 ± 1.16	39.78 ± 1.26	52.73 ± 0.48	36.93 ± 0.51	14.34 ± 0.22	36.48 ± 0.51	48.97 ± 0.36	33.26 ± 0.14
Russian	16.84 ± 0.42	38.79 ± 0.36	51.29 ± 0.66	35.64 ± 0.15	13.04 ± 0.20	33.86 ± 0.64	46.32 ± 0.40	31.07 ± 0.40
Arabic	17.42 ± 1.92	39.71 ± 0.92	52.66 ± 1.14	36.60 ± 1.33	13.92 ± 0.57	35.02 ± 0.90	47.74 ± 0.55	32.23 ± 0.65
Hindi	19.04 ± 0.66	<u>42.07</u> ± 0.73	54.38 ± 0.24	<u>38.50</u> ± 0.32	<u>14.92</u> ± 0.32	36.38 ± 0.34	48.57 ± 0.42	33.29 ± 0.35
German	18.05 ± 0.29	40.73 ± 0.65	53.30 ± 0.82	37.36 ± 0.30	14.41 ± 0.42	35.91 ± 0.46	48.43 ± 0.41	32.91 ± 0.15
Chinese(zh)	17.42 ± 0.97	38.85 ± 1.27	51.67 ± 1.34	35.98 ± 0.88	13.99 ± 0.57	34.80 ± 0.06	47.28 ± 0.06	32.02 ± 0.18
Swahili	16.59 ± 0.49	40.93 ± 1.96	53.56 ± 1.52	37.02 ± 1.31	13.15 ± 0.17	33.85 ± 0.21	46.09 ± 0.41	31.03 ± 0.18
Japanese	18.56 ± 1.16	40.32 ± 0.71	52.89 ± 0.53	37.26 ± 0.61	14.49 ± 0.33	35.90 ± 0.17	48.64 ± 0.29	33.01 ± 0.21
Turkish	18.98 ± 0.55	41.24 ± 0.94	52.76 ± 0.72	37.66 ± 0.29	14.46 ± 0.36	35.60 ± 0.46	47.62 ± 0.07	32.56 ± 0.26

Table E.7: Language-wise retrieval performance for the LE-Mixture 50% strategy. AudioCaps and ClothoV2 are shown in separate blocks. A2T and T2A denote Audio-to-Text and Text-to-Audio retrieval, respectively. Values are reported as Recall@k mean ± standard deviation across three runs. The best value for each metric within each dataset block is shown in bold, while the second-best value is underlined.

LE-Mixture 100%								
AudioCaps								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	54.63 ± 1.42	84.03 ± 0.63	93.14 ± 0.69	77.27 ± 0.88	39.96 ± 0.24	75.18 ± 0.05	87.17 ± 0.21	67.44 ± 0.08
Portuguese	51.07 ± 2.14	82.45 ± 1.07	91.90 ± 0.41	75.14 ± 0.58	36.74 ± 0.14	72.54 ± 0.19	85.26 ± 0.24	64.85 ± 0.16
Spanish	50.54 ± 1.08	82.38 ± 0.77	92.13 ± 0.23	75.02 ± 0.05	36.85 ± 0.37	72.02 ± 0.24	84.56 ± 0.20	64.48 ± 0.10
French	50.32 ± 1.07	82.82 ± 1.08	92.35 ± 1.22	75.17 ± 0.59	36.77 ± 0.31	72.63 ± 0.38	84.99 ± 0.32	64.80 ± 0.32
Russian	50.02 ± 1.90	81.66 ± 0.79	91.41 ± 0.43	74.37 ± 0.75	35.68 ± 0.39	71.18 ± 0.08	83.92 ± 0.39	63.59 ± 0.15
Arabic	50.77 ± 1.43	82.71 ± 1.42	91.98 ± 0.62	75.15 ± 1.04	36.60 ± 0.50	72.00 ± 0.33	84.42 ± 0.13	64.34 ± 0.27
Hindi	50.47 ± 1.31	81.21 ± 0.20	91.53 ± 0.17	74.41 ± 0.38	36.27 ± 0.47	72.30 ± 0.41	85.08 ± 0.31	64.55 ± 0.35
German	50.28 ± 1.11	<u>83.01</u> ± 1.17	<u>92.58</u> ± 0.81	75.29 ± 0.55	<u>37.29</u> ± 0.10	<u>72.82</u> ± 0.56	<u>85.78</u> ± 0.15	<u>65.29</u> ± 0.15
Chinese(zh)	49.34 ± 1.24	81.10 ± 1.07	90.70 ± 0.91	73.72 ± 0.43	36.06 ± 0.58	71.72 ± 0.35	84.67 ± 0.19	64.15 ± 0.26
Swahili	48.22 ± 0.64	80.31 ± 0.00	90.81 ± 0.46	73.12 ± 0.23	34.95 ± 0.34	69.89 ± 0.30	83.04 ± 0.31	62.62 ± 0.32
Japanese	48.97 ± 2.04	81.44 ± 0.70	91.08 ± 0.69	73.83 ± 0.75	36.21 ± 0.32	71.96 ± 0.26	84.92 ± 0.27	64.36 ± 0.14
Turkish	<u>52.01</u> ± 2.04	82.60 ± 1.61	91.79 ± 0.58	<u>75.47</u> ± 1.01	37.21 ± 0.44	72.47 ± 0.63	85.26 ± 0.28	64.98 ± 0.43
ClothoV2								
Language	Audio-to-Text				Text-to-Audio			
	R@1	R@5	R@10	R@Avg	R@1	R@5	R@10	R@Avg
English	20.09 ± 0.53	43.67 ± 0.43	56.65 ± 0.69	40.14 ± 0.24	16.21 ± 0.29	38.42 ± 0.17	50.88 ± 0.57	35.17 ± 0.22
Portuguese	18.88 ± 0.14	41.28 ± 0.77	54.32 ± 1.53	38.16 ± 0.73	<u>14.93</u> ± 0.76	36.65 ± 0.35	49.77 ± 0.41	33.79 ± 0.35
Spanish	18.50 ± 0.71	41.56 ± 1.25	<u>54.51</u> ± 0.81	38.19 ± 0.24	14.72 ± 0.34	<u>36.90</u> ± 0.63	49.81 ± 0.41	<u>33.81</u> ± 0.35
French	17.67 ± 0.77	40.54 ± 1.46	54.16 ± 1.18	37.46 ± 1.14	14.77 ± 0.45	36.60 ± 0.54	<u>49.84</u> ± 0.46	33.74 ± 0.40
Russian	18.28 ± 1.10	39.39 ± 1.78	52.51 ± 1.86	36.72 ± 1.45	14.15 ± 0.38	35.18 ± 0.83	48.17 ± 0.54	32.50 ± 0.55
Arabic	18.44 ± 0.54	40.45 ± 0.30	53.68 ± 0.63	37.52 ± 0.37	14.35 ± 0.27	35.47 ± 0.57	48.19 ± 0.71	32.66 ± 0.49
Hindi	19.04 ± 1.25	41.02 ± 0.28	54.45 ± 0.10	38.17 ± 0.49	14.85 ± 0.37	36.09 ± 0.12	48.56 ± 0.54	33.17 ± 0.25
German	17.99 ± 0.91	<u>42.39</u> ± 0.29	54.23 ± 0.20	<u>38.20</u> ± 0.32	14.74 ± 0.30	36.81 ± 0.60	49.28 ± 0.82	33.61 ± 0.37
Chinese(zh)	18.34 ± 0.64	39.81 ± 0.16	52.44 ± 0.42	36.86 ± 0.37	14.35 ± 0.05	35.95 ± 0.54	48.27 ± 0.41	32.86 ± 0.30
Swahili	17.32 ± 0.76	39.27 ± 0.68	52.44 ± 1.52	36.34 ± 0.96	13.85 ± 0.38	34.33 ± 0.26	47.14 ± 0.26	31.78 ± 0.12
Japanese	<u>19.75</u> ± 0.40	40.64 ± 1.21	53.33 ± 1.00	37.90 ± 0.85	14.68 ± 0.53	36.07 ± 0.35	48.99 ± 0.33	33.25 ± 0.19
Turkish	18.69 ± 0.83	40.57 ± 0.42	54.39 ± 0.78	37.88 ± 0.32	14.82 ± 0.38	35.73 ± 0.70	48.20 ± 0.57	32.92 ± 0.43

Table E.8: Language-wise retrieval performance for the LE-Mixture 100% strategy. AudioCaps and ClothoV2 are shown in separate blocks. A2T and T2A denote Audio-to-Text and Text-to-Audio retrieval, respectively. Values are reported as Recall@k mean ± standard deviation across three runs. The best value for each metric within each dataset block is shown in bold, while the second-best value is underlined.

E.2 Classification

Language	ESC-50	UrbanSound8K
English	83.50 ± 2.85	71.44 ± 5.10
Portuguese	12.47 ± 1.62	19.59 ± 3.81
Spanish	9.65 ± 1.77	<u>26.37</u> ± 3.37
French	<u>16.18</u> ± 2.29	22.69 ± 5.41
Russian	2.02 ± 1.13	9.03 ± 4.31
Arabic	1.53 ± 0.58	6.44 ± 3.61
Hindi	1.33 ± 1.00	5.56 ± 2.22
German	11.12 ± 1.23	21.08 ± 8.94
Chinese(zh)	1.08 ± 1.29	13.07 ± 7.68
Swahili	8.42 ± 1.79	24.41 ± 5.45
Japanese	1.23 ± 0.75	11.49 ± 0.44
Turkish	7.35 ± 0.83	19.20 ± 4.36

Table E.9: Zero-shot classification performance for English-only across evaluation languages. Values are reported as mean $\pm$ standard deviation across three runs. **Bold** values indicate the best language for each dataset, while underlined values indicate the second-best language.

Language	ESC-50	UrbanSound8K
English	90.12 ± 1.35	74.92 ± 4.60
Portuguese	<u>89.85</u> ± 0.94	71.87 ± 5.95
Spanish	89.15 ± 1.51	74.61 ± 6.29
French	86.70 ± 1.11	76.01 ± 6.05
Russian	80.88 ± 2.53	79.30 ± 1.45
Arabic	85.28 ± 1.80	72.16 ± 3.35
Hindi	83.03 ± 0.55	69.38 ± 3.33
German	88.23 ± 1.19	<u>79.23</u> ± 2.27
Chinese(zh)	86.65 ± 1.53	67.98 ± 2.58
Swahili	85.17 ± 1.98	75.37 ± 4.13
Japanese	88.22 ± 0.52	78.34 ± 5.21
Turkish	86.87 ± 1.55	66.45 ± 2.45

Table E.10: Zero-shot classification performance for Random Sampling across evaluation languages. Values are reported as mean $\pm$ standard deviation across three runs. **Bold** values indicate the best language for each dataset, while underlined values indicate the second-best language.

Language	ESC-50	UrbanSound8K
English	81.43 ± 2.39	73.90 ± 4.09
Portuguese	<u>77.97</u> ± 0.65	66.82 ± 1.38
Spanish	69.45 ± 0.74	63.01 ± 6.68
French	74.02 ± 2.47	71.94 ± 4.47
Russian	57.58 ± 4.65	60.09 ± 6.75
Arabic	62.70 ± 0.33	68.09 ± 2.02
Hindi	65.00 ± 4.31	61.52 ± 4.68
German	71.30 ± 0.87	<u>74.33</u> ± 6.32
Chinese(zh)	68.68 ± 1.24	62.37 ± 6.41
Swahili	75.43 ± 1.06	77.00 ± 3.27
Japanese	72.40 ± 0.82	68.64 ± 0.52
Turkish	63.60 ± 3.36	57.60 ± 4.41

Table E.11: Zero-shot classification performance for LE-Mixture 10% across evaluation languages. Values are reported as mean $\pm$ standard deviation across three runs. **Bold** values indicate the best language for each dataset, while underlined values indicate the second-best language.

Language	ESC-50	UrbanSound8K
English	86.53 ± 2.39	73.29 ± 1.90
Portuguese	<u>83.37</u> ± 2.97	69.94 ± 2.73
Spanish	80.87 ± 5.25	73.25 ± 2.16
French	80.08 ± 1.01	71.37 ± 4.02
Russian	68.67 ± 1.59	68.92 ± 4.08
Arabic	73.35 ± 3.04	69.03 ± 2.93
Hindi	71.32 ± 2.14	66.75 ± 2.84
German	79.37 ± 3.96	<u>77.07</u> ± 4.49
Chinese(zh)	77.83 ± 0.98	65.51 ± 3.15
Swahili	82.00 ± 0.68	77.90 ± 0.46
Japanese	79.78 ± 2.15	69.45 ± 2.72
Turkish	72.58 ± 5.31	63.08 ± 4.08

Table E.12: Zero-shot classification performance for LE-Mixture 20% across evaluation languages. Values are reported as mean $\pm$ standard deviation across three runs. **Bold** values indicate the best language for each dataset, while underlined values indicate the second-best language.

Language	ESC-50	UrbanSound8K
English	86.27 ± 3.34	74.12 ± 2.36
Portuguese	<u>86.15</u> ± 1.17	69.36 ± 5.44
Spanish	83.93 ± 2.84	68.78 ± 3.15
French	83.67 ± 2.30	75.54 ± 4.36
Russian	72.03 ± 2.86	71.92 ± 2.67
Arabic	76.85 ± 1.53	71.18 ± 5.35
Hindi	78.27 ± 2.20	68.17 ± 4.35
German	84.37 ± 2.40	<u>77.19</u> ± 2.69
Chinese(zh)	81.47 ± 0.29	65.17 ± 3.40
Swahili	81.37 ± 0.48	77.53 ± 3.89
Japanese	83.73 ± 0.95	74.40 ± 6.08
Turkish	79.10 ± 2.03	63.89 ± 3.78

Table E.13: Zero-shot classification performance for LE-Mixture 30% across evaluation languages. Values are reported as mean $\pm$ standard deviation across three runs. **Bold** values indicate the best language for each dataset, while underlined values indicate the second-best language.

Language	ESC-50	UrbanSound8K
English	84.57 ± 1.83	72.30 ± 2.22
Portuguese	86.20 ± 1.89	64.00 ± 8.25
Spanish	84.00 ± 2.56	71.89 ± 1.56
French	84.50 ± 0.46	74.47 ± 2.58
Russian	73.73 ± 2.36	74.06 ± 2.02
Arabic	77.12 ± 2.15	70.37 ± 5.02
Hindi	76.88 ± 1.49	64.00 ± 2.83
German	<u>86.12</u> ± 1.59	76.68 ± 2.62
Chinese(zh)	82.68 ± 0.60	61.46 ± 3.88
Swahili	81.33 ± 3.31	<u>75.46</u> ± 4.49
Japanese	83.67 ± 1.74	73.37 ± 0.65
Turkish	77.63 ± 0.94	55.00 ± 8.94

Table E.14: Zero-shot classification performance for LE-Mixture 40% across evaluation languages. Values are reported as mean $\pm$ standard deviation across three runs. **Bold** values indicate the best language for each dataset, while underlined values indicate the second-best language.

Language	ESC-50	UrbanSound8K
English	84.05 $\pm$ 1.01	70.69 $\pm$ 1.53
Portuguese	86.72 $\pm$ 1.09	71.60 $\pm$ 3.89
Spanish	84.45 $\pm$ 1.70	76.34 $\pm$ 1.45
French	84.23 $\pm$ 1.53	<u>78.02</u> $\pm$ 2.88
Russian	77.57 $\pm$ 1.76	73.85 $\pm$ 3.11
Arabic	77.02 $\pm$ 1.21	68.82 $\pm$ 3.09
Hindi	79.62 $\pm$ 1.01	66.09 $\pm$ 2.63
German	84.98 $\pm$ 1.55	77.60 $\pm$ 2.73
Chinese(zh)	84.08 $\pm$ 1.35	62.05 $\pm$ 2.70
Swahili	81.53 $\pm$ 1.69	78.11 $\pm$ 1.30
Japanese	<u>85.17</u> $\pm$ 2.22	72.33 $\pm$ 4.43
Turkish	79.85 $\pm$ 3.59	62.79 $\pm$ 3.42

Table E.15: Zero-shot classification performance for LE-Mixture 50% across evaluation languages. Values are reported as mean $\pm$ standard deviation across three runs. **Bold** values indicate the best language for each dataset, while underlined values indicate the second-best language.

Language	ESC-50	UrbanSound8K
English	82.53 $\pm$ 2.36	72.68 $\pm$ 5.53
Portuguese	86.27 $\pm$ 2.10	71.16 $\pm$ 3.74
Spanish	84.38 $\pm$ 2.45	66.26 $\pm$ 2.97
French	81.42 $\pm$ 0.63	<u>75.81</u> $\pm$ 2.55
Russian	77.47 $\pm$ 0.78	76.71 $\pm$ 2.35
Arabic	76.43 $\pm$ 1.13	70.29 $\pm$ 0.50
Hindi	75.22 $\pm$ 1.66	64.67 $\pm$ 3.11
German	<u>85.93</u> $\pm$ 0.85	72.53 $\pm$ 2.20
Chinese(zh)	81.45 $\pm$ 3.03	61.78 $\pm$ 0.72
Swahili	79.47 $\pm$ 0.98	74.88 $\pm$ 0.33
Japanese	84.42 $\pm$ 1.65	74.34 $\pm$ 4.89
Turkish	77.40 $\pm$ 5.08	62.01 $\pm$ 1.48

Table E.16: Zero-shot classification performance for LE-Mixture 100% across evaluation languages. Values are reported as mean $\pm$ standard deviation across three runs. **Bold** values indicate the best language for each dataset, while underlined values indicate the second-best language.

Mixture-of-Layers Implementation Details

This chapter gives implementation details about the different variations of Mixture-of-Layers (MoL). Considering a transformer backbone that exposes L stacked hidden states (the embedding output plus every encoder block), giving $\mathbf{H} \in \mathbb{R}^{B \times L \times T \times F}$. A *layer-aggregation* module collapses the layer axis into a single representation $\mathbf{Z} \in \mathbb{R}^{B \times T \times F}$, which is then masked-mean pooled over tokens and projected. The four strategies differ only in *how* the per-layer mixing weights are produced: fixed and uniform (MEAN), learned but input-independent (WEIGHTED-SUM), or *input-conditioned* through a gating network (MOL, with two refiner/pooling choices).

Symbol	Meaning
B, L, T, F	Batch Size, Stacked Layers, Sequence Length, Feature Dim
$\mathbf{H} \in \mathbb{R}^{B \times L \times T \times F}$	Stacked per-Layer hidden states
$\mathbf{H}^{(l)} := \mathbf{H}_{:,l,:,:} \in \mathbb{R}^{B \times T \times F}$	Hidden States of Layer l
$\mathbf{Z} \in \mathbb{R}^{B \times T \times F}$	Aggregated Representation
$\text{softmax}_l, \text{mean}_T, \text{mean}_L$	Softmax / Mean over the Layer or Token axis
$\odot$	Element-wise Product

Algorithm 2 MEAN Aggregation

Input: $\mathbf{H} \in \mathbb{R}^{B \times L \times T \times F}$

Output: $\mathbf{Z} \in \mathbb{R}^{B \times T \times F}$

```

1:  $\mathbf{Z} \leftarrow \frac{1}{L} \sum_{l=1}^L \mathbf{H}^{(l)}$ 
2: return  $\mathbf{Z}$ 

```

▷ uniform, parameter-free

Algorithm 3 WEIGHTED-SUM

Input: $\mathbf{H} \in \mathbb{R}^{B \times L \times T \times F}$

Learnable: layer logits $\mathbf{w} = (w_1, \dots, w_L) \in \mathbb{R}^L$, init. $\mathbf{0}$

Output: $\mathbf{Z} \in \mathbb{R}^{B \times T \times F}$

- 1: $\boldsymbol{\alpha} \leftarrow \text{softmax}_l(\mathbf{w})$ $\triangleright \boldsymbol{\alpha} \in \mathbb{R}^L$; init. $\mathbf{0} \Rightarrow$ uniform mean
 - 2: $\mathbf{Z} \leftarrow \sum_{l=1}^L \alpha_l \mathbf{H}^{(l)}$ $\triangleright$ weights shared across batch & tokens
 - 3: **return** $\mathbf{Z}$
-

Algorithm 4 MoL Gating: Self-Attention Refiner + Mean Pooling

Input: $\mathbf{H} \in \mathbb{R}^{B \times L \times T \times F}$

Modules: MHSA over layers, gate MLP $\text{GateMLP} : \mathbb{R}^F \rightarrow \mathbb{R}^L$

Output: $\mathbf{Z} \in \mathbb{R}^{B \times T \times F}$

- // 1. token pooling: one descriptor per layer*
 - 1: $\mathbf{P} \leftarrow \text{mean}_T(\mathbf{H})$ $\triangleright \mathbf{P} \in \mathbb{R}^{B \times L \times F}$
 - // 2. layer refinement: attention across the L layers*
 - 2: $\mathbf{R} \leftarrow \text{MHSA}(\mathbf{P}, \mathbf{P}, \mathbf{P})$ $\triangleright$ self-attn $\Rightarrow \mathbb{R}^{B \times L \times F}$
 - // 3. layer pooling (mean) $\rightarrow$ gate input*
 - 3: $\mathbf{g} \leftarrow \text{mean}_L(\mathbf{R})$ $\triangleright \mathbf{g} \in \mathbb{R}^{B \times F}$
 - // 4. per-sample gating distribution over layers*
 - 4: $\mathbf{s} \leftarrow \text{GateMLP}(\mathbf{g})$ $\triangleright \mathbf{s} \in \mathbb{R}^{B \times L}$ logits
 - 5: $\boldsymbol{\pi} \leftarrow \text{softmax}_l(\mathbf{s})$ $\triangleright \boldsymbol{\pi} \in \mathbb{R}^{B \times L}$
 - // 5. input-conditioned weighted combination*
 - 6: $\mathbf{Z} \leftarrow \sum_{l=1}^L \pi_{:,l} \mathbf{H}^{(l)}$ $\triangleright \pi_{:,l}$ broadcast over T, F
 - 7: **return** $\mathbf{Z}$
-

Algorithm 5 MoL Gating: Self-Attention refiner + Attentive Pooling

Input: $\mathbf{H} \in \mathbb{R}^{B \times L \times T \times F}$

Modules: MHSA; $\mathbf{W}_q, \mathbf{W}_k \in \mathbb{R}^{F \times D_a}$, $\mathbf{w}_s \in \mathbb{R}^{D_a}$, gate MLP $\text{GateMLP} : \mathbb{R}^F \rightarrow \mathbb{R}^L$

Output: $\mathbf{Z} \in \mathbb{R}^{B \times T \times F}$

```

    // 1. token pooling: one descriptor per layer
1:  $\mathbf{P} \leftarrow \text{mean}_T(\mathbf{H})$   $\triangleright \mathbf{P} \in \mathbb{R}^{B \times L \times F}$ 
    // 2. layer refinement: attention across the L layers
2:  $\mathbf{R} \leftarrow \text{MHSA}(\mathbf{P}, \mathbf{P}, \mathbf{P})$   $\triangleright \text{self-attn} \Rightarrow \mathbb{R}^{B \times L \times F}$ 
    // 3. attentive pooling over layers  $\rightarrow$  gate input
3:  $\mathbf{Q} \leftarrow \tanh(\mathbf{R}\mathbf{W}_q)$ ,  $\mathbf{K} \leftarrow \tanh(\mathbf{R}\mathbf{W}_k)$   $\triangleright \in \mathbb{R}^{B \times L \times D_a}$ 
4:  $\mathbf{u} \leftarrow (\mathbf{Q} \odot \mathbf{K}) \mathbf{w}_s$   $\triangleright \text{per-layer score } \mathbf{u} \in \mathbb{R}^{B \times L}$ 
5:  $\boldsymbol{\beta} \leftarrow \text{softmax}_l(\mathbf{u})$   $\triangleright \boldsymbol{\beta} \in \mathbb{R}^{B \times L}$ 
6:  $\mathbf{g} \leftarrow \sum_{l=1}^L \beta_{:,l} \mathbf{R}^{(l)}$   $\triangleright \mathbf{g} \in \mathbb{R}^{B \times F}$ 
    // 4. per-sample gating distribution over layers
7:  $\mathbf{s} \leftarrow \text{GateMLP}(\mathbf{g})$   $\triangleright \mathbf{s} \in \mathbb{R}^{B \times L}$  logits
8:  $\boldsymbol{\pi} \leftarrow \text{softmax}_l(\mathbf{s})$   $\triangleright \boldsymbol{\pi} \in \mathbb{R}^{B \times L}$ 
    // 5. input-conditioned weighted combination
9:  $\mathbf{Z} \leftarrow \sum_{l=1}^L \pi_{:,l} \mathbf{H}^{(l)}$ 
10: return  $\mathbf{Z}$ 

```
