UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

PROGRAMA DE PÓS GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO

WERIKCYANO LIMA GUIMARÃES

Aquisição Progressiva de Habilidades por meio de Curriculum Learning para Futebol de Robôs Multiagente

Goiânia
2025

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

[X] 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

Werikcyano Lima Guimarães

3. Título do trabalho

Aquisição Progressiva de Habilidades por meio de Curriculum Learning para Futebol de Robôs Multiagente

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

Concorda com a liberação total do documento [x] SIM [ ] NÃO[1]

[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(à) autor(a) e ao(à) 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 **Telma Woerle De Lima Soares**, **Professora do Magistério Superior**, em 26/05/2025, às 17:46, 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 **Werikcyano Lima Guimarães**, **Discente**, em 03/06/2025, às 13:35, 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 **5395459** e o código CRC **6AC1D1D5**.

Referência: Processo nº 23070.007884/2025-61 SEI nº 5395459

WERIKCYANO LIMA GUIMARÃES

Aquisição Progressiva de Habilidades por meio de Curriculum Learning para Futebol de Robôs Multiagente

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

Área de concentração: Ciência da Computação, Linha de Pesquisa: Sistemas Inteligentes e Aplicações.

Orientadora: Profa. Dra. Telma Woerle de Lima Soares

Goiânia
2025

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

Lima Guimarães, Werikcyano
Aquisição Progressiva de Habilidades por meio de Curriculum Learning para Futebol de Robôs Multiagente [manuscrito] / Werikcyano Lima Guimarães. - 2025.
LXXX, 80 f.: il.

Orientador: Prof. Telma Woerle de Lima Soares.
Dissertação (Mestrado) - Universidade Federal de Goiás, Instituto de Informática (INF), Programa de Pós-Graduação em Ciência da Computação, Goiânia, 2025.
Bibliografia.
Inclui gráfico, tabelas, lista de figuras, lista de tabelas.

1. Aprendizado por Reforço. 2. Curriculum Learning. 3. Futebol de Robôs. 4. Multiagente. 5. Treinamento Progressivo. I. Woerle de Lima Soares, Telma, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº **04/2025** da sessão de Defesa de Dissertação de **Werikcyano Lima Guimarães**, que confere o título de Mestre em Ciência da Computação, na área de concentração em Ciência da Computação.

Aos onze dias do mês de março de dois mil e vinte e cinco, a partir das catorze horas e trinta minutos, via webconferência, realizou-se a sessão pública de Defesa de Dissertação intitulada **"Aquisição Progressiva de Habilidades por meio de Curriculum Learning para Futebol de Robôs Multiagente"**. Os trabalhos foram instalados pela Orientadora, Professora Doutora Telma Woerle de Lima Soares (INF/UFG) com a participação dos demais membros da Banca Examinadora: Professor Doutor Aldo André Diaz Salazar (INF/UFG), membro titular externo; Professor Doutor Marcos Ricardo Omena de Albuquerque Máximo (ITA), membro titular externo. A realização 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 Telma Woerle de Lima Soares, 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 onze dias do mês de março de dois mil e vinte e cinco.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Telma Woerle De Lima Soares**, **Professora do Magistério Superior**, em 11/03/2025, às 17:03, 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 **Aldo Andre Diaz Salazar**, **Professor do Magistério Superior**, em 11/03/2025, às 17:04, 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 **Marcos Ricardo Omena de Albuquerque Maximo**, **Usuário Externo**, em 11/03/2025, às 17:04, 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 **Werikcyano Lima Guimarães**, **Discente**, em 12/03/2025, às 12:06, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

Referência: Processo nº 23070.007884/2025-61 SEI nº 5213788

Werikcyano Lima Guimarães

Graduado em Engenharia de Computação pela Escola de Engenharia Elétrica, Mecânica e Computação - EMC na Universidade Federal de Goiás - UFG. Tem experiência nas áreas de inteligência artificial com foco em aprendizado por reforço e em prospecção de dados. Atualmente é pesquisador no Centro de Excelência em Inteligência Artificial em Goiás, desenvolvendo projetos naquelas áreas.

Agradecimentos

Primeiramente, agradeço a Deus por me conceder força, sabedoria e perseverança durante toda esta jornada acadêmica.

À minha orientadora, Profa. Dra. Telma Woerle de Lima Soares, por sua dedicação, paciência e orientação inestimável. Obrigado por me aturar nos momentos difíceis e por acreditar no meu potencial quando nem eu mesmo acreditava.

À minha mãe, meu alicerce e exemplo de vida, por todo amor incondicional e apoio constante em cada etapa do meu percurso.

À minha irmã, pela amizade, cumplicidade e por sempre torcer pelo meu sucesso.

À minha noiva e toda a sua família, pelo companheirismo, compreensão nos momentos de ausência e por ser minha motivação para seguir em frente.

Aos meus amigos, que tornaram esta jornada mais leve e prazerosa. Obrigado por cada palavra de incentivo, pelos momentos de descontração e por toda ajuda que me ofereceram.

À minha família, de forma geral, pelo apoio e por compreenderem minha ausência em diversos momentos importantes.

À equipe SSL-EL do núcleo de robótica Pequi Mecânico, por compartilharem conhecimentos, desafios e conquistas que contribuíram significativamente para meu crescimento acadêmico e profissional.

A todos que, direta ou indiretamente, fizeram parte da minha formação, meu sincero agradecimento.

Resumo

Guimarães Lima, Werikcyano. **Aquisição Progressiva de Habilidades por meio de Curriculum Learning para Futebol de Robôs Multiagente**. Goiânia, 2025. 83p. Dissertação de Mestrado. Programa de Pós Graduação em Ciência da Computação, Instituto de Informática, Universidade Federal de Goiás.

Este trabalho investiga a integração de Curriculum Learning com Self-play para aprendizado por reforço no contexto do futebol de robôs da categoria SSL-EL. A pesquisa aborda o desafio do desenvolvimento de políticas eficientes em ambientes complexos multiagente, propondo uma metodologia estruturada que decompõe o aprendizado em estágios progressivos. O framework implementado estabelece critérios adaptativos de transição entre tarefas, permitindo que os agentes desenvolvam inicialmente habilidades fundamentais antes de enfrentarem cenários competitivos completos. Os resultados experimentais demonstram claramente a superioridade da abordagem combinada, com taxa de vitória significativamente maior em torneios competitivos quando comparada ao Full Self-play tradicional, além de expressivo aumento na média de gols por partida. Adicionalmente, observou-se redução substancial no tempo total de treinamento e maior estabilidade no processo de aprendizado, evidenciada por métricas como entropia da política, perda da política e variância explicada. As análises confirmam que o Curriculum Learning proporciona uma base técnica sólida que potencializa os benefícios do Self-play, resultando em agentes com capacidades táticas mais sofisticadas e eficientes.

Palavras–chave

Aprendizado por Reforço, Curriculum Learning, Futebol de Robôs, Multiagente, Sistema de Recompensas, Treinamento Progressivo

Abstract

Guimarães Lima, Werikcyano. . Goiânia, 2025. 83p. MSc. Dissertation. Programa de Pós Graduação em Ciência da Computação, Instituto de Informática, Universidade Federal de Goiás.

This work investigates the integration of Curriculum Learning with Self-play for reinforcement learning in the context of SSL-EL robot soccer. The research addresses the challenge of developing efficient policies in complex multi-agent environments by proposing a structured methodology that decomposes learning into progressive stages. The implemented framework establishes adaptive criteria for transitioning between tasks, allowing agents to initially develop fundamental skills before facing complete competitive scenarios. The experimental results clearly demonstrate the superiority of the combined approach, with significantly higher win rates in competitive tournaments compared to traditional Full Self-play, as well as an expressive increase in the average goals per match. Additionally, a substantial reduction in total training time and greater stability in the learning process were observed, evidenced by metrics such as policy entropy, policy loss, and explained variance. The analyses confirm that Curriculum Learning provides a solid technical foundation that enhances the benefits of Self-play, resulting in agents with more sophisticated and efficient tactical capabilities.

Keywords

Reinforcement Learning, Curriculum Learning, Robot Soccer, Multi-agent, Reward System, Progressive Training

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

O futebol de robôs representa um domínio desafiador para a aplicação de técnicas de Inteligência Artificial, combinando aspectos complexos de percepção, tomada de decisão e controle em tempo real. Neste contexto, o Aprendizado por Reforço (*Reinforcement Learning* - RL) emergiu como uma abordagem promissora, permitindo que agentes desenvolvam comportamentos sofisticados através da interação direta com o ambiente [Sutton e Barto 2018]. No entanto, a complexidade inerente ao domínio do futebol de robôs apresenta desafios significativos para o aprendizado efetivo.

O futebol de robôs surgiu como um desafio-problema na RoboCup, uma iniciativa internacional dedicada à promoção da pesquisa em robótica e inteligência artificial. Dentre as várias categorias da competição, a *Small Size League - Entry Level* (SSL-EL) representa uma versão simplificada da liga SSL tradicional, projetada para facilitar a entrada de novas equipes. Na SSL-EL, cada equipe opera até três robôs autônomos em um campo de 5,5m × 4m, sendo um dos robôs designado como goleiro. Os robôs possuem formato cilíndrico padronizado (0,18m de diâmetro por 0,15m de altura) e são identificados por padrões visuais no topo, que permitem sua detecção por um sistema de visão computacional centralizado. A partida é estruturada em dois tempos de 5 minutos, com regras específicas para faltas, penalidades e situações de jogo. Este ambiente controlado oferece um excelente campo de testes para algoritmos de IA, combinando problemas de percepção (através do sistema de visão), planejamento (estratégias de jogo) e ação (controle dos robôs) em um cenário dinâmico multiagente [30].

As características específicas da categoria SSL-EL introduzem desafios técnicos particulares que influenciam diretamente o desenvolvimento de soluções baseadas em aprendizado de máquina. O tamanho reduzido do campo (comparado à categoria SSL completa) e o número limitado de robôs (três por equipe) simplificam alguns aspectos, mas também impõem restrições específicas. Por exemplo, as regras severas sobre velocidade máxima dos robôs (1,5m/s) e da bola (3m/s), as limitações nas áreas de atuação, e as penalidades acumulativas exigem que os agentes desenvolvam não apenas habilidades técnicas, mas também observem as regras do jogo durante suas ações. Adicionalmente, a necessidade de especialização dos papéis (goleiro e jogadores de linha) impõe uma dimen-

são extra de complexidade na coordenação da equipe. Estes fatores tornam a aplicação de *Curriculum Learning* particularmente relevante, pois permite que os agentes primeiro dominem habilidades básicas dentro das restrições das regras, antes de progredirem para comportamentos táticos e estratégicos mais sofisticados.

Um dos obstáculos no desenvolvimento de agentes para futebol de robôs através de RL é a necessidade de aprender múltiplas habilidades interdependentes simultaneamente [35]. Os agentes precisam dominar desde capacidades básicas, como navegação e controle da bola [Haarnoja et al. 2024], até comportamentos táticos complexos que envolvem coordenação multiagente [Brandão et al. 2022]. Esta multiplicidade de habilidades, combinada com a natureza contínua do espaço de estados e ações, torna o processo de aprendizagem particularmente desafiador [Haarnoja et al. 2024].

Para endereçar estes desafios, este trabalho propõe a aplicação de *Curriculum Learning* [Narvekar et al. 2020] como estratégia para melhorar a eficiência e eficácia do processo de aprendizagem em futebol de robôs. O *Curriculum Learning* permite uma abordagem estruturada ao aprendizado, onde os agentes são expostos a tarefas progressivamente mais complexas, facilitando a aquisição gradual de competências fundamentais. Esta abordagem é especialmente relevante no contexto da categoria *SSL-EL* (*Small Size League - Entry Level*), onde os agentes precisam desenvolver habilidades básicas antes de enfrentar cenários competitivos completos [30].

A motivação principal deste trabalho surge da observação de trabalhos anteriores [Brandão et al. 2022], que demonstrou a viabilidade do uso de Aprendizado por Reforço no contexto de futebol de robôs através do algoritmo *Proximal Policy Optimization* (PPO) em uma abordagem multi-agente com política compartilhada. No entanto, aplicar diretamente métodos de RL ao problema completo do futebol de robôs, sem uma estrutura progressiva de aprendizado, frequentemente resulta em um processo ineficiente e instável. Inspirado pela forma como jogos populares como *FIFA* e *Rocket League* introduzem novos jogadores através de centros de treinamento antes de permitir a competição direta, este trabalho propõe uma abordagem baseada em *Curriculum Learning* para estruturar o processo de aprendizagem em etapas progressivas. Esta estratégia permite que os agentes desenvolvam primeiro habilidades fundamentais antes de enfrentarem cenários mais complexos, similar ao processo natural de desenvolvimento de habilidades em jogadores humanos [Gupta et al. 2019].

O objetivo geral deste trabalho é desenvolver e avaliar uma metodologia baseada em *Curriculum Learning* para melhorar o processo de aprendizagem de agentes em futebol de robôs. Especificamente, busca-se:

1. Desenvolver uma estrutura de *curriculum* que decomponha o aprendizado em estágios progressivos, começando com habilidades fundamentais como chute básico e progredindo até comportamentos mais complexos;

2. Implementar um sistema de transição adaptativo entre níveis do *curriculum*, garantindo que os agentes desenvolvam uma base sólida antes de progredir para tarefas mais desafiadoras;

3. Avaliar comparativamente o desempenho de agentes treinados com e sem *Curriculum Learning*, considerando métricas como eficiência no aprendizado e qualidade final do comportamento aprendido.

Este trabalho está estruturado de forma a apresentar progressivamente os conceitos, metodologias e resultados da pesquisa. No Capítulo 2, são aprofundados os conceitos teóricos essenciais para a compreensão do trabalho, abordando Aprendizado por Reforço, o algoritmo PPO, *Multi-agent RL*, *Self-play*, *Curriculum Learning* e aspectos específicos do futebol de robôs. O Capítulo 3 detalha o ambiente de simulação utilizado, a arquitetura do sistema proposto e a implementação do *Curriculum Learning*, incluindo seus estágios progressivos e mecanismos de transição.

No Capítulo 4, são apresentados os experimentos realizados, incluindo análises comparativas entre diferentes abordagens de treinamento e os resultados dos torneios, evidenciando a eficácia da metodologia proposta. Por fim, o Capítulo 5 sintetiza as principais descobertas e contribuições do trabalho, discutindo suas implicações para o campo do aprendizado por reforço em ambientes multiagentes, apresentando limitações identificadas e sugerindo direções para trabalhos futuros. O trabalho inclui ainda uma subseção dedicada à reprodutibilidade dos experimentos, sendo complementado por gráficos, tabelas e referências a vídeos demonstrativos disponíveis online.

As principais contribuições esperadas deste trabalho incluem uma metodologia estruturada para aplicação de *Curriculum Learning* em futebol de robôs, um *framework* adaptativo para progressão entre níveis de complexidade e evidências empíricas sobre a efetividade do *Curriculum Learning* em melhorar o processo de aprendizagem em ambientes complexos multiagente. Este trabalho utiliza como base a implementação [15] do trabalho *Multiagent Reinforcement Learning for Strategic Decision Making and Control in Robotic Soccer Through Self-Play* [Brandão et al. 2022] realizada pela equipe Pequi Mecânico [Site oficial do Pequi Mecânico 2025], que por sua vez foi desenvolvida utilizando o *framework* RSoccer [Martins et al. 2022] da equipe RobôCIn [Site oficial da RobôCin 2025]. O código fonte completo desta solução está disponível em `https://github.com/Werikcyano/RL-SSL-EL`.

CAPÍTULO 2

Fundamentação Teórica

2.1 Aprendizado por Reforço

O Aprendizado por Reforço (*Reinforcement Learning* - RL) é uma abordagem de aprendizado baseada na interação entre um agente e seu ambiente, onde o objetivo principal é maximizar um sinal de recompensa acumulada ao longo do tempo [Sutton e Barto 2018]. O agente aprende a tomar decisões através de tentativa e erro, ajustando suas ações com base no *feedback* recebido. Diferentemente do aprendizado supervisionado, que utiliza exemplos rotulados, o aprendizado por reforço explora a recompensa como único sinal de desempenho, lidando com a complexidade de recompensas atrasadas e incertezas na transição de estados. Formalmente, o RL é modelado através de **processos de decisão de Markov** (*Markov Decision Processes* - MDPs) [Sutton e Barto 2018], que definem as interações em termos de estados, ações e recompensas, sendo amplamente aplicável a problemas de decisão sequencial em diversas áreas.

Dentre os métodos avançados de aprendizado por reforço, destaca-se o ***Proximal Policy Optimization* (PPO)** [Schulman et al. 2017], que é amplamente utilizado devido à sua estabilidade em ambientes complexos. Quando aplicado ao **aprendizado por reforço multiagente** (*Multi-agent RL*), permite que diversos agentes aprendam simultaneamente, interagindo de maneira colaborativa ou competitiva. Estratégias como o *self-play* têm mostrado grande eficácia ao permitir que agentes aprendam uns com os outros em cenários competitivos, como no futebol de robôs. Além disso, o *curriculum learning* [Narvekar et al. 2020] tem sido utilizado para estruturar a aprendizagem progressiva, começando com tarefas simples e avançando para desafios mais complexos [Foglino e Leonetti 2019], um aspecto crucial em domínios como o **futebol de robôs**, onde os agentes precisam coordenar habilidades motoras e estratégias de equipe para alcançar um bom desempenho.

2.1.1 Conceitos Básicos

A interação entre o agente e o ambiente é representada esquematicamente na Figura 2.1. O agente observa o estado atual S_t do ambiente e, com base em sua política de decisão, escolhe uma ação A_t. Após a execução dessa ação, o ambiente evolui para um novo estado S_{t+1} e retorna ao agente uma recompensa R_{t+1} associada a essa transição.

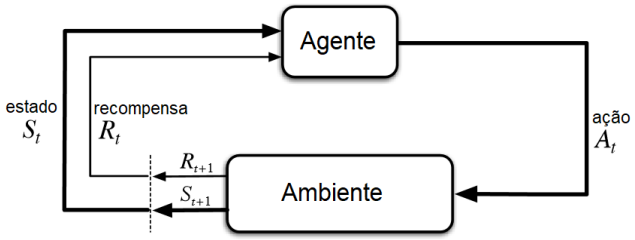

Figura 2.1: *Interação agente-ambiente no aprendizado por reforço. O agente toma decisões com base no estado atual S_t e recebe do ambiente uma recompensa R_{t+1} após realizar a ação A_t.*

Elementos Fundamentais:

1. **Agente e Ambiente:** O agente é a entidade responsável por tomar decisões, enquanto o ambiente é tudo aquilo que responde às ações do agente e fornece *feedback*.
2. **Política (π):** Define a estratégia do agente, especificando a probabilidade de selecionar uma ação específica A_t em um estado S_t.
3. **Sinal de Recompensa (R_{t+1}):** Indica o valor imediato recebido pelo agente após realizar uma ação. O objetivo é maximizar a soma acumulada das recompensas ao longo do tempo.
4. **Função de Valor ($v(s)$ e $q(s,a)$):** Estima o valor esperado da recompensa futura a partir de um estado s ou de um par estado-ação (s,a).

Exploração vs. Intensificação:

O dilema entre exploração e intensificação é um aspecto fundamental do aprendizado por reforço, especialmente em aprendizado multiagente [Schulman et al. 2017]. Em sistemas multiagente, este dilema torna-se complexo devido à não-estacionariedade do ambiente, onde os agentes precisam explorar enquanto se adaptam aos comportamentos em evolução de outros agentes [Brandão et al. 2022]. Técnicas como aprendizado centralizado com execução descentralizada (*Centralized Training with Decentralized Execution* - CTDE) e modelagem de oponentes são empregadas para gerenciar eficientemente este

compromisso, permitindo que os agentes compartilhem experiências de exploração enquanto mantêm estratégias eficazes de intensificação [Shen 2024].

Modelagem por Processos de Decisão de Markov (MDPs)

O aprendizado por reforço é frequentemente modelado por **Processos de Decisão de Markov** (*Markov Decision Processes* - MDPs) [Sutton e Barto 2018], uma estrutura matemática que captura os aspectos estocásticos e sequenciais da tomada de decisão. Um MDP é definido pela quádrupla (S, A, P, R), onde:

- S é o conjunto de estados possíveis do ambiente.
- A é o conjunto de ações possíveis que o agente pode tomar.
- $P(s'|s,a)$ é a distribuição de probabilidade de transição para o estado s' dado o estado atual s e a ação a tomada.
- $R(s,a)$ é a função de recompensa esperada ao tomar a ação a no estado s.

O agente busca maximizar a soma esperada das recompensas acumuladas ao longo do tempo, definida pelo **retorno** G_t [Sutton e Barto 2018]. No caso de problemas de horizonte finito, o retorno é por

$$G_t = \sum_{k=0}^{T} \gamma^k R_{t+k+1}. \tag{2-1}$$

Onde γ é o fator de desconto, que no caso de tarefas contínuas, aplica-se $\gamma \in [0,1]$ que controla o peso das recompensas futuras,

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}. \tag{2-2}$$

A modelagem dos MDPs requer duas funções principais de valor, que capturam a expectativa de recompensas futuras:

1. Função de valor de estado ($v_\pi(s)$) [Sutton e Barto 2018]: Define o valor esperado de se estar no estado s e seguir uma política π a partir desse estado, conforme a Equação

$$v_\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right]. \tag{2-3}$$

Essa função pode ser definida recursivamente pela **equação de Bellman**, como mostrado na Equação

$$v_\pi(s) = \sum_{a \in A} \pi(a|s) \sum_{s' \in S} P(s'|s,a) \left[R(s,a) + \gamma v_\pi(s') \right]; \tag{2-4}$$

2. Função de valor de ação ($q_\pi(s,a)$) [Sutton e Barto 2018]: Define o valor esperado de se tomar a ação a no estado s e seguir a política π posteriormente, como apresentado na Equação

$$q_\pi(s,a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right]. \tag{2-5}$$

Também pode ser escrita de forma recursiva, conforme a Equação

$$q_\pi(s,a) = \sum_{s'} P(s'|s,a) \left[R(s,a) + \gamma \sum_{a'} \pi(a'|s') q_\pi(s',a') \right]. \tag{2-6}$$

Equação de Bellman para o Ótimo [Sutton e Barto 2018]: Para políticas ótimas π^*, temos as funções de valor ótimo, dada pela Equação

$$v^*(s) = \max_a \sum_{s'} P(s'|s,a) \left[R(s,a) + \gamma v^*(s') \right], \tag{2-7}$$

e a função de valor de ação ótima, apresentada na Equação

$$q^*(s,a) = \sum_{s'} P(s'|s,a) \left[R(s,a) + \gamma \max_{a'} q^*(s',a') \right]. \tag{2-8}$$

Essas equações de Bellman são fundamentais para algoritmos de planejamento, como *Value Iteration* e *Policy Iteration*, mas quando a dinâmica do ambiente não é conhecida, métodos de aprendizado baseados em interação, como *Q-learning*, se tornam necessários.

Os MDPs constituem o núcleo matemático de muitos algoritmos de aprendizado por reforço, e são essenciais para o desenvolvimento de métodos de otimização de políticas, como o ***Proximal Policy Optimization* (PPO)**. Este algoritmo combina exploração e estabilidade ao ajustar políticas dentro de uma região de confiança definida.

2.1.2 PPO (*Proximal Policy Optimization*)

O *Proximal Policy Optimization* (PPO) representa um marco significativo no desenvolvimento de algoritmos de aprendizado por reforço, destacando-se pela sua combinação única de simplicidade de implementação, eficiência computacional e estabilidade durante o treinamento [Schulman et al. 2017]. Desenvolvido pela OpenAI, o PPO surgiu como uma evolução do *Trust Region Policy Optimization* (TRPO), introduzindo mecanismos mais eficientes para controlar a magnitude das atualizações de política [OpenAI 2018].

O diferencial do PPO está em sua abordagem para otimização de políticas, que utiliza uma *função objetivo substituta recortada* (*clipped surrogate objective*) [Schulman et al. 2017, OpenAI 2018]. Esta função limita efetivamente as mudanças na política, prevenindo atualizações muito grandes que poderiam desestabilizar o aprendizado. Matematicamente, a função objetivo do PPO é expressa como:

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) A_t, \text{clip}(r_t(\theta), 1-\varepsilon, 1+\varepsilon) A_t \right) \right], \tag{2-9}$$

onde $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ representa a razão entre as probabilidades das políticas nova e antiga, A_t é a estimativa da vantagem, e ε é um hiperparâmetro que controla o tamanho máximo da atualização [DLR-RM 2019, Schulman et al. 2017].

O PPO implementa uma arquitetura *actor-critic*, onde o *actor* é responsável pela política que mapeia estados para ações, enquanto o *critic* estima os valores dos estados para auxiliar no cálculo das vantagens [Schulman et al. 2017, PyTorch 2024]. Esta estrutura dual permite um equilíbrio eficiente entre exploração e aproveitamento do conhecimento adquirido. O processo de treinamento ocorre em episódios, onde experiências são coletadas e utilizadas para atualizar tanto a política quanto a função de valor [Schulman et al. 2017].

Uma característica fundamental do PPO é sua capacidade de lidar com espaços de ação tanto discretos quanto contínuos, tornando-o particularmente adequado para aplicações em robótica e controle [Sutton e Barto 2018, Schulman et al. 2017]. O algoritmo mantém um equilíbrio entre exploração e exploração através de um termo de entropia adicional na função objetivo, que incentiva a política a manter um nível apropriado de aleatoriedade nas decisões [Schulman et al. 2017].

O PPO tem demonstrado resultados impressionantes em uma variedade de domínios complexos, desde jogos até tarefas de robótica [Verleysen 2023]. Sua eficácia é particularmente notável em ambientes que requerem aprendizado contínuo e adaptação, como no caso do futebol de robôs, onde os agentes precisam constantemente ajustar suas estratégias em resposta a situações dinâmicas [Brandão et al. 2022].

No contexto do aprendizado multiagente, o PPO pode ser estendido para treinar múltiplos agentes simultaneamente, permitindo o desenvolvimento de comportamentos cooperativos ou competitivos [PyTorch 2024]. Esta característica é especialmente relevante para cenários de equipe, como o futebol de robôs, onde os agentes precisam coordenar suas ações para alcançar objetivos comuns [Verleysen 2023, Brandão et al. 2022].

2.1.3 Multi-agent RL

O Aprendizado por Reforço Multiagente (*Multi-agent Reinforcement Learning* - MARL) representa uma extensão fundamental do RL tradicional, onde múltiplos agentes

autônomos interagem em um ambiente compartilhado, aprendendo simultaneamente através de suas experiências coletivas. Como destacado em [Brandão et al. 2022], esta abordagem expande o paradigma clássico dos Processos de Decisão Markovianos (MDPs) para acomodar as complexidades inerentes às interações entre múltiplos tomadores de decisão.

A transição do RL *single-agent* para o MARL introduz desafios significativos, principalmente relacionados à não-estacionariedade do ambiente. Como observado em [6], quando múltiplos agentes atualizam suas políticas simultaneamente, o ambiente torna-se não-estacionário do ponto de vista de cada agente individual, violando premissas fundamentais do RL tradicional. Esta característica exige o desenvolvimento de algoritmos especializados que possam lidar com a natureza dinâmica das interações multiagente.

Um aspecto crucial do MARL é a distinção entre cenários cooperativos e competitivos [Oroojlooyjadid e Hajinezhad 2019]. Em ambientes cooperativos, os agentes trabalham juntos para maximizar uma recompensa global compartilhada, enquanto em cenários competitivos, cada agente busca otimizar sua própria recompensa, potencialmente em detrimento dos outros.

A escalabilidade representa outro desafio significativo no MARL. De acordo com [Chang et al. 2022], o espaço de ações conjunto cresce exponencialmente com o número de agentes, tornando necessário o desenvolvimento de técnicas de decomposição e aproximação. Abordagens recentes, como o MAZero, têm demonstrado sucesso ao combinar planejamento baseado em modelo com técnicas de busca em árvore *Monte Carlo* para navegar eficientemente por estes espaços complexos.

Para lidar com estes desafios, diversas arquiteturas e algoritmos têm sido propostos. Almeida [Almeida 2013] destaca a importância de métodos como QMIX e MADDPG, que utilizam estruturas de valor decompostas e críticos centralizados para facilitar o aprendizado em ambientes multiagente. Estas abordagens permitem o treinamento centralizado com execução descentralizada, um paradigma que tem se mostrado particularmente eficaz em aplicações práticas.

A aplicação do MARL em domínios reais tem demonstrado resultados promissores. Como evidenciado em [Martins 2023], no contexto do futebol de robôs, abordagens multiagente superam significativamente métodos *single-agent*, especialmente em cenários que exigem coordenação complexa entre os membros da equipe. No entanto, o sucesso destas aplicações depende crucialmente da modelagem adequada das funções de recompensa e da implementação de mecanismos eficientes de comunicação entre agentes.

Desenvolvimentos recentes no campo, como destacado em [Huh e Mohapatra 2024], têm explorado a integração do MARL com outras tecnologias emergentes, como grandes modelos de linguagem e técnicas de aprendizado por demonstração. Estas integrações abrem novos caminhos para o desenvolvimento de sistemas multiagente mais sofistica-

dos e adaptáveis, capazes de lidar com tarefas cada vez mais complexas em ambientes dinâmicos.

2.1.4 *Self-play*

O *Self-play* emergiu como uma técnica fundamental no aprendizado por reforço, permitindo que agentes artificiais desenvolvam habilidades avançadas através do treinamento contra versões de si mesmos. Como destacado em [DiGiovanni e Zell 2021], esta abordagem permite que os agentes criem um currículo automático de aprendizado, onde enfrentam adversários progressivamente mais desafiadores, evitando assim a estagnação em estratégias subótimas.

A eficácia do *self-play* foi demonstrada de forma notável em diversos domínios complexos. Conforme documentado em [Silver et al. 2017], sistemas como AlphaGo e AlphaZero alcançaram desempenho sobre-humano em jogos estratégicos complexos, aprendendo exclusivamente através da interação consigo mesmos.

Um aspecto crucial do *self-play*, como apontado em [Zhang et al. 2024], é sua capacidade de gerar dados de treinamento de alta qualidade sem necessidade de supervisão humana. Esta característica é particularmente valiosa em domínios onde exemplos de especialistas são escassos ou caros de obter. O processo permite que os agentes explorem o espaço de estratégias de forma mais abrangente do que seria possível com dados puramente supervisionados.

A implementação moderna do *self-play* frequentemente incorpora arquiteturas sofisticadas. De acordo com [Ji et al. 2024], o método SPAC (*Self-Play Actor-Critic*) introduziu inovações significativas ao integrar um crítico que considera as observações de ambos os agentes em ambientes competitivos, superando métodos tradicionais em eficácia.

No entanto, o *self-play* também apresenta desafios significativos. Como observado em [Bai e Jin 2020], existe o risco de convergência para equilíbrios subótimos, especialmente em jogos com múltiplos equilíbrios de Nash. Para mitigar este problema, técnicas como diversificação de oponentes e regularização de entropia têm sido empregadas com sucesso.

Desenvolvimentos recentes, descritos em [DiGiovanni e Zell 2021, Zhang et al. 2024], têm explorado a integração do *self-play* com paradigmas emergentes como meta-aprendizado e modelos de linguagem grandes. Esta convergência tem aberto novos caminhos para o desenvolvimento de agentes mais adaptativos e versáteis.

A aplicação do *self-play* estende-se além dos jogos tradicionais. Como documentado em [Zhang et al. 2024] e [Brandão et al. 2022], a técnica tem sido aplicada com sucesso em domínios como robótica, simulações financeiras e sistemas de controle autô-

nomo. Em cada caso, o *self-play* permite que os agentes desenvolvam estratégias robustas através da exploração sistemática do espaço de possibilidades.

O futuro do *self-play* parece promissor, com pesquisas contínuas focando na melhoria da eficiência computacional e na expansão de suas aplicações. Como sugerido em [Face 2024], novas técnicas de otimização e paralelização estão tornando o treinamento mais acessível, permitindo sua aplicação em uma gama cada vez maior de problemas práticos.

2.2 *Curriculum Learning*

O *Curriculum Learning* (CL) representa uma abordagem metodológica inspirada nos princípios pedagógicos humanos, onde o processo de aprendizagem é estruturado de forma progressiva, começando com tarefas mais simples e avançando gradualmente para desafios mais complexos. Esta metodologia tem demonstrado resultados significativos na otimização do treinamento de agentes de aprendizado, especialmente em contextos de *Reinforcement Learning* (RL), como destacado por [Soviany et al. 2022].

2.2.1 Conceitos Fundamentais

O *Curriculum Learning* fundamenta-se em três componentes principais que trabalham em conjunto para estruturar o processo de aprendizagem. Como descrito em [Narvekar et al. 2020], estes componentes são: geração de tarefas, sequenciamento e transferência de conhecimento.

A geração de tarefas envolve a criação sistemática de desafios intermediários que estabelecem uma ponte entre o estado inicial do agente e o objetivo final desejado. De acordo com [Klink et al. 2022], este processo pode incluir a modificação controlada de parâmetros ambientais, como a densidade de recompensas ou a complexidade dos estados apresentados ao agente.

O sequenciamento, por sua vez, representa a ordenação estratégica das tarefas geradas, garantindo uma progressão coerente e efetiva no processo de aprendizagem. Como destacado em [Narvekar et al. 2020], esta ordenação pode ser realizada através de métodos como Interpolação de Distribuições ou *Optimal Transport*, que otimizam a trajetória de aprendizado do agente.

A transferência de conhecimento, elemento crucial do CL, permite que o aprendizado adquirido em tarefas anteriores seja efetivamente utilizado para acelerar o domínio de novos desafios. [Sayar et al. 2024] demonstra que técnicas como o *Boosted Curriculum RL* (BCRL) podem aproximar funções de valor como somas de resíduos treinados incrementalmente, aumentando significativamente a capacidade expressiva do modelo.

Um aspecto fundamental do CL, conforme apresentado por [Narvekar et al. 2020], é a utilização de métricas de dificuldade para calibrar a progressão do currículo. Estas métricas podem incluir a entropia das políticas aprendidas ou a taxa de sucesso em submetas específicas, permitindo um ajuste dinâmico do processo de aprendizagem.

No contexto específico do *Reinforcement Learning*, o CL se integra naturalmente com os Processos de Decisão de Markov (MDPs), onde o agente interage com ambientes parametrizados de forma progressiva [Sutton e Barto 2018]. Esta integração, como explicado em [Klink et al. 2022], permite uma acumulação estruturada de experiência, fundamental para o desenvolvimento de políticas robustas e eficientes.

2.2.2 Aplicações em RL

No contexto do *Reinforcement Learning*, o *Curriculum Learning* tem se mostrado particularmente eficaz em cenários complexos que apresentam desafios significativos para métodos tradicionais de aprendizagem. Como destacado por [Klink et al. 2022], a aplicação de currículos estruturados tem permitido avanços notáveis em domínios que exigem raciocínio hierárquico e planejamento de longo prazo.

Um exemplo significativo é encontrado em ambientes de navegação autônoma, onde [Portelas et al. 2021, Face 2024] demonstra que agentes treinados com CL desenvolvem estratégias mais robustas ao serem expostos gradualmente a ambientes de complexidade crescente. O processo começa com cenários simplificados, como navegação em espaços abertos, e progride para ambientes com obstáculos dinâmicos e restrições temporais, resultando em políticas mais generalizáveis.

Em aplicações de robótica, [Narvekar et al. 2020] apresenta uma abordagem onde o CL é utilizado para decompor tarefas motoras complexas em subcomponentes mais gerenciáveis. Os autores demonstram que esta decomposição hierárquica não apenas acelera o aprendizado, mas também melhora significativamente a qualidade das políticas aprendidas, especialmente em tarefas que requerem coordenação motora fina.

A integração do CL com técnicas de aprendizado multiagente tem produzido resultados promissores, como evidenciado por [Narvekar 2017]. Em cenários competitivos e cooperativos, o uso de currículos adaptativos permite que os agentes desenvolvam estratégias sofisticadas através da exposição gradual a adversários ou parceiros de diferentes níveis de habilidade. Esta abordagem tem se mostrado particularmente eficaz em domínios como o futebol de robôs, onde a complexidade das interações entre agentes pode tornar o aprendizado direto impraticável.

Um aspecto crucial na aplicação do CL em RL, conforme destacado por [18], é a capacidade de automatizar a geração e adaptação de currículos. Técnicas recentes utilizam meta-aprendizado para otimizar a sequência de tarefas, permitindo que o currículo se

ajuste dinamicamente ao progresso do agente. Esta automatização não apenas reduz a necessidade de intervenção humana, mas também permite a descoberta de sequências de treinamento não intuitivas que podem levar a um melhor desempenho.

A eficácia do CL em RL também se estende a domínios com espaços de estado contínuos e de alta dimensionalidade. [Zhou et al. 2022] demonstra como currículos bem projetados podem guiar a exploração em espaços complexos, reduzindo significativamente o tempo necessário para encontrar políticas ótimas. Esta abordagem tem se mostrado particularmente valiosa em aplicações industriais, onde o custo de exploração aleatória pode ser proibitivo.

Diversos avanços significativos têm sido feitos na área de *Curriculum Learning*, incluindo automatização do design de currículos, integração com aprendizado profundo, *frameworks* baseados em modelos de difusão para geração automática de tarefas, coordenação de currículos paralelos em sistemas multiagente e desenvolvimento de currículos auto-adaptativos baseados em meta-aprendizado [Klink et al. 2022, Sayar et al. 2024, Narvekar 2017, 18]. Estas inovações têm contribuido para melhorar a forma como os currículos são estruturados e aplicados, permitindo uma evolução significativa na área.

2.2.3 Vantagens e desafios

A implementação do *Curriculum Learning* apresenta tanto benefícios substanciais quanto desafios significativos que precisam ser cuidadosamente considerados. Como observado por [Narvekar 2017], as vantagens incluem uma aceleração significativa no processo de aprendizagem e uma melhoria na qualidade das políticas aprendidas.

Entre as principais vantagens, [Klink et al. 2022] destaca a redução significativa no tempo de treinamento, com alguns experimentos mostrando uma diminuição de até 60% no número de iterações necessárias para convergência. Além disso, os autores observam uma melhoria na robustez das políticas aprendidas, com agentes demonstrando maior capacidade de generalização para cenários não vistos durante o treinamento.

No entanto, existem desafios importantes a serem considerados. Como apontado por [Klink et al. 2022], um dos principais obstáculos é a definição apropriada de métricas de dificuldade para diferentes tipos de tarefas. A subjetividade inerente à noção de "dificuldade"pode tornar complexa a automatização completa do processo de *design* do currículo.

[Narvekar et al. 2020] identifica outro desafio significativo: o balanceamento entre exploração e exploração durante o processo de aprendizagem. Currículos mal calibrados podem levar a uma convergência prematura para soluções subótimas ou, alternativamente, resultar em uma exploração excessiva que compromete a eficiência do treinamento.

Um aspecto particularmente desafiador, conforme destacado por [Zhou et al. 2022], é a necessidade de recursos computacionais significativos para a implementação efetiva de currículos adaptativos. O custo de gerar e validar sequências de treinamento personalizadas pode ser proibitivo para algumas aplicações, especialmente em domínios que requerem simulações complexas ou processamento em tempo real.

Apesar destes desafios, na literatura, como evidenciado por [Soviany et al. 2022], a expectativa é que os benefícios do *Curriculum Learning* geralmente superam suas limitações, especialmente em domínios complexos onde abordagens tradicionais de aprendizado por reforço mostram-se inadequadas. A contínua evolução de técnicas automatizadas para *design* de currículos e a crescente disponibilidade de recursos computacionais sugerem um futuro promissor para esta abordagem.

2.3 Futebol de Robôs

O futebol de robôs representa uma das principais plataformas para pesquisa e desenvolvimento em robótica móvel e sistemas multiagentes. Dentro deste contexto, a *RoboCup Small Size League Entry Level* (SSL-EL) emerge como uma categoria especialmente projetada para facilitar a entrada de novas equipes, mantendo os desafios técnicos fundamentais do futebol de robôs enquanto simplifica aspectos complexos da competição [30].

2.3.1 Visão geral

A SSL-EL foi estabelecida como uma divisão de entrada da *Small Size League*, visando democratizar o acesso à competição de futebol de robôs. Diferentemente da divisão principal, a SSL-EL opera com 6 robôs por equipe em um campo reduzido, permitindo que equipes iniciantes desenvolvam suas habilidades técnicas e estratégicas de forma progressiva [30].

O ambiente de jogo é estruturado em torno de um sistema centralizado de visão (*SSL-Vision*), que fornece informações em tempo real sobre as posições dos robôs e da bola através de câmeras suspensas [Technical Overview of the Small Size League 2025]. Esta arquitetura permite que as equipes foquem no desenvolvimento de estratégias de controle e coordenação, sem a necessidade inicial de implementar sistemas complexos de percepção local.

Os robôs da SSL-EL, embora sujeitos a restrições dimensionais similares às da divisão principal (diâmetro ≤ 18 cm, altura ≤ 15 cm) [30], podem ser construídos com soluções mais acessíveis. A equipe *TurtleRabbit*, por exemplo, demonstrou a viabilidade de

construir robôs competitivos com orçamento reduzido, utilizando componentes comerciais e estruturas impressas em 3D [TurtleRabbit 2024 SSL Team Description Paper 2024].

A competição promove desafios técnicos específicos, como o *Ball Placement Challenge*, que permitem às equipes desenvolverem e testarem capacidades fundamentais de forma isolada [Technical Overview of the Small Size League 2025]. Esta abordagem gradual ao desenvolvimento de habilidades tem se mostrado efetiva para a evolução das equipes, como evidenciado pelos resultados de competições recentes [Technical Overview of the Small Size League 2025].

2.3.2 Desafios específicos

A SSL-EL apresenta desafios únicos que equilibram complexidade técnica com acessibilidade. Um dos principais desafios é a coordenação multiagente em tempo real, onde os robôs devem operar de forma sincronizada em velocidades consideráveis, tomando decisões em milissegundos [30]. Esta coordenação torna-se ainda mais complexa devido aos ruídos de percepção e latência de comunicação inerentes ao sistema.

O aspecto financeiro representa outro desafio significativo para equipes iniciantes. No entanto, iniciativas como a da equipe *TurtleRabbit* demonstram que é possível desenvolver soluções competitivas com orçamento limitado [TurtleRabbit 2024 SSL Team Description Paper 2024]. Através do uso de componentes comerciais acessíveis e técnicas de manufatura como impressão 3D, equipes podem construir robôs funcionais mantendo os custos controlados.

O processamento em tempo real das informações do *SSL-Vision* apresenta desafios técnicos específicos [Technical Overview of the Small Size League 2025]. As equipes precisam desenvolver sistemas robustos para lidar com:

- Fusão de dados de múltiplas câmeras;
- Compensação de falhas momentâneas na detecção;
- Filtragem de ruídos e correção de discrepâncias;
- Predição de movimentos para compensar latências.

A implementação de estratégias de jogo efetivas também representa um desafio significativo. Como documentado pela equipe *RoboCIn* [RobôCIn Small Size League Extended Team Description Paper for RoboCup 2024 2024], é necessário desenvolver algoritmos sofisticados para navegação, controle de bola e coordenação tática, mesmo no contexto simplificado da SSL-EL.

2.3.3 Trabalhos relacionados

Diversos trabalhos têm contribuído para o avanço da SSL-EL, focando em diferentes aspectos da competição. As equipes *ITAndroids* e *OrcaBOT*, por exemplo, desenvolveu uma abordagem inovadora para o *design* de robôs, documentada em seu *TDP* (*Team Description Paper*) [OrcaBOT Team Description Paper 2024 2024, ITAndroids Small Size League Team Description Paper for RoboCup 2023 2023], demonstrando a viabilidade de construir sistemas competitivos com recursos limitados.

No campo do controle e estratégia, trabalhos significativos têm emergido da comunidade acadêmica [Brandão et al. 2022, Bassani et al. 2020]. Estas pesquisas demonstram o potencial de algoritmos de aprendizado de máquina para desenvolver estratégias de jogo mais sofisticadas.

A organização da *RoboCup* tem contribuído significativamente através da documentação e padronização de regras e especificações técnicas [30]. Este trabalho de documentação tem sido fundamental para permitir que novas equipes ingressem na competição com uma compreensão clara dos requisitos e desafios.

Iniciativas de código aberto, como as disponibilizadas pela comunidade brasileira de robótica [Repositório oficial da RoboCup SSL Brasil 2024], têm facilitado o desenvolvimento de novas equipes. Estes recursos incluem implementações de referência para sistemas de controle, simuladores e ferramentas de desenvolvimento, permitindo que equipes iniciantes construam sobre uma base sólida de conhecimento estabelecido.

CAPÍTULO 3

Metodologia

3.1 Reprodutibilidade

Esta seção detalha os aspectos técnicos necessários para a reprodução dos experimentos realizados neste trabalho. A implementação segue uma formulação baseada em Processo de Decisão de Markov Parcialmente Observável Descentralizado (Dec-POMDP), com detalhamento do ambiente de simulação, configurações, estados, ações, função de recompensa e toda a estrutura de Curriculum Learning implementada.

A Figura 3.1 apresenta uma visão geral do pipeline de treinamento implementado, destacando a integração entre as abordagens de Curriculum Learning e Self-play, que serão detalhadas ao longo desta seção.

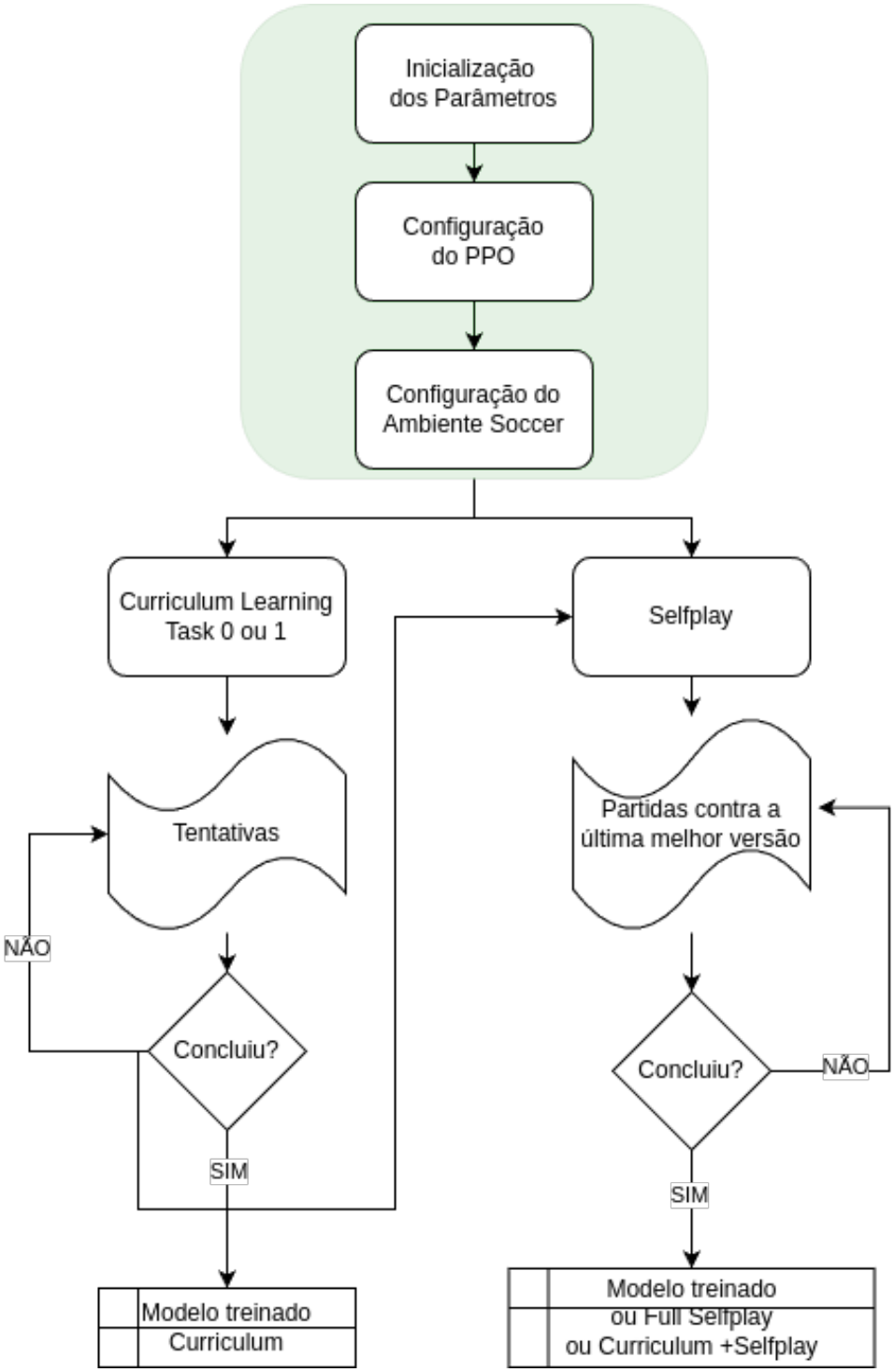

Figura 3.1: *Fluxograma do pipeline de treinamento, destacando a integração entre Curriculum Learning e Self-play. O processo inicia com a inicialização e configuração dos componentes, seguido pela divisão entre as abordagens de Curriculum Learning e Self-play direto. A fase de Curriculum Learning inclui tarefas progressivas (Task 0 ou 1) seguidas por ciclos de tentativas até conclusão do critério de promoção, resultando em um modelo treinado que serve como base para o Self-play subsequente. Fonte: Elaborado pelo autor.*

3.1.1 Formulação Dec-POMDP

O problema de controle multiagente em futebol de robôs é formulado como um Dec-POMDP, representado pela tupla:

$$G = \langle D, A, S, Z, P, R \rangle.$$

Onde:

- D: Conjunto de agentes robóticos que participam do jogo;
- A: Espaço de ações disponíveis para cada agente;
- S: Espaço de estados do ambiente (não completamente observável pelos agentes);
- Z: Espaço de observações parciais que cada agente recebe;
- P: Função de transição que determina a dinâmica do ambiente;
- R: Função de recompensa que guia o aprendizado dos agentes.

A formulação Dec-POMDP é particularmente adequada para este domínio, pois os agentes precisam tomar decisões baseadas em observações locais parciais, enquanto colaboram para alcançar objetivos comuns, como marcar gols e evitar que o adversário marque.

3.1.2 Conjunto de Agentes D

O conjunto de agentes varia conforme o estágio do curriculum learning e o tipo de experimento. A configuração padrão para o jogo completo é:

$$D_{azul} = \{d_1, d_2, d_3\}$$

para o time azul e

$$D_{amarelo} = \{d_4, d_5, d_6\}$$

para o time amarelo.

O número de agentes é dinamicamente ajustado durante as diferentes fases do curriculum learning:

- **Curriculum Task 0 (Fundamentos Básicos)**: $D_{azul} = \{d_1\}$ e $D_{amarelo} = \{\emptyset\}$ (nenhum agente adversário);
- **Curriculum Task 1 (Interação com Oponentes Estáticos)**: $D_{azul} = \{d_1, d_2, d_3\}$ e $D_{amarelo} = \{d_4\}$ (apenas um agente adversário estático);
- **Self-play**: $D_{azul} = \{d_1, d_2, d_3\}$ e $D_{amarelo} = \{d_4, d_5, d_6\}$ (configuração completa).

Todos os agentes de um mesmo time compartilham os mesmos parâmetros de política (policy sharing), o que facilita o aprendizado coletivo e reduz a dimensionalidade

do problema. Essa abordagem de compartilhamento de política permite que comportamentos emergentes de coordenação surjam naturalmente durante o treinamento.

3.1.3 Espaço de Ações A

Cada agente possui um espaço de ação contínuo com 4 dimensões:

$$a_d = (v_x, v_y, \omega_\theta, k) \in [-1.0, 1.0]^4.$$

Onde:

- v_x: Velocidade normalizada no eixo x (movimento lateral);
- v_y: Velocidade normalizada no eixo y (movimento frontal);
- ω_θ: Velocidade angular normalizada (rotação);
- k: Ação de chute (contínua, onde valores > 0 executam o chute).

Estas ações de alto nível são convertidas para comandos de baixo nível das quatro rodas do robô usando a cinemática do robô omnidirecional. A conversão segue o seguinte processo:

Transformação de coordenadas

Primeiro, as velocidades no referencial global são transformadas para o referencial local do robô:

$$v'_x, v'_y = v_x \cdot \cos(\theta) + v_y \cdot \sin(\theta), -v_x \cdot \sin(\theta) + v_y \cdot \cos(\theta).$$

Onde θ é a orientação atual do robô.

Cálculo das velocidades angulares das rodas

Em seguida, as velocidades locais e a velocidade angular são convertidas para comandos de velocidade das quatro rodas omnidirecionais:

$$\omega_{roda0} = \frac{v'_y}{r} - \frac{L \cdot \omega_\theta}{r}.$$

$$\omega_{roda1} = \frac{v'_x}{r} + \frac{L \cdot \omega_\theta}{r}.$$

$$\omega_{roda2} = -\frac{v'_y}{r} - \frac{L \cdot \omega_\theta}{r}.$$

$$\omega_{roda3} = -\frac{v'_x}{r} + \frac{L \cdot \omega_\theta}{r}.$$

Onde r é o raio das rodas e L é a distância do centro do robô até as rodas.

Desnormalização e limitação

Os valores normalizados são desnormalizados para as velocidades físicas reais:

- Velocidades lineares são multiplicadas por 1.5 m/s
- Velocidade angular é multiplicada por 10 rad/s

Os valores resultantes são limitados às velocidades máximas permitidas pelos atuadores do robô. Esta abordagem permite que os agentes aprendam comportamentos complexos emergentes, como driblar, passar e defender, a partir deste conjunto de ações de baixo nível.

3.1.4 Espaço de Estados S

O estado completo S do ambiente representa todas as informações físicas do sistema simulado, incluindo:

- Posição (x, y) e orientação θ de todos os robôs;
- Velocidades lineares (v_x, v_y) e angulares ω de cada robô;
- Posição (x, y) e velocidade (v_x, v_y) da bola;
- Estado dos atuadores (velocidades das rodas);
- Forças e interações físicas entre todos os objetos do ambiente;
- Informações sobre limites do campo, área de gol e outras regiões relevantes.

Este estado completo é gerenciado pelo simulador RL-SSL-EL, que implementa a física do ambiente. No entanto, os agentes não têm acesso direto a todas essas informações, apenas a um conjunto de observações parciais derivadas do estado completo, muitas vezes com ruído adicionado para simular imperfeições sensoriais.

A formulação como Dec-POMDP é particularmente adequada neste contexto, pois reconhece que cada agente tem uma visão parcial e potencialmente ruidosa do ambiente, precisando tomar decisões com informação incompleta, semelhante ao problema enfrentado por robôs reais em um jogo físico.

3.1.5 Espaço de Observações Z

Cada agente d recebe uma observação parcial $z_d \in Z$ do estado do ambiente. Esta observação é um vetor de 77 valores que inclui informações relevantes para a tomada de decisão do agente. Para capturar aspectos temporais e dinâmicos do jogo, estas observações são empilhadas com as 7 observações anteriores, resultando em um vetor final de entrada com 616 valores (8×77).

O vetor de observação é composto pelas seguintes categorias de informação:

Posições Cartesianas

- Posição (x, y) normalizada do próprio robô relativa ao centro do campo;
- Posições (x, y) normalizadas dos companheiros de time relativas ao centro do campo;
- Posições (x, y) normalizadas dos adversários relativas ao centro do campo;
- Posição (x, y) normalizada da bola relativa ao centro do campo.

Orientações Angulares

- Seno e cosseno da orientação do próprio robô para representação contínua sem descontinuidades;
- Arcotangente da orientação do próprio robô como informação complementar;
- Seno, cosseno e arcotangente dos ângulos formados entre o robô e seus aliados;
- Seno, cosseno e arcotangente dos ângulos formados entre o robô e os adversários;
- Seno, cosseno e arcotangente dos ângulos formados entre a bola e o centro de cada gol.

Distâncias Euclidianas

- Entre o robô e cada companheiro de time;
- Entre o robô e cada adversário;
- Entre o robô e a bola;
- Entre a bola e ambos os gols (aliado e adversário).

Informações Temporais e Contextuais

- Ações anteriores de todos os companheiros de time no tempo $t-1$;
- Tempo restante no episódio (normalizado para o intervalo $[0, 1]$).

Todas as observações são normalizadas para o intervalo $[-1.0, 1.0]$ para facilitar o treinamento da rede neural. A normalização é realizada dividindo os valores brutos por constantes específicas para cada tipo de dado:

- Posições: divididas pelo tamanho máximo do campo;
- Distâncias: divididas pela diagonal do campo;
- Velocidades: divididas pelas velocidades máximas permitidas.

O empilhamento temporal de 8 frames a 10Hz (correspondendo a 0.8 segundos de jogo) permite que o agente infira informações sobre velocidades e trajetórias, mesmo sem acesso direto a estes dados. Esta representação do estado foi cuidadosamente projetada para fornecer ao agente informações suficientes para tomar decisões tático-estratégicas eficazes.

3.1.6 Função de Transição P

A função de transição $P(s'|s,a)$ representa a dinâmica do ambiente, determinando como o estado s evolui para o próximo estado s' após a execução da ação conjunta a por todos os agentes. No contexto deste trabalho, a dinâmica é implementada pelo ambiente de simulação *RL-SSL-EL*, que oferece simulação física do futebol de robôs, incluindo movimentação dos robôs, dinâmica da bola, e interações entre os agentes no campo.

Dinâmica Física Simulada

A simulação inclui todos os aspectos físicos relevantes:

- **Robôs**: Modelados como corpos rígidos com quatro rodas independentes, massa e momento de inércia realistas;
- **Bola**: Modelada como uma esfera com propriedades físicas apropriadas para simulação de deslizamento, rolamento e colisões;
- **Colisões**: Detecção de colisões entre robôs, bola e limites do campo;
- **Atrito**: Implementação de atrito entre diferentes superfícies, tanto estático quanto dinâmico;
- **Mecanismo de chute**: Simulação da força aplicada à bola quando a ação de chute é ativada.

Regras do Jogo

Além da física pura, a função de transição também implementa as regras específicas do jogo de futebol de robôs:

- **Resets após gol**: Quando um gol é marcado, todos os robôs e a bola são reposicionados para as suas posições iniciais;
- **Resets laterais**: Quando a bola sai pelos limites laterais do campo, os robôs e a bola são reposicionados, mas o episódio continua;
- **Resets de linha de fundo**: Quando a bola sai pelos limites de fundo do campo, os robôs e a bola são reposicionados, mas o episódio continua;
- **Faltas**: Implementação de regras básicas para identificação e penalização de faltas.

Detalhes da Implementação

A simulação opera com os seguintes parâmetros:

- **Frequência de simulação física**: 30 FPS (frames por segundo) para garantir estabilidade e precisão;
- **Frequência de observação e ação**: 10 Hz para os agentes;

- **Duração dos episódios**: 40 segundos de jogo simulado;
- **Limite de passos por episódio**: Variável conforme o estágio do curriculum (300 a 500 passos)

Esta função de transição procura equilibrar o realismo físico necessário para simular adequadamente o futebol de robôs com a eficiência computacional exigida para treinamento em aprendizado por reforço, que requer milhões de interações com o ambiente.

3.1.7 Função de Recompensa R

A função de recompensa é um componente crítico do sistema de aprendizado por reforço, pois guia o comportamento dos agentes em direção aos objetivos desejados. Em nosso trabalho, projetamos uma estrutura de recompensa adaptativa que evolui conforme os estágios do curriculum learning, permitindo o desenvolvimento progressivo de habilidades.

Recompensa Padrão (Self-play)

No estágio final de Self-play, a recompensa combina elementos contínuos e discretos. A componente contínua r é formada pela soma ponderada de quatro termos:

$$r = 0.7 \cdot r_{speed} + 0.1 \cdot r_{dist} + 0.1 \cdot r_{off} + 0.1 \cdot r_{def}.$$

Onde:

- $r_{speed} = \text{clip}\left(\frac{\text{dist}(b_{t-1},G) - \text{dist}(b_t,G) - 0.05}{0.14}, -1.0, 1.0\right)$.
  Onde b_t é a posição da bola no tempo t, G é a posição do gol adversário, e dist() é a distância euclidiana. Os fatores 0.05 e 0.14 são parâmetros de calibração que estabelecem o limiar mínimo de movimento necessário e a escala de normalização, respectivamente.
- $r_{dist} = \begin{cases} -1, & \text{se } \min(P) \geq 1 \\ -\min(P), & \text{se } \min(P) < 1 \end{cases}$.
  Onde P é o conjunto das distâncias normalizadas entre cada robô do time e a bola. Esta recompensa incentiva que pelo menos um jogador esteja próximo da bola, penalizando situações onde todos os agentes estão distantes da bola.
- $r_{off} = \frac{\theta_{RBG}}{\pi} - 1$.
  Onde θ_{RBG} é o ângulo formado pelo robô (R), a bola (B) e o gol adversário (G). Este ângulo é normalizado para o intervalo $[-1.0, 0.0]$, onde valores mais próximos de zero representam posições ofensivas mais vantajosas.

- $r_{def} = \frac{\theta_{GRB}}{\pi} - 1$.
  Onde θ_{GRB} é o ângulo formado pelo gol aliado (G), o robô (R) e a bola (B). Similar ao r_{off}, esta recompensa incentiva posições defensivas eficazes para bloquear possíveis ataques adversários.

Recompensas de Eventos

Além da recompensa contínua, eventos discretos importantes geram recompensas significativas:

- **Gol marcado**: $+10$;
- **Gol sofrido**: -10;
- **Bola fora do campo**: -1.

Estes valores sobrepõem-se à recompensa contínua quando ocorrem, criando sinais fortes para guiar o aprendizado em momentos críticos.

Recompensas Específicas do Curriculum Learning

Cada estágio do curriculum possui ajustes específicos na função de recompensa, adaptados aos objetivos pedagógicos de cada fase:

- **Curriculum Task 0 (Fundamentos Básicos):**
  - **Recompensa principal**: $+10.0$ quando o robô toca na bola pela primeira vez;
  - **Recompensa de aproximação**: $-dist(rob, bola)$, incentivando a aproximação contínua à bola;
  - **Penalidade por tempo**: -0.01 por passo, incentivando ações rápidas;
  - **Critério de sucesso**: Tocar na bola pelo menos uma vez durante o episódio.
- **Curriculum Task 1 (Interação com Oponentes Estáticos):**
  - **Proximidade da bola**: $-0.1 \times dist(rob\_mais\_prximo, bola)$;
  - **Posse de bola**: $+0.05 \times \min(frames\_com\_posse/30, 1.0)$, recompensando o controle contínuo da bola;
  - **Aproximação do gol**: $+0.3 \times (1 - dist\_normalizada(bola, gol\_adversrio))$;
  - **Gol marcado**: $+10.0$;
  - **Gol sofrido**: -10.0;
  - **Penalidade por sair do campo**: -1.0 (robô), -0.5 (bola);
  - **Critério de sucesso**: Marcar pelo menos um gol durante o episódio.

Esta estrutura de recompensa progressiva permite que os agentes desenvolvam habilidades em uma sequência pedagógica eficaz, facilitando o aprendizado de comportamentos complexos a partir de capacidades fundamentais mais simples. A transição suave

entre os diferentes regimes de recompensa é essencial para o sucesso da abordagem de curriculum learning implementada.

3.1.8 Curriculum Learning

O Curriculum Learning implementado neste trabalho consiste em uma abordagem estruturada e progressiva para o aprendizado de habilidades complexas em futebol de robôs. Diferentemente da abordagem tradicional de Self-play, onde os agentes enfrentam diretamente o ambiente completo, o curriculum divide o aprendizado em estágios sequenciais de complexidade crescente, facilitando o desenvolvimento gradual de competências.

Estrutura do Curriculum

Nosso curriculum foi cuidadosamente projetado com dois estágios principais, cada um focado em um conjunto específico de habilidades fundamentais:

1. **Curriculum Task 0 (Fundamentos Básicos)**:
    - **Objetivo**: Desenvolver habilidades básicas de navegação e interação com a bola;
    - **Configuração**: 1 agente azul sem oponentes em campo;
    - **Ambiente**: Campo simplificado com a bola posicionada no centro;
    - **Posições iniciais**: Fixas com pequenas variações aleatórias (±0.3 metros);
    - **Critério de promoção**: 80% de sucesso (tocar na bola) em 100 episódios consecutivos;
    - **Duração típica**: 55 iterações de treinamento (aproximadamente 20 minutos).
2. **Curriculum Task 1 (Interação com Oponentes Estáticos)**:
    - **Objetivo**: Desenvolver habilidades de coordenação entre agentes e estratégias ofensivas;
    - **Configuração**: 3 agentes azuis contra 1 oponente amarelo estático;
    - **Ambiente**: Campo completo com posicionamento tático dos agentes;
    - **Posições iniciais**: Semi-aleatórias em configurações taticamente relevantes;
    - **Critério de promoção**: 80% de sucesso (marcar gol) em 100 episódios consecutivos;
    - **Duração típica**: 60 iterações de treinamento (aproximadamente 22 minutos).

Mecanismo de Promoção Adaptativa

A transição entre os estágios do curriculum é controlada por um sistema de promoção adaptativa implementado na classe `CurriculumCallback`. Este mecanismo opera da seguinte forma:

- Mantém uma janela deslizante dos últimos 100 episódios;
- Calcula continuamente a taxa de sucesso com base nos critérios específicos de cada estágio;
- Monitora o desempenho dos agentes em todos os ambientes paralelos;
- Promove automaticamente para o próximo estágio quando a taxa de sucesso atinge 80
- Preserva os pesos da política durante a transição, permitindo transferência de conhecimento.

Integração com Self-play

Após a conclusão do último estágio do curriculum, o sistema transita automaticamente para o treinamento com Self-play, onde os agentes continuam seu desenvolvimento enfrentando versões cada vez mais competentes de si mesmos. Esta integração segue o seguinte processo:

1. Os pesos da política treinada com curriculum são utilizados para inicializar tanto a política dos agentes azuis quanto dos agentes amarelos;
2. O sistema de Self-play é ativado, com atualizações periódicas da política dos agentes amarelos quando os agentes azuis atingem um desempenho superior (diferença de gols 0.6);
3. O treinamento continua por 785 iterações adicionais (aproximadamente 7 horas);
4. As métricas de desempenho são continuamente monitoradas para avaliar a eficácia da abordagem.

3.1.9 Implementação do Algoritmo

Para o treinamento dos agentes, utilizamos o algoritmo Proximal Policy Optimization (PPO), uma abordagem moderna de policy gradient que oferece estabilidade, eficiência e desempenho consistente. A implementação foi realizada utilizando o framework RLlib da biblioteca Ray, que proporciona escalabilidade e distribuição eficiente do processo de treinamento.

Estrutura da Rede Neural

A arquitetura neural utilizada consiste em:

- **Rede de política (actor)**: Rede MLP (Multi-Layer Perceptron) com camadas ocultas [300, 200, 100];
- **Função de ativação**: ReLU (Rectified Linear Unit) para todas as camadas ocultas;

- **Rede de valor (critic)**: Rede MLP com mesma estrutura, mas parâmetros independentes;
- **Camada de saída da política**: Parâmetros de distribuição Beta para cada componente da ação, proporcionando ações contínuas limitadas;
- **Camada de saída do valor**: Valor escalar representando a estimativa de retorno.

A distribuição Beta foi escolhida para modelar as ações contínuas porque naturalmente limita os valores ao intervalo [0, 1], que são posteriormente remapeados para [-1, 1] para controle dos robôs. Isso proporciona maior estabilidade durante o treinamento, comparado a outras distribuições como a Gaussiana.

Compartilhamento de Política

Uma característica importante da implementação é o compartilhamento de parâmetros entre todos os agentes do mesmo time (policy sharing). Isto significa que:

- Todos os robôs do time azul compartilham os mesmos parâmetros de rede neural;
- O time amarelo, no estágio de self-play, utiliza uma cópia (potencialmente desatualizada) da política do time azul;
- Cada agente recebe observações específicas à sua posição, mas processa-as através da mesma política.

Este compartilhamento reduz o número de parâmetros a serem aprendidos, acelera o treinamento e facilita a emergência de comportamentos coordenados.

Hiperparâmetros e Configurações

Os principais hiperparâmetros utilizados no treinamento foram:

- **Taxa de aprendizado**: 0.0004;
- **Fator de desconto (gamma)**: 0.99;
- **Parâmetro lambda para GAE (Generalized Advantage Estimation)**: 0.95;
- **Coeficiente de entropia**: 0.01 (incentiva exploração);
- **Parâmetro de clipping**: 0.2 (limita mudanças abruptas na política);
- **Batch size**: 96000 amostras (calculado como workers $\times$ envs $\times$ fragment);
- **Mini-batch size**: 24000 amostras (batch/4);
- **Épocas por atualização**: 5;
- **Workers**: 12 processos paralelos;
- **Ambientes por worker**: 4.

Paralelização e Distribuição

O treinamento foi distribuído para maximizar a eficiência computacional:

- Utilização de 12 trabalhadores paralelos, cada um gerenciando 4 ambientes;
- Coleta simultânea de experiências em 48 ambientes (12×4);
- Processamento em lote de experiências para atualização da política;
- Implementação de buffers de experiência para reduzir a correlação entre amostras.

Avaliação Durante Treinamento

Durante o treinamento, realizamos avaliações periódicas para monitorar o progresso:

- A cada 10 iterações, executamos 50 episódios de avaliação;
- Durante a avaliação, desativamos a exploração para medir o desempenho da política aprendida;
- Registramos métricas como taxa de vitória, número de gols e duração dos episódios;
- Estas métricas são visualizadas em tempo real via TensorBoard.

A implementação do algoritmo foi cuidadosamente otimizada para garantir eficiência computacional sem sacrificar a qualidade do aprendizado, permitindo treinamento efetivo mesmo com os recursos computacionais disponíveis (CPU AMD Ryzen 7 7700x com 16 núcleos virtuais, 32GB RAM, e GPU NVIDIA GeForce RTX 3090).

3.1.10 Self-Play

O treinamento por Self-play é uma técnica fundamental para desenvolver agentes competitivos em ambientes multiagente, onde a qualidade dos oponentes evolui à medida que o próprio agente melhora. Em nosso trabalho, implementamos um mecanismo de Self-play estruturado para refinar as habilidades dos agentes após a fase de Curriculum Learning.

Mecanismo de Self-play Implementado

Nossa implementação do Self-play segue os seguintes princípios:

- O time azul (agentes em treinamento ativo) enfrenta o time amarelo (oponentes);
- Os oponentes utilizam uma cópia congelada da política do time azul de uma versão anterior;
- A política do time azul é atualizada continuamente através do algoritmo PPO;

- A política do time amarelo é atualizada discretamente, apenas quando determinados critérios são atingidos.

Esta configuração cria um equilíbrio entre estabilidade e desafio: os oponentes são suficientemente estáveis para permitir aprendizado consistente, mas periodicamente atualizados para apresentar novos desafios.

Critérios de Atualização dos Oponentes

A decisão sobre quando atualizar a política dos oponentes é crucial para o sucesso do Self-play. Utilizamos um mecanismo baseado em desempenho:

- Mantemos um contador de pontuação (ScoreCounter) que monitora os resultados das últimas 100 partidas;
- Calculamos o escore médio como: $\frac{gols\_marcados - gols\_sofridos}{nmero\_de\_jogos}$;
- Quando o escore médio atinge ou ultrapassa 0.6, consideramos que a política atual é suficientemente superior;
- Neste momento, copiamos os pesos da política do time azul para o time amarelo;
- Após a atualização, resetamos o contador para reiniciar a avaliação.

O limiar de 0.6 foi cuidadosamente escolhido para garantir melhorias significativas antes da atualização, evitando oscilações prematuras que poderiam desestabilizar o treinamento.

Implementação Técnica

A lógica de Self-play foi implementada na classe `SelfPlayUpdateCallback`, que herda de `DefaultCallbacks` do RLlib. Esta classe realiza as seguintes funções:

- Rastreia o desempenho do time azul contra o time amarelo;
- Gerencia o contador de pontuação e calcula métricas relevantes;
- Executa a transferência de pesos quando os critérios são atendidos;
- Registra estatísticas para análise posterior.

A cada episódio, a callback atualiza suas estatísticas internas e verifica se os critérios de atualização foram atingidos. Todo este processo é automatizado e ocorre durante o treinamento, sem necessidade de intervenção manual.

Integração com Curriculum Learning

Na abordagem combinada proposta neste trabalho, a fase de Self-play inicia após a conclusão do curriculum. Esta integração apresenta características específicas:

- A política inicial para ambos os times (azul e amarelo) é derivada do modelo treinado no curriculum;
- O contador de pontuação é inicializado do zero no início da fase de Self-play;
- As primeiras atualizações tendem a ocorrer rapidamente, uma vez que o time azul continua melhorando sobre a base estabelecida pelo curriculum;
- À medida que o treinamento avança, as atualizações tornam-se menos frequentes, indicando convergência gradual.

Vantagens Observadas

Os experimentos demonstraram diversas vantagens do Self-play como mecanismo de refinamento após o Curriculum Learning:

- **Currículo automático**: Cria naturalmente uma progressão de desafios adaptados ao nível do agente;
- **Exploração eficiente**: Incentiva a descoberta de estratégias inovadoras para superar oponentes cada vez mais sofisticados;
- **Robustez aumentada**: Desenvolve políticas que funcionam contra diversas estratégias adversárias;
- **Emergência de comportamentos avançados**: Táticas complexas como passes, posicionamento defensivo e coordenação emergem naturalmente

As métricas coletadas durante os experimentos comprovam a eficácia desta abordagem, com o Self-play após Curriculum Learning superando significativamente o Self-play puro em todas as métricas relevantes.

3.1.11 Condições de Finalização do Episódio

A definição de quando um episódio termina é crucial para o aprendizado eficiente, pois determina o horizonte temporal com que os agentes lidam. Em nosso trabalho, um episódio é encerrado quando uma das seguintes condições é satisfeita:

- **Gol marcado**: Um gol é marcado por qualquer uma das equipes;
- **Tempo máximo**: O limite de 40 segundos (1200 passos de simulação a 30 fps) é atingido.

É importante distinguir entre a finalização de um episódio e os "resets"durante o episódio. Quando a bola sai do campo (pelas laterais ou linhas de fundo), ocorre um reset das posições dos jogadores e da bola para configurações pré-definidas, mas o episódio continua em andamento, mantendo-se o placar e o tempo decorrido. O sistema registra estas ocorrências como "resets laterais"ou "resets de linha de fundo".

3.1.12 Ambiente de Simulação

O ambiente de simulação utilizado neste trabalho é o *RL-SSL-EL*[1] na versão de implementação da equipe de robótica Pequi Mecânico, conforme ilustrado na Figura 3.2. Esta plataforma foi projetada especificamente para o futebol de robôs na categoria *Small Size League - Entry League (SSL-EL)*, simplificando o processo de aprendizado por reforço aplicado ao futebol de robôs.

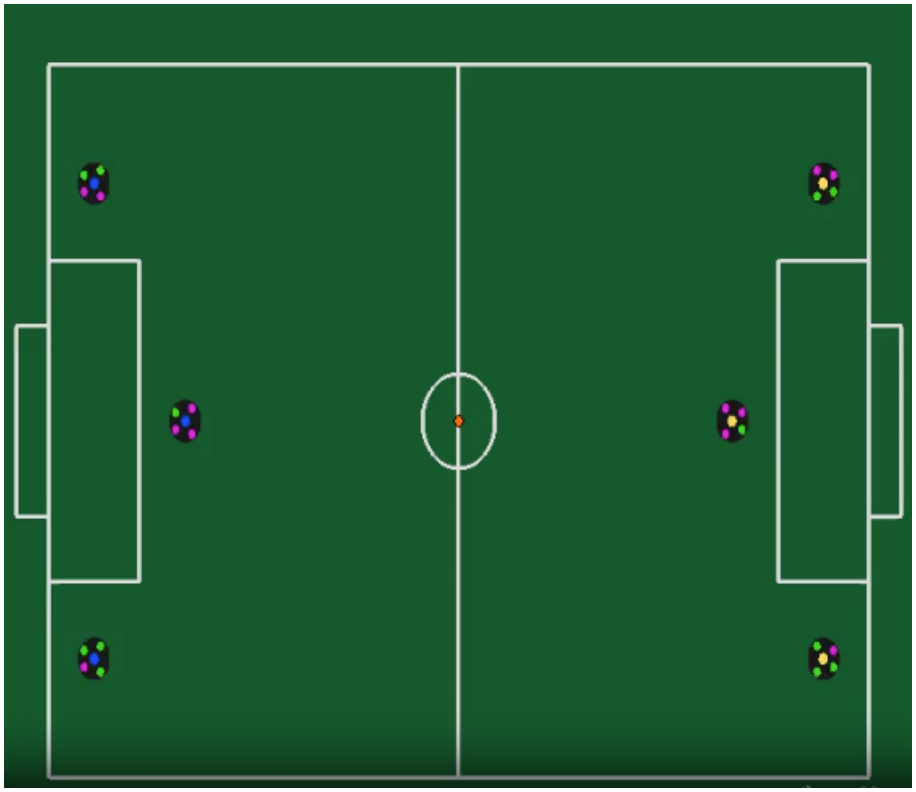

Figura 3.2: *Visualização do ambiente de simulação RL-SSL-EL mostrando o campo de jogo com os robôs e a bola*

Características Técnicas

- **Integração com RLlib**: Compatibilidade total com o *framework RLlib* para treinamento distribuído;
- **Frequência de simulação**: 30 FPS (frames por segundo);
- **Frequência de observação/ação**: 10 Hz para os agentes, representando o ciclo de percepção-ação;
- **Duração máxima do episódio**: 40 segundos (1200 passos de simulação);
- **Dimensões do campo**: Conforme especificações oficiais da categoria SSL-EL;

[1] https://github.com/Werikcyano/RL-SSL-EL

- **Modelagem dos robôs**: Baseada nas especificações físicas do Pequi Mecânico (massa, dimensões, limites de velocidade).

Modificações para Curriculum Learning

O ambiente base foi estendido para suportar os diferentes estágios do curriculum learning, com implementações específicas:

- **Classe SSLCurriculumEnv**: Estende o ambiente base com funcionalidades necessárias para o curriculum;
- **Configuração dinâmica**: Permite alteração do número de agentes, posicionamento e regras durante o treinamento;
- **Funções de recompensa adaptativas**: Diferentes sinais de recompensa para cada estágio do curriculum;
- **Sistema de monitoramento**: Métricas específicas para avaliar o progresso em cada estágio.

Visualização

O ambiente inclui um módulo de visualização baseado em OpenGL que permite a renderização do campo, robôs e bola para inspeção visual durante o treinamento e avaliação. Esta visualização pode ser ativada através da flag `-evaluation` durante o treinamento.

3.1.13 Configuração Computacional

Os experimentos foram realizados em uma máquina com a seguinte configuração:

- **CPU**: AMD Ryzen 7 7700x com 16 núcleos virtuais;
- **RAM**: 32GB DDR5;
- **GPU**: NVIDIA GeForce RTX 3090 com 24GB de VRAM;
- **Armazenamento**: SSD NVMe de 2TB;
- **Sistema Operacional**: Ubuntu 22.04 LTS.

Para reprodução dos experimentos, recomenda-se uma configuração mínima de:

- CPU com pelo menos 8 núcleos lógicos;
- 16GB de RAM;
- GPU com suporte a CUDA e pelo menos 8GB de VRAM;
- 100GB de espaço em disco para armazenar checkpoints e logs.

Paralelização e Tempo de Treinamento

O treinamento distribuído foi configurado com:

- 12 workers em paralelo;
- 4 ambientes por worker;
- Total de 48 ambientes simultâneos.

Com esta configuração, o tempo total de treinamento foi aproximadamente:

- **Curriculum + Self-play (abordagem proposta)**: total de 900 iterações
  - **Curriculum Task 0**: 20 minutos (55 iterações);
  - **Curriculum Task 1**: 22 minutos (60 iterações);
  - **Self-play após Curriculum**: 7 horas (785 iterações);
  - **Tempo total**: 7.7 horas (900 iterações).
- **Full Self-play (baseline)**: 8.7 horas (900 iterações)

É importante ressaltar que ambas as abordagens utilizaram o mesmo critério de parada: 900 iterações totais de treinamento. Esta equivalência garante uma comparação justa entre os métodos, evidenciando que a abordagem de Curriculum + Self-play fez em menos tempo (7.7 horas versus 8.7 horas).

Todas as configurações específicas, scripts auxiliares e documentação detalhada estão disponíveis no repositório do projeto. Para facilitar a reprodução, fornecemos também um container Docker com todas as dependências pré-configuradas.

3.1.14 Visão Geral da Arquitetura do *Curriculum*

A arquitetura do *curriculum learning* implementada neste trabalho segue uma estrutura de estágios progressivos, onde cada estágio representa um nível de complexidade específico no aprendizado do futebol de robôs. A progressão entre estes estágios é controlada por critérios de desempenho predefinidos, garantindo que os agentes desenvolvam as habilidades necessárias antes de avançar para desafios mais complexos.

A transição do cenário curricular é projetada para preservar a continuidade do aprendizado, evitando mudanças abruptas que poderiam prejudicar o desenvolvimento dos agentes. Esta arquitetura foi inspirada em sistemas de treinamento progressivo observados em jogos eletrônicos populares, como FIFA e *Rocket League*, onde os jogadores são introduzidos gradualmente a conceitos mais complexos. A Figura 3.3 ilustra o fluxo completo do processo de treinamento com *curriculum learning*, destacando as etapas e transições entre os diferentes estágios do aprendizado.

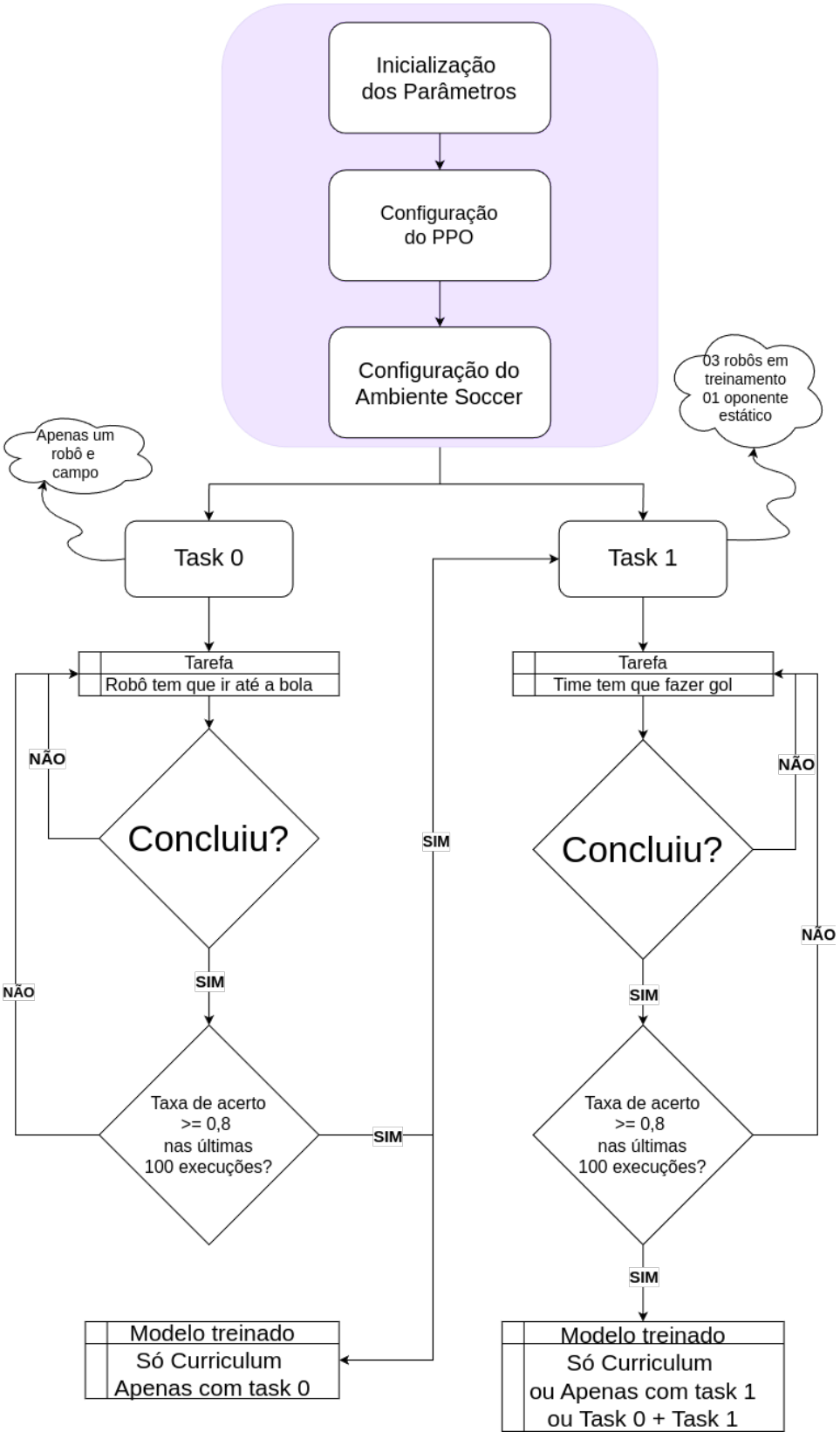

Figura 3.3: *Diagrama de fluxo do processo de treinamento com curriculum learning. Fonte: Elaborado pelo autor.*

O fluxo do processo inicia com a definição dos estágios e seus respectivos

parâmetros no arquivo de configuração `config.yaml`. Estes parâmetros incluem critérios de sucesso, sistemas de recompensa específicos, e configurações do ambiente para cada estágio. Durante o treinamento, o `CurriculumCallback` monitora continuamente o desempenho dos agentes, calculando a taxa de sucesso com base em uma janela deslizante de episódios recentes. Quando esta taxa atinge o limiar de promoção predefinido, o sistema avança automaticamente para o próximo estágio do *curriculum*.

3.2 Métricas de Avaliação

Para avaliar o desempenho dos agentes e comparar a eficácia das abordagens de treinamento, foi desenvolvido um conjunto abrangente de métricas que captura diferentes aspectos do comportamento dos agentes. Estas métricas são coletadas durante o treinamento e utilizadas para análises comparativas.

3.2.1 Tempo dos Episódios

O tempo dos episódios é uma métrica importante para avaliar a eficiência do jogo e a capacidade dos agentes de alcançar seus objetivos rapidamente. Esta métrica é analisada em diferentes dimensões:

- Duração média dos episódios (em passos);
- Evolução da duração ao longo do treinamento;
- Distribuição dos tempos de episódio.

Um aspecto particularmente relevante desta métrica é sua relação com a progressão do treinamento. Tipicamente, espera-se que episódios mais curtos indiquem agentes mais eficientes na realização de seus objetivos.

3.2.2 Métrica de Continuidade

As métricas de continuidade foram desenvolvidas especificamente para este trabalho, visando avaliar a fluidez do jogo e a capacidade dos agentes de manter a bola em jogo por períodos prolongados. Estas métricas incluem:

- Número total de *resets* durante todo o treinamento;
- Média de *resets* por episódio.

Estas métricas são particularmente importantes para avaliar a qualidade do jogo produzido pelos agentes, uma vez que um jogo com menos interrupções tende a ser mais dinâmico e interessante.

3.2.3 Recompensa Acumulada

A recompensa acumulada representa a métrica fundamental do aprendizado por reforço, refletindo diretamente o objetivo de otimização dos agentes. Esta métrica é analisada em várias dimensões:

- Recompensa média por episódio;
- Evolução da recompensa ao longo do treinamento.

A análise da recompensa acumulada permite avaliar a convergência do treinamento e comparar diretamente diferentes abordagens em termos de sua eficácia em otimizar o comportamento dos agentes.

3.2.4 Avaliação em Torneio

Para uma avaliação mais abrangente e objetiva do desempenho dos modelos, será realizado um torneio competitivo entre o modelo *baseline* (treinado com *self-play* padrão) e o modelo proposto (incorporando *curriculum learning*). Este torneio permitirá:

- Comparação direta do desempenho em condições controladas;
- Avaliação da robustez das estratégias aprendidas;
- Análise da consistência dos resultados em múltiplas partidas;
- Identificação de possíveis vantagens táticas específicas.

O formato do torneio será estruturado para garantir uma avaliação estatisticamente significativa, com múltiplas partidas entre os modelos. Os resultados deste torneio fornecerão evidências importantes sobre a eficácia prática das modificações propostas no processo de treinamento.

Além das métricas específicas descritas anteriormente, também serão utilizadas algumas métricas padrão do aprendizado por reforço:

- *Entropy* - Mede a aleatoriedade das ações selecionadas pela política, indicando o nível de exploração do agente;
- *Policy Loss* - Quantifica o erro na política atual em relação à política ótima estimada;
- *VF Explained* - Indica quanto da variância nas recompensas é explicada pelo modelo de valor, medindo a qualidade das estimativas do valor.

CAPÍTULO 4

Experimentos e Resultados

4.1 Configuração Experimental

Os experimentos realizados neste trabalho buscaram avaliar a eficácia da abordagem proposta, que combina *curriculum learning* e *self-play* para o treinamento de agentes no ambiente de futebol de robôs. Para garantir uma avaliação consistente e comparativa, foi estabelecida uma configuração experimental padronizada, detalhada nesta seção.

4.1.1 Hardware Utilizado

Todos os experimentos foram executados em um ambiente computacional de alto desempenho, com especificações adequadas para treinamento de aprendizado por reforço distribuído. A configuração de *hardware* incluiu:

- Processador: AMD Ryzen 7 7700x com 16 núcleos virtuais
- Memória RAM: 32GB DDR5
- GPU: NVIDIA GeForce RTX 3090 com 24GB de memória VRAM
- Armazenamento: SSD NVMe de 2TB para leitura e escrita rápidas de *checkpoints*

A utilização de *hardware* especializado foi fundamental para viabilizar o treinamento distribuído com múltiplos trabalhadores paralelos, acelerando significativamente o processo experimental.

4.1.2 Parâmetros de Treinamento

Os parâmetros de treinamento foram cuidadosamente selecionados para garantir um equilíbrio entre eficiência computacional e qualidade do aprendizado, tentando mater o mais próximo possível do padrão utilizado no artigo original [Brandão et al. 2022]. Os principais parâmetros utilizados incluem:

- **Algoritmo**: *Proximal Policy Optimization* (PPO)

- **Taxa de aprendizado**: 0.0004 - Define o tamanho dos passos durante a otimização, controlando a velocidade do aprendizado
- **Função de ativação**: *ReLU* - Função não-linear que permite à rede neural aprender padrões complexos, zerando valores negativos
- **Arquitetura da rede neural**: *Fully Connected* com camadas [300, 200, 100] - Estrutura da rede neural com três camadas ocultas totalmente conectadas
- **Batch size**: 96000 (calculado como *workers* x *envs* x *fragment*) - Quantidade total de amostras processadas em cada iteração de treinamento
- **Mini-batch size**: 24000 (*batch*/4) - Subdivisão do *batch* para processamento em lotes menores, otimizando o uso de memória
- **Número de workers**: 12 (ambientes paralelos) - Quantidade de processos paralelos executando o ambiente de simulação
- **Ambientes por workers**: 4 - Número de ambientes simultâneos gerenciados por cada *worker*
- **Gamma**: 0.99 - Fator de desconto que determina a importância de recompensas futuras
- **Lambda**: 0.95 - Parâmetro de equilíbrio entre viés e variância no cálculo da vantagem generalizada
- **Coeficiente de entropia**: 0.01 - Incentiva a exploração ao adicionar aleatoriedade na política
- **Clip param**: 0.2 - Limita o tamanho das atualizações da política para evitar mudanças muito grandes
- **Iterações SGD**: 8 - Número de passos de otimização realizados em cada *batch* de dados

No caso específico do *curriculum learning*, foram configurados dois estágios progressivos, com parâmetros específicos para cada um, conforme detalhado no Capítulo 3. A taxa de promoção entre estágios foi estabelecida em 80% de sucesso em uma janela de 100 episódios.

Para todos os experimentos realizados, foi definido um limite máximo de 900 iterações de treinamento, proporcionando uma base comparativa consistente entre as diferentes abordagens. No entanto, o número efetivo de iterações variou conforme o tipo de experimento e a complexidade das tarefas envolvidas, como pode ser observado na Tabela 4.1.

Esta distribuição de iterações evidencia a diferença de complexidade entre as fases do treinamento, com as tarefas iniciais do *curriculum* exigindo significativamente menos iterações para convergência em comparação com as fases de *self-play*.

Tabela 4.1: *Número de iterações por experimento*

Experimento	Total de Iterações
Curriculum Task 0	55
Curriculum Task 1	60
Selfplay Após *Curriculum*	785
Full Selfplay	900

4.1.3 Tempo de Treinamento

O tempo total de treinamento para cada experimento foi medido em termos de horas de execução na configuração de *hardware* utilizada. A Figura 4.1 apresenta uma comparação dos tempos de treinamento para as diferentes abordagens implementadas.

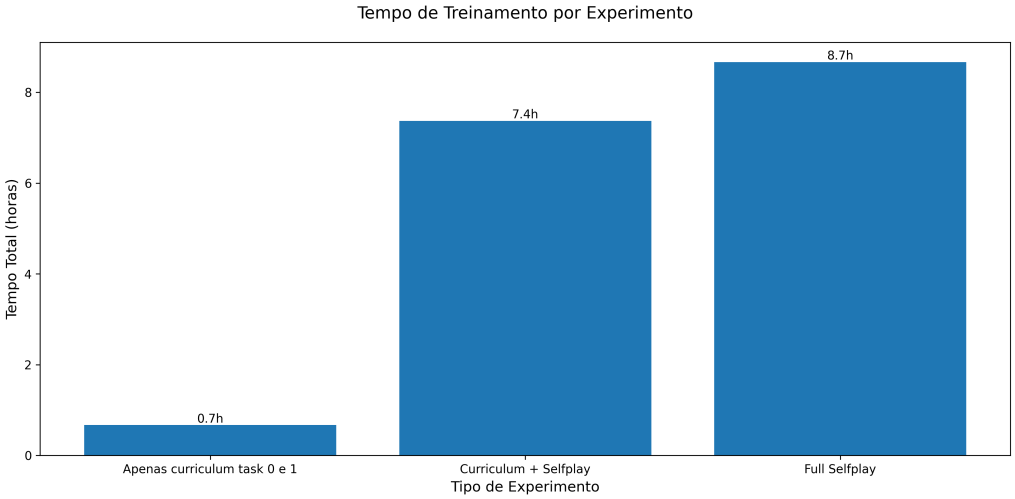

Figura 4.1: *Comparação do tempo total de treinamento em horas para cada abordagem experimental*

A análise dos tempos de treinamento revela diferenças significativas entre as abordagens:

- **Apenas *curriculum* (*tasks* 0 e 1)**: Aproximadamente 0,7 horas (42 minutos)
- ***Curriculum* + *Self-play***: Aproximadamente 7,4 horas
- ***Full Self-play***: Aproximadamente 8,7 horas

Estas diferenças nos tempos de treinamento refletem a complexidade e a natureza de cada abordagem. O treinamento exclusivo com *curriculum tasks* é significativamente mais rápido, pois envolve ambientes mais simples e objetivos bem definidos. A abordagem combinada (*curriculum* + *self-play*) apresenta um tempo de treinamento aproximadamente 15% menor em relação ao *full self-play* tradicional.

Esta economia de tempo é uma observação particularmente relevante do ponto de vista prático, especialmente considerando que a abordagem combinada também propor-

cionou resultados superiores (conforme demonstrado nas seções posteriores). Esta vantagem em termos de eficiência computacional representa um aspecto importante para aplicações práticas, onde os recursos de processamento são frequentemente limitados.

A maior eficiência da abordagem combinada pode ser atribuída ao fato de que as habilidades fundamentais desenvolvidas durante o *curriculum* permitem que o agente aproveite melhor a fase de *self-play*, convergindo mais rapidamente para políticas efetivas. Os agentes que iniciam diretamente no *self-play* gastam mais tempo em fases iniciais de exploração aleatória, necessitando de iterações adicionais para desenvolver as mesmas capacidades que são adquiridas de forma estruturada no *curriculum*.

4.2 Análise Comparativa

Para avaliar a eficácia da abordagem proposta, realizamos uma análise comparativa entre o modelo treinado apenas com *self-play* (*baseline*) e o modelo treinado com a combinação de *curriculum learning* e *self-play* (modelo proposto). Esta análise abrange diversas métricas relevantes para o domínio do futebol de robôs.

4.2.1 Evolução da Recompensa

A análise da evolução da recompensa média ao longo do treinamento oferece *insights* valiosos sobre o processo de aprendizagem dos agentes. A Figura 4.2 apresenta esta evolução para ambas as abordagens.

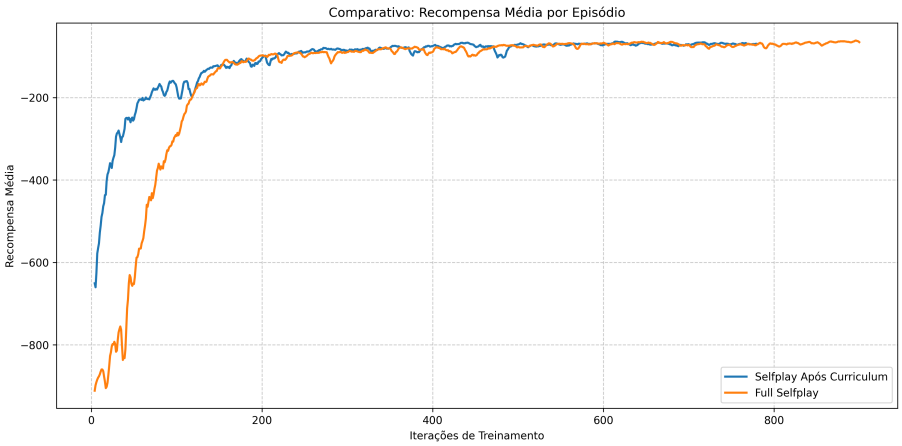

Figura 4.2: *Evolução da recompensa média por episódio ao longo do treinamento*

O gráfico revela que o modelo treinado com *curriculum learning* apresenta um crescimento mais acelerado da recompensa nas fases iniciais do treinamento. Esta vantagem inicial é atribuída à aprendizagem estruturada de habilidades fundamentais durante

os estágios do *curriculum*. Embora ambas as abordagens apresentem convergência em termos de recompensa acumulada, o modelo proposto atinge níveis equivalentes com menos *timesteps* de treinamento, sugerindo maior eficiência no processo de aprendizagem.

Nota-se também que o modelo proposto apresenta menor variabilidade na curva de recompensa, indicando maior estabilidade durante o processo de treinamento.

Para compreender melhor como as habilidades fundamentais são desenvolvidas durante o *curriculum*, é interessante analisar a evolução da recompensa em cada estágio específico. As Figuras 4.3 e 4.4 apresentam esta evolução para os estágios *Task* 0 e *Task* 1, respectivamente.

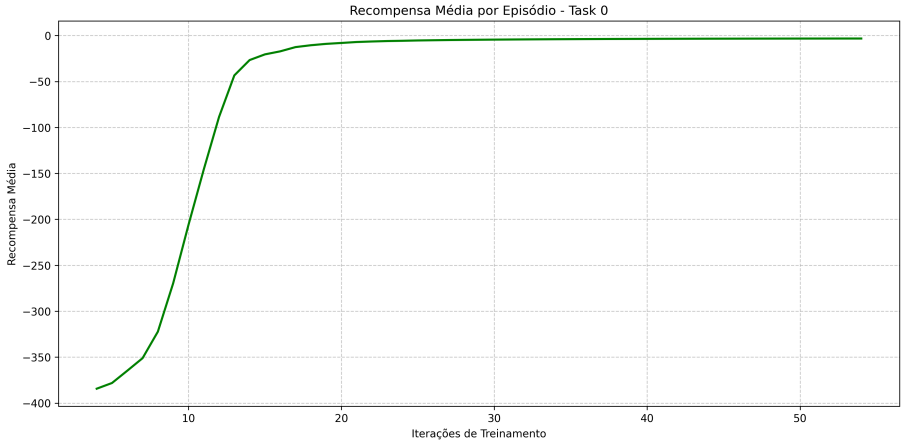

Figura 4.3: *Evolução da recompensa média por episódio durante o Curriculum Task 0*

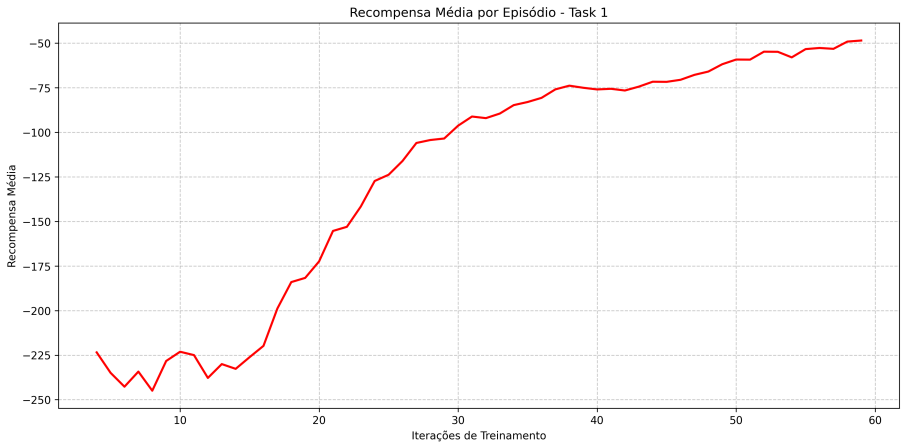

Figura 4.4: *Evolução da recompensa média por episódio durante o Curriculum Task 1*

Estes gráficos revelam padrões distintos de aprendizado em cada estágio do *curriculum*. No *Task* 0 (Figura 4.3), observa-se um crescimento acentuado da recompensa nas primeiras 15 iterações, partindo de valores próximos a -400 e estabilizando rapidamente

em torno de 0. Esta evolução demonstra que os agentes aprendem eficientemente as habilidades básicas de controle e aproximação da bola, atingindo o critério de promoção em poucas iterações.

Já no *Task* 1 (Figura 4.4), nota-se um padrão de aprendizado mais gradual e complexo. A curva apresenta oscilações iniciais entre as iterações 5 e 15, seguidas por um crescimento consistente até a iteração 60, quando a recompensa média atinge aproximadamente -50. Este comportamento reflete a maior complexidade desta tarefa, que exige coordenação entre múltiplos agentes e estratégias ofensivas mais sofisticadas na presença de oponentes estáticos.

A comparação entre estes dois estágios ilustra claramente a progressão de complexidade no *curriculum* e como os agentes desenvolvem diferentes habilidades em cada fase. O rápido progresso no *Task* 0 estabelece a base motora fundamental, enquanto o aprendizado mais gradual no *Task* 1 desenvolve capacidades táticas e coordenativas que serão cruciais durante o posterior treinamento com *self-play*.

4.2.2 Desempenho Ofensivo

O desempenho ofensivo dos agentes foi avaliado principalmente através da análise da taxa de sucesso na conclusão dos objetivos. A Figura 4.5 apresenta a evolução desta métrica ao longo do treinamento para ambas as abordagens.

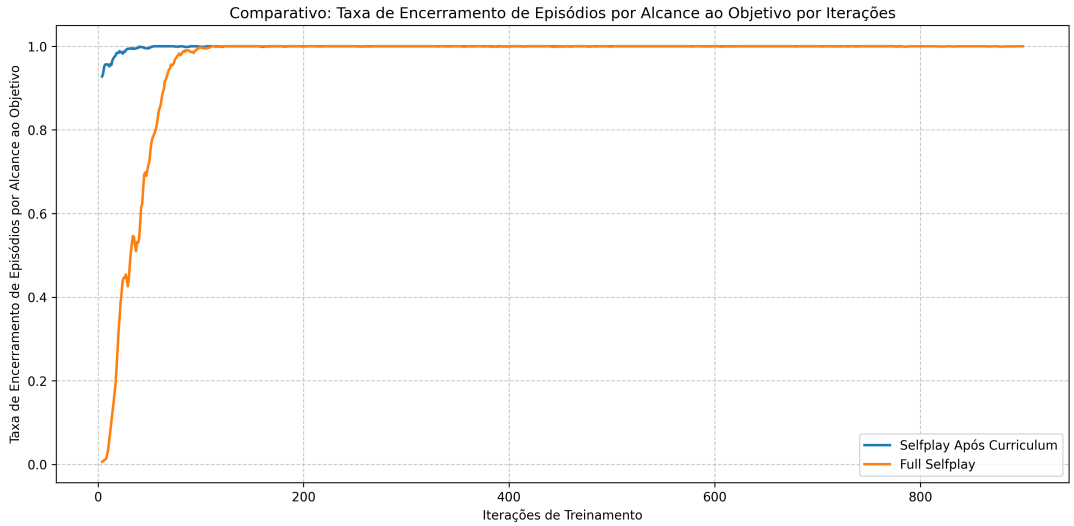

Figura 4.5: *Comparativo: Taxa de Encerramento de Episódios por Alcance ao Objetivo por Iterações entre as abordagens Selfplay após Curriculum e Full Selfplay*

A análise comparativa do gráfico revela padrões interessantes na evolução da capacidade dos agentes de atingir seus objetivos. Ambas as abordagens apresentam uma

progressão crescente na taxa de sucesso, eventualmente atingindo valores próximos a 100%. No entanto, observam-se diferenças significativas no processo de aprendizado:

- **Velocidade de convergência**: O *Selfplay* após *Curriculum* (linha azul) atinge o patamar próximo a 100% de sucesso muito mais rapidamente, convergindo nas primeiras 50 iterações, enquanto o *Full Selfplay* (linha laranja) requer cerca de 150 iterações para alcançar desempenho semelhante.
- **Fase inicial**: Nas primeiras iterações, o *Selfplay* após *Curriculum* já começa com uma taxa de sucesso em torno de 90%, demonstrando que as habilidades adquiridas durante a fase de *curriculum* proporcionam um ponto de partida significativamente mais avançado.
- **Estabilidade**: Ambas as abordagens eventualmente atingem estabilidade, mas o *Selfplay* após *Curriculum* apresenta menor variabilidade durante todo o processo, indicando um aprendizado mais consistente e robusto.

Esta comparação com iterações alinhadas evidencia de forma clara uma vantagem significativa da abordagem proposta: ao iniciar o *self-play* com agentes já treinados em tarefas fundamentais através do *curriculum*, obtém-se uma aceleração substancial na capacidade de atingir objetivos. Esta característica é particularmente valiosa em cenários com restrições de tempo computacional, onde a convergência mais rápida para políticas de alta qualidade representa uma vantagem considerável.

4.2.3 Eficiência e Continuidade do Jogo

No contexto deste trabalho, um *reset* representa o reposicionamento dos robôs às suas posições iniciais, ocorrendo sempre que a bola sai do campo. Este reposicionamento é necessário para garantir a continuidade da partida.

Um aspecto diferencial da abordagem proposta é a melhoria significativa nas métricas relacionadas à continuidade do jogo, que refletem a capacidade dos agentes de manter a bola em jogo por períodos mais longos. A Figura 4.6 apresenta a evolução do número médio de *resets* por episódio.

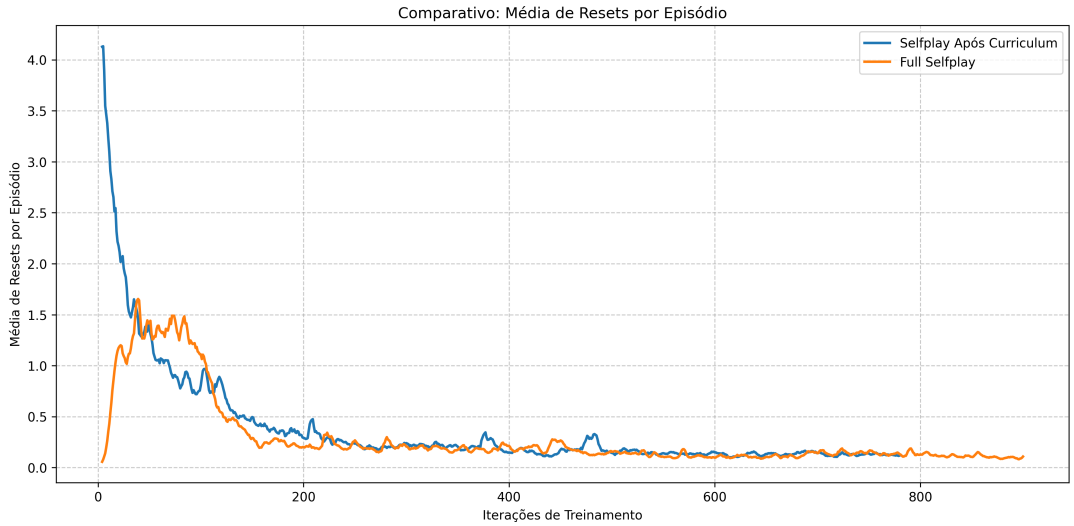

Figura 4.6: *Comparativo da média de resets por episódio: Selfplay após Curriculum e Full Selfplay*

A análise do gráfico revela diferenças significativas nos padrões de aprendizado relacionados à continuidade do jogo. Inicialmente, o *Selfplay* após *Curriculum* (linha azul) apresenta um pico maior de *resets* por episódio, mas rapidamente consegue reduzir esse número. O *Full Selfplay* (linha laranja) mostra um comportamento diferente, com um aumento gradual seguido por uma redução mais lenta.

Após aproximadamente 200 iterações, ambas as abordagens convergem para valores similares, com ligeira vantagem para o *Full Selfplay* nas iterações finais. No entanto, é notável que o *Selfplay* após *Curriculum* consegue reduzir o número de *resets* de forma mais rápida nas fases iniciais do treinamento, evidenciando a transferência positiva das habilidades adquiridas durante o *curriculum*.

Esta análise demonstra que, embora ambas as abordagens eventualmente atinjam desempenhos similares em termos de continuidade do jogo em suas fases finais, o *Selfplay* após *Curriculum* oferece um processo de aprendizado mais eficiente e estável, especialmente durante as fases iniciais e intermediárias do treinamento.

4.2.4 Duração dos Episódios

A análise da duração média dos episódios ao longo do treinamento (Figura 4.7) revela padrões interessantes sobre a evolução das estratégias desenvolvidas pelos agentes.

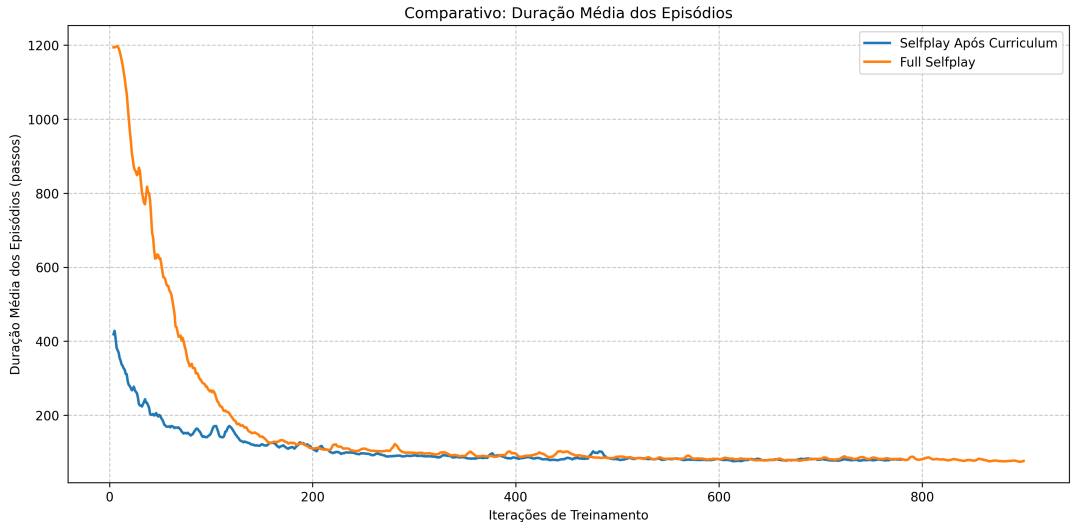

Figura 4.7: *Comparativo da duração média dos episódios: Selfplay após Curriculum e Full Selfplay*

A análise do gráfico revela diferenças marcantes no comportamento dos agentes em relação à duração dos episódios. O *Full Selfplay* (linha laranja) inicia com episódios significativamente mais longos, atingindo aproximadamente 1200 passos nas primeiras iterações, enquanto o *Selfplay* após *Curriculum* (linha azul) começa com episódios bem mais curtos, em torno de 400 passos. Esta diferença inicial demonstra que as habilidades adquiridas durante as fases de *curriculum* proporcionam um ponto de partida mais eficiente, permitindo aos agentes atingir seus objetivos em menos passos desde o início.

Ao longo do treinamento, ambas as abordagens apresentam uma redução gradual na duração dos episódios, convergindo para valores similares após aproximadamente 200 iterações, quando estabilizam em torno de 100 passos por episódio. Esta redução na duração dos episódios é um indicador positivo, pois demonstra que os agentes estão se tornando mais eficientes em marcar gols, já que cada episódio é encerrado quando um gol é marcado. É notável que o *Selfplay* após *Curriculum* apresenta uma curva de aprendizado mais suave e consistente, sem as oscilações pronunciadas observadas no *Full Selfplay*, indicando um processo de aprendizagem mais estável e uma evolução mais consistente na capacidade de finalização. Nas iterações finais, ambas as abordagens mantêm desempenho similar em termos de velocidade para marcar gols, mas o caminho percorrido pelo *Selfplay* após *Curriculum* para atingir este patamar demonstra maior eficiência, especialmente nas fases críticas iniciais e intermediárias do treinamento.

4.2.5 Avaliação por Torneios

Para uma avaliação mais abrangente e realista do desempenho dos modelos treinados, foram realizados torneios controlados com 500 partidas cada, utilizando o sistema *Arena Serra Dourada*, implementado especificamente para este trabalho. Este

sistema permitiu a realização de partidas completas com 10 minutos de duração entre agentes treinados por diferentes métodos.

Os torneios foram organizados em dois confrontos distintos:

- **Torneio 1**: *Full Selfplay* vs *Curriculum* - 500 partidas entre agentes treinados apenas com *self-play* e agentes treinados apenas com *curriculum learning*.
- **Torneio 2**: *Full Selfplay* vs *Curriculum* + *Selfplay* - 500 partidas entre agentes treinados apenas com *self-play* e agentes treinados com a abordagem combinada (*curriculum* seguido por *self-play*).

Cada partida tinha duração de 10 minutos e seguia as regras básicas semelhantes a do futebol de robôs, como contabilização de gols e *resets*. Esta configuração experimental permitiu avaliar não apenas a eficácia em marcar gols, mas também a estabilidade das políticas aprendidas em jogos completos e a capacidade de adaptação a diferentes situações de jogo.

A Tabela 4.2 apresenta um resumo dos resultados dos dois torneios realizados.

Torneio	Partidas	Equipe	Vitórias	Gols	Empates
Torneio 1	500	*Full Selfplay*	247 (49,4%)	503	220 (44%)
		Curriculum	33 (6,6%)	58	
Torneio 2	500	*Full Selfplay*	7 (1,4%)	9	63 (12,6%)
		Curriculum + *Selfplay*	430 (86%)	1012	

Tabela 4.2: *Resumo dos resultados dos torneios realizados com 500 partidas cada*

Os resultados revelam padrões interessantes sobre o desempenho de cada abordagem. No Torneio 1 (*Full Selfplay* vs *Curriculum*), observamos uma clara superioridade do modelo treinado apenas com *self-play*, que obteve 247 vitórias (49,4%) contra apenas 33 vitórias (6,6%) do modelo treinado somente com *curriculum learning*. É notável também o alto número de empates (220, correspondendo a 44% dos jogos), sugerindo que o modelo de *curriculum*, apesar de seu desempenho inferior em termos de vitórias, desenvolveu capacidades defensivas significativas.

No Torneio 2 (*Full Selfplay* vs *Curriculum* + *Selfplay*), os resultados evidenciam a acentuada superioridade do modelo treinado com a combinação de *curriculum learning* e *self-play*, que obteve 430 vitórias (86%) em comparação com apenas 7 vitórias (1,4%) do modelo *full self-play*. O número de empates foi significativamente menor (63, apenas 12,6% dos jogos), indicando partidas mais decisivas e menos equilibradas.

Esta diferença acentuada no desempenho destaca o poderoso resultado obtido ao combinar as duas abordagens. Enquanto o *curriculum learning* isolado apresenta

limitações significativas em um cenário competitivo completo, sua integração com o *self-play* resulta em políticas substancialmente mais eficazes do que aquelas desenvolvidas apenas com *self-play*.

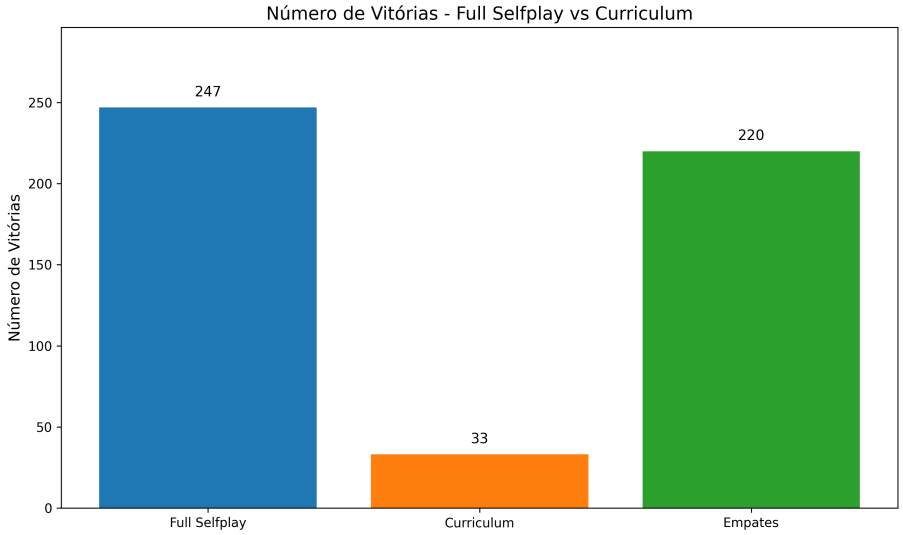

Figura 4.8: *Distribuição de resultados no Torneio 1: Full Selfplay vs Curriculum*

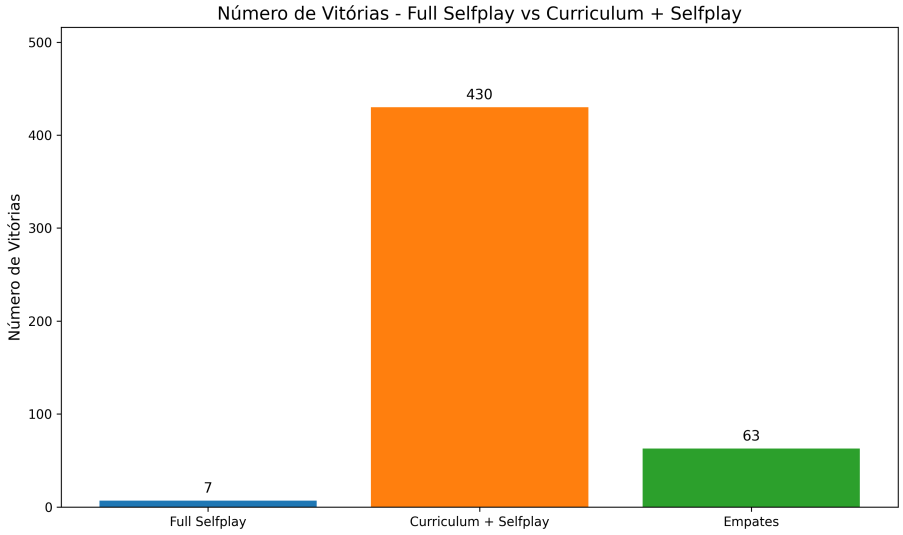

Figura 4.9: *Distribuição de resultados no Torneio 2: Full Selfplay vs Curriculum + Selfplay*

A comparação visual das Figuras 4.8 e 4.9 evidencia claramente a inversão de desempenho entre os torneios e o impacto transformador da abordagem combinada.

4.2.6 Análise de Gols nos Torneios

Além da taxa de vitória, analisamos também o desempenho ofensivo dos modelos nos torneios realizados, com foco na capacidade de marcar gols, que representa um

indicador fundamental de eficácia no futebol de robôs.

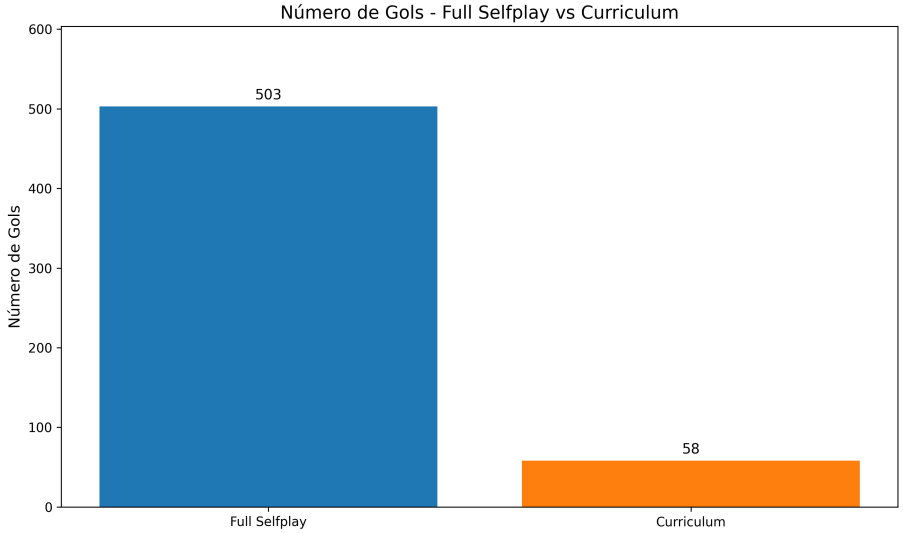

Figura 4.10: *Comparação de gols marcados no Torneio 1: Full Selfplay vs Curriculum*

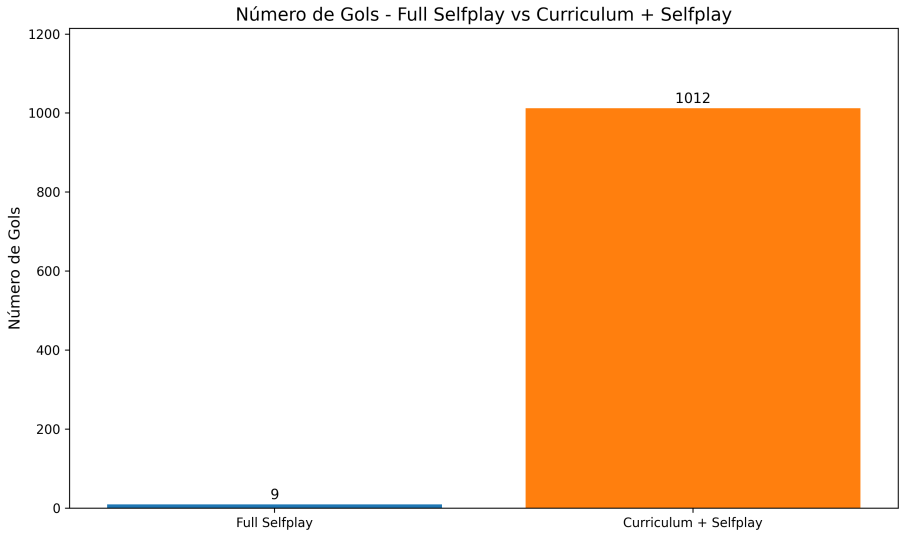

Figura 4.11: *Comparação de gols marcados no Torneio 2: Full Selfplay vs Curriculum + Selfplay*

A análise dos gols marcados em cada torneio revela padrões consistentes com os resultados gerais das partidas, mas com diferenças de magnitude ainda mais pronunciadas.

No Torneio 1 (*Full Selfplay* vs *Curriculum*), o modelo treinado apenas com *self-play* demonstrou capacidade ofensiva significativamente superior, marcando 503 gols durante as 500 partidas, o que corresponde a uma média de 1,006 gols por partida. Em contraste, o modelo treinado exclusivamente com *curriculum learning* marcou apenas 58 gols (média de 0,116 por partida), aproximadamente 11,5% do desempenho ofensivo do *Full Selfplay*.

Esta disparidade sugere que, embora o *curriculum learning* desenvolva habilidades fundamentais, ele não é suficiente para criar agentes com capacidade ofensiva efetiva em um cenário competitivo completo. A limitação ofensiva explica o alto número de empates e a baixa taxa de vitórias do modelo *Curriculum*.

No Torneio 2 (*Full Selfplay* vs *Curriculum + Selfplay*), observamos uma inversão dramática deste padrão. O modelo combinado (*curriculum + self-play*) apresentou uma capacidade ofensiva diferenciada, marcando 1.012 gols durante as 500 partidas, o que corresponde a uma impressionante média de 2,024 gols por partida. Em contraste, o modelo *Full Selfplay* marcou apenas 9 gols (média de 0,018 por partida), menos de 1% do desempenho ofensivo do modelo combinado (9 gols em 500 partidas).

Esta grande disparidade na capacidade ofensiva pode ser atribuída ao resultado sinérgico da abordagem combinada. O *curriculum learning* proporciona uma base sólida de habilidades fundamentais que, quando refinadas através do *self-play*, resultam em estratégias ofensivas excepcionalmente eficazes. O contraste entre o desempenho do *Full Selfplay* nos dois torneios sugere que o modelo combinado não apenas desenvolve políticas superiores, mas também consegue neutralizar efetivamente as estratégias do *Full Selfplay*, limitando drasticamente sua capacidade ofensiva.

4.2.7 Análise de Trade-offs entre Abordagens

Uma observação importante que surge da análise dos dados dos torneios é o claro *trade-off* entre as diferentes abordagens de treinamento. A Tabela 4.3 resume as principais métricas para cada abordagem nos confrontos diretos, facilitando a visualização destes *trade-offs*.

Métrica	Curriculum	Curriculum + Self-play
Vitórias vs Full Self-play	33 (6,6%)	430 (86%)
Empates vs Full Self-play	220 (44%)	63 (12,6%)
Derrotas vs Full Self-play	247 (49,4%)	7 (1,4%)
Gols marcados vs Full Self-play	58	1.012
Gols sofridos vs Full Self-play	503	9
Média de gols/partida	0,116	2,024

Tabela 4.3: *Comparação detalhada entre as abordagens de treinamento nos torneios*

A análise desta tabela revela padrões claros que caracterizam cada abordagem:

1. **Curriculum Learning puro**: Desenvolve agentes com capacidades defensivas significativas, evidenciadas pelo alto número de empates (44%) mesmo contra o

Full Self-play. No entanto, apresenta limitações severas na capacidade ofensiva, com apenas 0,116 gols por partida e baixa taxa de vitórias (6,6%). Sua estratégia parece focada na neutralização do adversário, mas com dificuldades para criar situações ofensivas efetivas.

2. **Full Self-play**: Produz agentes com capacidades ofensivas e defensivas moderadamente equilibradas, conseguindo dominar o *Curriculum* puro (49,4% de vitórias), mas sendo completamente superado pela abordagem combinada (apenas 1,4% de vitórias). Esta abordagem baseia-se na exploração auto-dirigida do espaço de estados, resultando em políticas funcionais, mas subótimas.
3. **Curriculum + Self-play**: Representa uma síntese poderosa que maximiza os benefícios de ambas as abordagens anteriores. Demonstra capacidade ofensiva extraordinária (2,024 gols por partida) combinada com defesa quase impenetrável (apenas 9 gols sofridos em 500 partidas). O resultado é uma dominância esmagadora sobre o *Full Self-play*, com 86% de vitórias e mais de 100 vezes mais gols marcados.

Estes resultados confirmam a hipótese central deste trabalho: o *curriculum learning* proporciona fundamentos sólidos que, quando refinados através do *self-play* competitivo, resultam em políticas significativamente superiores às desenvolvidas por qualquer uma das abordagens isoladamente.

A análise dos torneios revela aspectos particularmente interessantes sobre a transferência de conhecimento entre fases de treinamento. O modelo *Curriculum* puro, embora limitado ofensivamente, demonstra capacidades defensivas substanciais, como evidenciado pelo alto número de empates contra o *Full Self-play*. Quando estas capacidades defensivas são combinadas com o refinamento tático proporcionado pelo *self-play*, o resultado é um agente com defesa sólida e ataque extremamente eficaz.

Um aspecto notável é a diferença no desempenho do *Full Self-play* entre os dois torneios. Contra o *Curriculum* puro, ele demonstra dominância clara, mas contra o *Curriculum + Self-play*, seu desempenho colapsa. Isto sugere que a abordagem combinada não apenas desenvolve políticas eficazes, mas também consegue neutralizar estrategicamente as políticas aprendidas pelo *Full Self-play*, explorando suas vulnerabilidades de forma sistemática.

Em síntese, a abordagem combinada *Curriculum + Self-play* demonstra o melhor equilíbrio entre capacidades ofensivas e defensivas, com uma superioridade que transcende a simples soma das vantagens individuais de cada método. Este efeito sinérgico valida a importância do desenvolvimento estruturado de habilidades fundamentais antes da exposição a cenários competitivos complexos, estabelecendo um paradigma promissor para o treinamento de agentes em ambientes multiagente como o futebol de robôs.

Para complementar a análise quantitativa apresentada, foram registradas gravações ilustrativas de algumas partidas dos torneios realizados. Estas gravações servem

como demonstração visual das diferentes estratégias e comportamentos emergentes discutidos anteriormente, e podem ser encontradas online[1].

4.3 Análise das Métricas de Aprendizado por Reforço

Além das métricas específicas do domínio do futebol de robôs, uma análise detalhada das métricas básicas de aprendizado por reforço fornece *insights* valiosos sobre os processos internos dos algoritmos durante o treinamento. Esta seção explora três métricas fundamentais: entropia da política, perda da política e variância explicada da função valor, comparando o comportamento dessas métricas entre as abordagens *Selfplay* após *Curriculum* e *Full Selfplay*.

4.3.1 Entropia da Política

A entropia da política é uma métrica que quantifica o grau de aleatoriedade ou exploração nas decisões do agente. Valores mais altos (menos negativos) indicam maior exploração, enquanto valores mais baixos (mais negativos) sugerem maior certeza nas ações escolhidas. A Figura 4.12 apresenta a comparação da entropia da política entre as duas abordagens ao longo do treinamento.

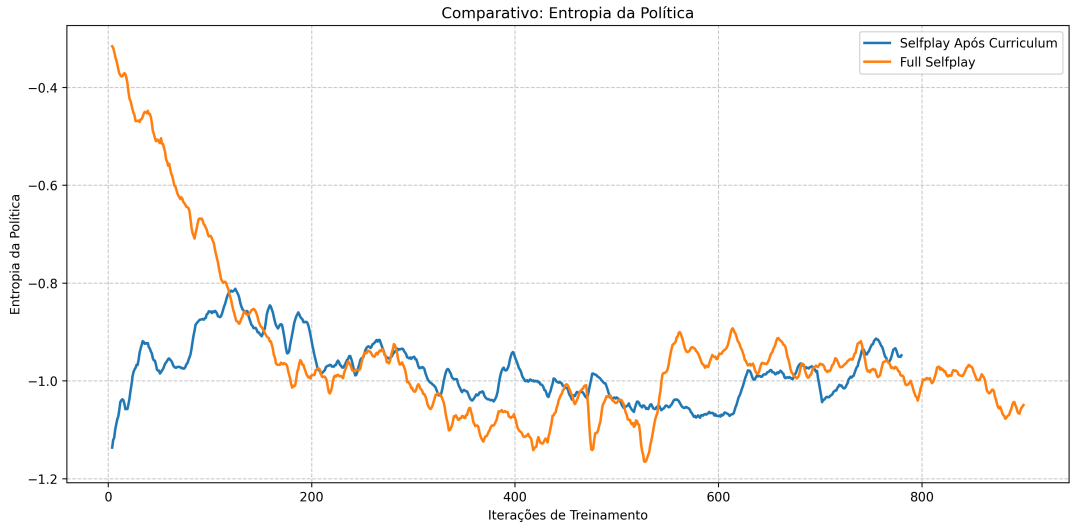

Figura 4.12: *Comparativo da entropia da política: Selfplay após Curriculum e Full Selfplay*

A análise do gráfico revela diferenças significativas nos padrões de exploração-explotação entre as duas abordagens. O *Full Selfplay* (linha laranja) inicia o treinamento

[1] https://drive.google.com/drive/folders/1dT-Bp7ocl20wf2JIIIZPvPQepCzyFvfx?usp=sharing

com valores de entropia mais altos (próximos a -0,3), indicando uma política mais exploratória, o que é esperado para agentes que começam o aprendizado sem conhecimento prévio. Em contraste, o *Selfplay* após *Curriculum* (linha azul) começa com valores de entropia significativamente mais baixos (aproximadamente -1,1), sugerindo uma política já mais determinística.

Esta diferença inicial é particularmente reveladora: agentes treinados com *curriculum learning* iniciam a fase de *selfplay* com políticas mais refinadas e menos aleatórias, evidenciando que o conhecimento adquirido durante os estágios do *curriculum* proporciona maior certeza nas ações a serem tomadas.

Após aproximadamente 150-200 iterações, observa-se uma convergência nas entropias, com ambas as abordagens chegando a valores similares. No entanto, é notável que o *Full Selfplay* apresenta maior volatilidade ao longo de todo o treinamento, com oscilações mais pronunciadas, enquanto o *Selfplay* após *Curriculum* mantém níveis mais estáveis de entropia, sugerindo um processo de aprendizado mais consistente e menos errático.

4.3.2 Perda da Política

A perda da política é uma métrica que reflete a divergência entre a política atual e a política que seria ótima segundo as estimativas atuais da função de valor. Em algoritmos como *PPO*, a minimização desta perda é um dos objetivos principais do processo de otimização. A Figura 4.13 apresenta a evolução desta métrica durante o treinamento.

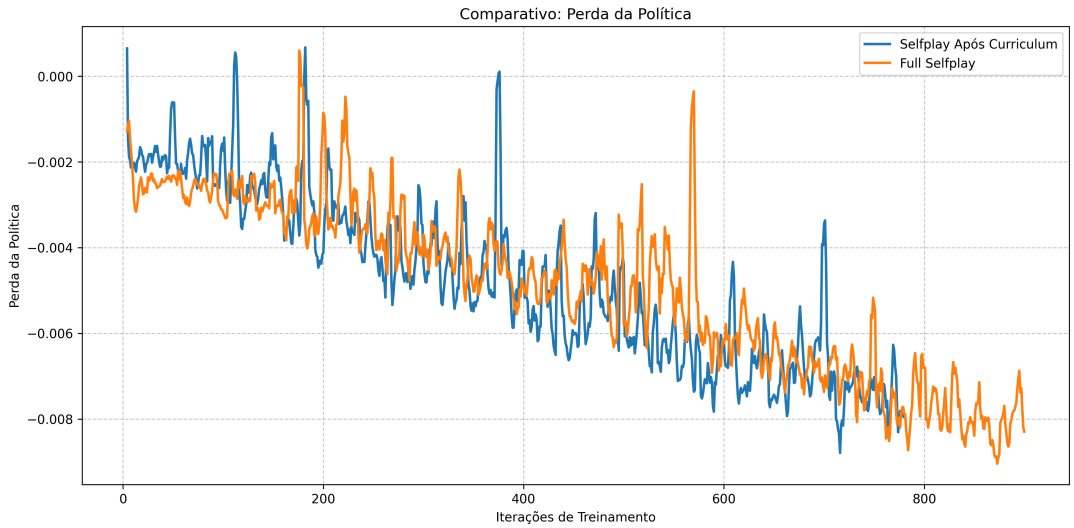

Figura 4.13: *Comparativo da perda da política: Selfplay após Curriculum e Full Selfplay*

O gráfico de perda da política mostra um comportamento interessante: ambas as abordagens iniciam com valores similares e seguem uma tendência geral de redução da

perda ao longo do treinamento, o que indica uma melhoria progressiva nas políticas. No entanto, há diferenças notáveis na trajetória dessa redução.

Observa-se que ambas as abordagens apresentam alta volatilidade, com várias oscilações ao longo do processo. Esta característica é típica de ambientes competitivos como o *self-play*, onde as mudanças na política de um oponente podem temporariamente aumentar a perda até que o agente se adapte.

Um aspecto particularmente relevante é que o *Full Selfplay* continua seu treinamento por mais iterações e alcança valores de perda mais negativos nas fases finais. Isto pode indicar que, sem o benefício do *curriculum* inicial, esta abordagem requer um período mais longo de refinamento para atingir níveis comparáveis de otimização da política.

A análise conjunta com as outras métricas sugere que, embora o *Full Selfplay* eventualmente alcance valores de perda similares ou até melhores, o caminho para chegar a este ponto é mais longo e menos eficiente comparado ao *Selfplay* após *Curriculum*.

4.3.3 Variância Explicada da Função Valor

A variância explicada é uma métrica que avalia a qualidade da função valor aprendida pelo agente, indicando quão bem o modelo consegue prever retornos futuros. Valores próximos a 1 indicam alta precisão nas previsões, enquanto valores mais baixos sugerem maior incerteza. A Figura 4.14 apresenta a comparação desta métrica entre as duas abordagens.

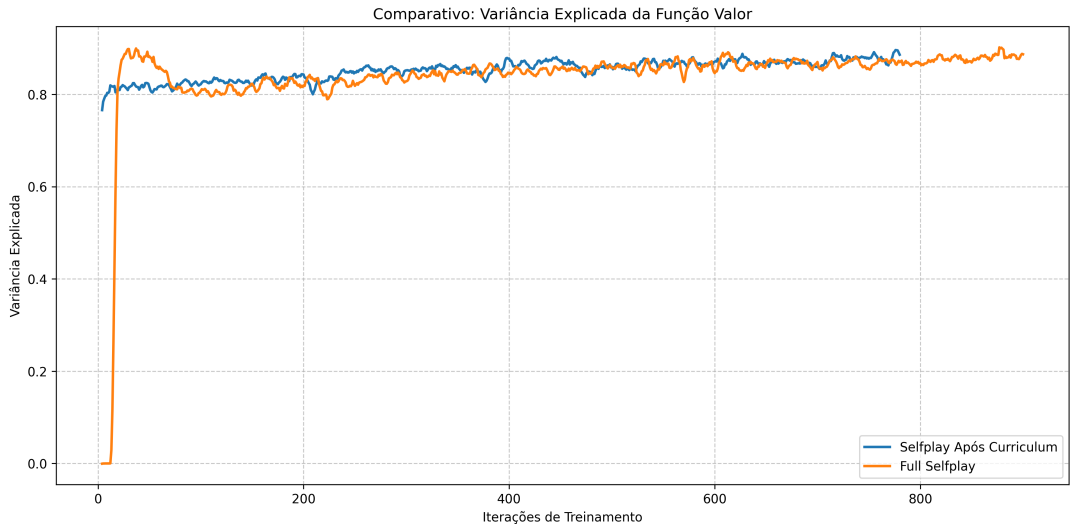

Figura 4.14: *Comparativo da variância explicada da função valor: Selfplay após Curriculum e Full Selfplay*

A análise do gráfico de variância explicada revela padrões distintos no desenvolvimento da função valor. O *Full Selfplay* (linha laranja) apresenta um comportamento

curioso nas primeiras iterações, com um pico inicial seguido por uma queda abrupta. Este padrão pode ser atribuído a uma superestimação inicial da capacidade preditiva, seguida por um ajuste à medida que o agente enfrenta situações mais diversificadas.

Em contraste, o *Selfplay* após *Curriculum* (linha azul) inicia com valores mais estáveis, sem os extremos observados no *Full Selfplay*. Esta estabilidade inicial é mais uma evidência dos benefícios do treinamento curricular prévio, que proporciona ao agente uma base mais sólida para estimar recompensas futuras.

Após aproximadamente 200 iterações, ambas as abordagens convergem para valores similares de variância explicada, em torno de 0,85, indicando que ambos os métodos eventualmente desenvolvem funções valor de qualidade comparável. No entanto, o caminho para atingir esta convergência é notavelmente diferente, com o *Selfplay* após *Curriculum* demonstrando maior consistência ao longo do processo.

Nas fases finais do treinamento, após a convergência, ambas as abordagens mantêm níveis similares e estáveis de variância explicada, sugerindo que, embora o processo de aprendizado seja diferente, o resultado final em termos de capacidade preditiva da função valor é comparável.

4.3.4 Implicações para o Processo de Aprendizagem

A análise integrada das três métricas básicas de aprendizado por reforço revela padrões consistentes que destacam as diferenças fundamentais entre as abordagens *Selfplay* após *Curriculum* e *Full Selfplay*.

Em primeiro lugar, observa-se que o *Selfplay* após *Curriculum* consistentemente demonstra maior estabilidade nas fases iniciais e intermediárias do treinamento. Esta característica é particularmente valiosa em cenários complexos como o futebol de robôs, onde a volatilidade excessiva pode levar a políticas subótimas ou comportamentos indesejados.

Em segundo lugar, a transição mais suave nas métricas de aprendizado do *Selfplay* após *Curriculum* sugere que o conhecimento adquirido durante os estágios do *curriculum* proporciona um ponto de partida mais avançado para o desenvolvimento de políticas competitivas. Esta vantagem inicial se traduz em um processo de aprendizado mais eficiente, requerendo menos iterações para atingir níveis comparáveis de desempenho.

Por fim, embora ambas as abordagens eventualmente convirjam para valores similares nas métricas analisadas, o caminho para esta convergência é significativamente diferente. O *Selfplay* após *Curriculum* oferece um processo de aprendizado mais direto e consistente, enquanto o *Full Selfplay* requer um período mais longo de ajustes e adaptações antes de atingir estabilidade.

Estas observações corroboram a hipótese central deste trabalho: o *curriculum learning* como fase preparatória proporciona um alicerce mais sólido para o desenvolvimento de políticas complexas, resultando em um processo de aprendizado mais eficiente e estável durante o subsequente treinamento competitivo via *self-play*.

4.4 Discussão dos Resultados

A análise dos resultados experimentais permite estabelecer conclusões substanciais sobre a eficácia da abordagem proposta. Esta seção sintetiza as principais descobertas e suas implicações.

4.4.1 Síntese dos Resultados Experimentais

Os experimentos realizados revelam superioridade consistente da abordagem combinada (*Curriculum + Self-play*) em relação às alternativas. Esta superioridade manifesta-se em três dimensões principais:

1. **Desempenho competitivo**: A taxa de vitória de 86% (430 vitórias em 500 partidas) contra o *Full Self-play*, que obteve apenas 1,4% (7 vitórias), representa evidência inequívoca da eficácia da abordagem proposta. A capacidade de marcar gols também se destaca, com média de 2,024 gols por partida (1012 gols em 500 partidas), muito superior aos 0,018 gols por partida do *Full Self-play*.
2. **Eficiência computacional**: A redução de aproximadamente 15% no tempo total de treinamento (7,4 horas versus 8,7 horas) demonstra ganho significativo de eficiência, fator crítico para aplicações práticas.
3. **Estabilidade do aprendizado**: As métricas básicas de aprendizado por reforço (entropia da política, perda da política e variância explicada) evidenciam um processo mais estável e consistente, com menor volatilidade durante as fases críticas do treinamento.

Os dados apresentam ainda um padrão claro de complementaridade entre as abordagens. O *Curriculum Learning* isolado produz agentes com desempenho limitado (apenas 33 vitórias contra 247 do *Full Self-play* em 500 partidas), enquanto o *Self-play* puro desenvolve agentes com capacidades mais equilibradas, porém inferiores à abordagem combinada. A estratégia de *Curriculum + Self-play* potencializa as vantagens de ambas as abordagens, resultando em agentes tecnicamente refinados e taticamente eficazes.

4.4.2 Confirmação da Hipótese

Os resultados obtidos confirmam consistentemente a hipótese central deste trabalho: o *curriculum learning* como fase preparatória para o *self-play* melhora significativamente as políticas aprendidas, resultando em agentes com desempenho superior.

Esta confirmação apoia-se em evidências estatísticas, obtidas no torneio proposto neste trabalho, onde foram realizadas 500 partidas entre os diferentes agentes. O *curriculum learning* proporciona fundamentos técnicos que permitem ao agente aproveitar melhor a fase competitiva do *self-play*, acelerando e otimizando o desenvolvimento de políticas eficazes, como demonstrado pela taxa de vitória de 86% contra o *Full Self-play*.

4.4.3 Limitações do Estudo

Apesar dos resultados promissores, algumas limitações devem ser consideradas:

- **Generalização para ambientes físicos**: Os experimentos foram conduzidos exclusivamente em simulação, existindo o desafio conhecido do *reality gap* na transferência para robôs reais.
- **Sensibilidade paramétrica**: A eficácia do *curriculum learning* depende do design apropriado das tarefas e critérios de promoção, cuja otimização sistemática não foi completamente explorada.
- **Especificidade do domínio**: Embora os princípios sejam potencialmente generalizáveis, os resultados foram validados especificamente no contexto do futebol de robôs.

4.4.4 Implicações para Aprendizado por Reforço

As descobertas deste trabalho têm implicações que transcendem o domínio específico do futebol de robôs:

1. **Valor do aprendizado estruturado**: Em domínios complexos com espaço de ações amplo e *feedback* esparso, o treinamento progressivo demonstra benefícios significativos.
2. **Importância de métricas diversificadas**: A análise exclusiva de métricas convencionais (como recompensa acumulada) pode obscurecer nuances importantes no processo de aprendizado e qualidade das políticas.
3. **Complementaridade de abordagens**: Diferentes técnicas de treinamento podem desenvolver habilidades complementares, cuja combinação resulta em agentes com desempenho superior à soma das partes.

A metodologia desenvolvida neste trabalho oferece um *framework* transferível para o design de trajetórias de aprendizado em ambientes multiagentes complexos, especialmente aqueles que compartilham características como necessidade de coordenação, *feedback* esparso e complexidade estratégica.

CAPÍTULO 5

Conclusão

Este trabalho investigou a integração de *Curriculum Learning* com *Self-play* para aprendizado por reforço no contexto do futebol de robôs da categoria *SSL-EL*. Os experimentos e análises realizados permitem extrair conclusões importantes sobre a eficácia da abordagem proposta e suas implicações para o treinamento de agentes em ambientes complexos e multiagentes.

5.1 Principais Descobertas e Contribuições

A análise integrada dos resultados experimentais permitiu identificar as seguintes descobertas e contribuições principais:

1. **Metodologia estruturada**: Desenvolvimento de um *framework* para integração de *Curriculum Learning* e *Self-play*, com definição clara de estágios progressivos e critérios de transição adaptativos.
2. **Eficácia do curriculum learning**: A abordagem proposta, combinando *curriculum learning* e *self-play*, demonstrou superioridade estatisticamente significativa em termos de desempenho global, evidenciada pela maior taxa de vitória em confrontos diretos (86% de vitórias contra 1,4% do *Full Self-play*) e melhor equilíbrio entre métricas ofensivas e defensivas.
3. **Evidência empírica**: Comprovação quantitativa dos benefícios da abordagem combinada, com ganhos de 86% na taxa de vitória em torneios comparativos (430 vitórias em 500 partidas) e aumento impressionante na média de gols por partida (2,024 vs 0,018 do *Full Self-play*, mais de 100 vezes superior).
4. **Desenvolvimento de habilidades fundamentais**: O *curriculum learning* promoveu o desenvolvimento eficiente de habilidades fundamentais em apenas 42 minutos de treinamento, resultando em melhorias significativas nas métricas de continuidade do jogo e controle técnico.
5. **Eficiência computacional**: Redução de aproximadamente 15% no tempo total de treinamento (7,4 horas versus 8,7 horas), fator crítico para aplicações práticas.

6. **Estabilidade aprimorada**: A abordagem proposta apresentou maior estabilidade durante o processo de aprendizagem, com menor variabilidade nas métricas de desempenho e progressão mais consistente.

Estas descobertas corroboram a premissa central deste trabalho: o aprendizado estruturado e progressivo proporcionado pelo *curriculum learning* oferece vantagens significativas para o treinamento de agentes em ambientes complexos como o futebol de robôs, proporcionando uma base técnica sólida que potencializa os benefícios do *Self-play*. Agentes treinados com esta abordagem desenvolvem inicialmente habilidades defensivas e de controle básico que, quando refinadas através do *Self-play*, evoluem naturalmente para comportamentos táticos sofisticados e eficientes.

5.2 Implicações para Aprendizado por Reforço em Ambientes Multiagentes

Os resultados obtidos têm implicações mais amplas para o campo do aprendizado por reforço em ambientes multiagentes, estendendo-se além do domínio específico do futebol de robôs:

1. **Valor do aprendizado estruturado**: Em domínios complexos com espaço de ações amplo e *feedback* esparso, o treinamento progressivo demonstra benefícios significativos. O *curriculum learning* oferece uma abordagem estruturada para decompor problemas complexos em desafios gerenciáveis, facilitando o desenvolvimento de competências em uma sequência lógica.
2. **Importância de métricas diversificadas**: A análise exclusiva de métricas convencionais (como recompensa acumulada) pode obscurecer nuances importantes no processo de aprendizado e qualidade das políticas aprendidas.
3. **Complementaridade de abordagens**: Diferentes técnicas de treinamento podem desenvolver habilidades complementares, cuja combinação resulta em agentes com desempenho superior à soma das partes.
4. **Desenvolvimento de políticas robustas**: O equilíbrio superior entre características ofensivas e defensivas observado no modelo proposto sugere que o *curriculum learning* pode promover o desenvolvimento de políticas mais robustas e versáteis, capazes de adaptar-se a diferentes contextos e adversários.

A metodologia desenvolvida neste trabalho oferece um *framework* transferível para o design de trajetórias de aprendizado em ambientes multiagentes complexos, especialmente aqueles que compartilham características como necessidade de coordenação, *feedback* esparso e complexidade estratégica.

5.3 Transferibilidade dos Resultados

Uma consideração importante é a transferibilidade dos resultados obtidos para outros domínios e aplicações. Embora os experimentos tenham sido realizados no contexto específico do futebol de robôs, os princípios fundamentais da abordagem proposta podem ser adaptados para diversos cenários multiagentes.

O *framework* de *curriculum learning* desenvolvido neste trabalho oferece uma metodologia generalizável para o design de trajetórias de aprendizado em ambientes complexos. Os critérios de promoção adaptativos e a integração com *self-play* representam contribuições que podem ser aplicadas em domínios que compartilham características como:

- Espaço de ações amplo e contínuo
- Necessidade de coordenação multiagente
- *Feedback* esparso ou atrasado
- Complexidade estratégica e tática
- Oposição adaptativa

Exemplos potenciais de aplicação incluem robótica colaborativa, sistemas de transporte autônomos coordenados, gerenciamento de recursos distribuídos e simulações militares.

A metodologia experimental desenvolvida, incluindo as métricas de avaliação e o sistema de torneios, também oferece um *template* valioso para a avaliação comparativa de diferentes abordagens de treinamento em ambientes complexos.

5.4 Limitações

As principais limitações identificadas neste estudo incluem:

- **Incerteza sobre o *Reality gap***: Todos os experimentos foram realizados em simulação, persistindo o desafio de transferência para robôs físicos.
- **Sensibilidade paramétrica**: A eficácia do *curriculum* depende significativamente do design apropriado das tarefas e critérios de promoção.
- **Especificidade do domínio**: Embora os princípios sejam potencialmente generalizáveis, a validação ocorreu especificamente no contexto do futebol de robôs.

5.5 Trabalhos Futuros

O potencial demonstrado pela abordagem sugere diversas direções promissoras:

- **Transferência para robôs reais**: Investigação de técnicas para mitigar o *reality gap* e aplicar o conhecimento adquirido em simulação em ambientes físicos.
- **Generalização a outros domínios**: Adaptação da metodologia para diferentes contextos multiagente competitivos e cooperativos.
- **Otimização automática de currículos**: Desenvolvimento de algoritmos para geração e adaptação automática de sequências de tarefas, reduzindo a necessidade de design manual.
- **Exploração de arquiteturas híbridas**: Integração do *framework* proposto com técnicas emergentes como aprendizado por demonstração e modelos baseados em memória.

Em síntese, este trabalho demonstra que a abordagem estruturada do aprendizado, combinando *Curriculum Learning* e *Self-play*, representa uma estratégia eficaz para desenvolver agentes com desempenho superior em ambientes complexos multiagente. A metodologia proposta oferece um caminho promissor para superar limitações atuais em RL e avançar o estado da arte em sistemas autônomos inteligentes.

Referências

[Almeida 2013]ALMEIDA, H. O. de. *Agentes Inteligentes e Sistemas Multi-Agentes*. Tese (Doutorado) — Pontifícia Universidade Católica do Rio de Janeiro, 2013. Disponível em: <https://www.maxwell.vrac.puc-rio.br/21194/21194$_3$.PDF>.

[Bai e Jin 2020]BAI, Y.; JIN, C. Provable self-play algorithms for competitive reinforcement learning. In: III, H. D.; SINGH, A. (Ed.). *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020. (Proceedings of Machine Learning Research, v. 119), p. 551–560. Disponível em: <https://proceedings.mlr.press/v119/bai20a.html>.

[Bassani et al. 2020]BASSANI, H. F. et al. Learning to play soccer by reinforcement and applying sim-to-real to compete in the real world. *CoRR*, abs/2003.11102, 2020. Disponível em: <https://arxiv.org/abs/2003.11102>.

[Brandão et al. 2022]BRANDãO, B. et al. Multiagent reinforcement learning for strategic decision making and control in robotic soccer through self-play. *IEEE Access*, v. 10, p. 72628–72642, 2022.

[Chang et al. 2022]CHANG, C. et al. E-mapp: Efficient multi-agent reinforcement learning with parallel program guidance. In: KOYEJO, S. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2022. v. 35, p. 12154–12168. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2022/file/4f2accafe6fa355624f3ee42207cc7b8-Paper-Conference.pdf>.

[6]CHRISTIANOS, F.; SCHäFER, L.; ALBRECHT, S. Shared experience actor-critic for multi-agent reinforcement learning. In: LAROCHELLE, H. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020. v. 33, p. 10707–10717. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2020/file/7967cc8e3ab559e68cc944c44b1cf3e8-Paper.pdf>.

[DiGiovanni e Zell 2021]DIGIOVANNI, A.; ZELL, E. Survey of self-play in reinforcement learning. *ArXiv*, abs/2107.02850, 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:235755071>.

[DLR-RM 2019]DLR-RM. *Stable Baselines3*. 2019. https://github.com/DLR-RM/stable-baselines3. Acessado em: outubro de 2024.

[Face 2024]FACE, H. *(Automatic) Curriculum Learning for RL - Deep RL Course*. 2024. https://huggingface.co/learn/deep-rl-course/unitbonus3/curriculum-learning. Acessado em: março de 2024. Disponível em: <https://huggingface.co/learn/deep-rl-course/unitbonus3/curriculum-learning>.

[Face 2024]FACE, H. *Self-Play: a classic technique to train competitive agents in adversarial games*. 2024. https://huggingface.co/learn/deep-rl-course/unit7/self-play. Accessed in 2024.

[Foglino e Leonetti 2019]FOGLINO, F.; LEONETTI, M. An optimization framework for task sequencing in curriculum learning. *2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, p. 207–214, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:59523682>.

[Gupta et al. 2019]GUPTA, A. et al. *Relay Policy Learning: Solving Long-Horizon Tasks via Imitation and Reinforcement Learning*. 2019. Disponível em: <https://arxiv.org/abs/1910.11956>.

[Haarnoja et al. 2024]HAARNOJA, T. et al. Learning agile soccer skills for a bipedal robot with deep reinforcement learning. *Science Robotics*, v. 9, n. 89, p. eadi8022, 2024. Disponível em: <https://www.science.org/doi/abs/10.1126/scirobotics.adi8022>.

[Huh e Mohapatra 2024]HUH, D.; MOHAPATRA, P. *Multi-agent Reinforcement Learning: A Comprehensive Survey*. 2024. Disponível em: <https://arxiv.org/abs/2312.10256>.

[15]IMPLEMENTAçãO do trabalho de Brandão, Bruno et al. 2024. https://github.com/Pequi-Mecanico-SSL/RL. Disponível em: <https://github.com/Pequi-Mecanico-SSL/RL>.

[ITAndroids Small Size League Team Description Paper for RoboCup 2023 2023] ITANDROIDS Small Size League Team Description Paper for RoboCup 2023. 2023. https://ssl.robocup.org/wp-content/uploads/2023/02/2023_TDP_ITAndroids.pdf. Disponível em: <https://ssl.robocup.org/wp-content/uploads/2023/02/2023$_TDP_ITAndroids.pdf$>.

[Ji et al. 2024]JI, X. et al. *Self-Play with Adversarial Critic: Provable and Scalable Offline Alignment for Language Models*. 2024. Disponível em: <https://arxiv.org/abs/2406.04274>.

[18]KIM, S.; LEE, K.; CHOI, J. Variational curriculum reinforcement learning for unsupervised discovery of skills. In: *Proceedings of the 40th International Conference on Machine Learning*. [S.l.]: JMLR.org, 2023. (ICML'23).

[Klink et al. 2022]KLINK, P. et al. Boosted curriculum reinforcement learning. In: *International Conference on Learning Representations*. [s.n.], 2022. Disponível em: <https://openreview.net/forum?id=anbBFIX1tJ1>.

[Klink et al. 2022]KLINK, P. et al. Curriculum reinforcement learning via constrained optimal transport. In: CHAUDHURI, K. et al. (Ed.). *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022. (Proceedings of Machine Learning Research, v. 162), p. 11341–11358. Disponível em: <https://proceedings.mlr.press/v162/klink22a.html>.

[Martins 2023]MARTINS, F. B. *Exploring multi-agent deep reinforcement learning in IEEE very small size soccer*. Tese (Doutorado) — Universidade Federal de Pernambuco, 2023.

[Martins et al. 2022]MARTINS, F. B. et al. rsoccer: A framework for studying reinforcement learning in small and very small size robot soccer. In: ALAMI, R. et al. (Ed.). *RoboCup 2021: Robot World Cup XXIV*. Cham: Springer International Publishing, 2022. p. 165–176. ISBN 978-3-030-98682-7.

[Narvekar 2017]NARVEKAR, S. Curriculum learning in reinforcement learning. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. [s.n.], 2017. p. 5195–5196. Disponível em: <https://doi.org/10.24963/ijcai.2017/757>.

[Narvekar et al. 2020]NARVEKAR, S. et al. Curriculum learning for reinforcement learning domains: A framework and survey. *J. Mach. Learn. Res.*, JMLR.org, v. 21, n. 1, jan. 2020. ISSN 1532-4435.

[OpenAI 2018]OPENAI. *OpenIA - Proximal Policy Optimization*. 2018. https://spinningup.openai.com/en/latest/algorithms/ppo.html. Acessado em: 15 de outubro de 2024.

[OrcaBOT Team Description Paper 2024 2024]ORCABOT Team Description Paper 2024. 2024. https://ssl.robocup.org/wp-content/uploads/2024/04/2024_TDP_OrcaBOT.pdf. Disponível em: <https://ssl.robocup.org/wp-content/uploads/2024/04/2024$_TDP_OrcaBOT.pdf$>.

[Oroojlooyjadid e Hajinezhad 2019]OROOJLOOYJADID, A.; HAJINEZHAD, D. A review of cooperative multi-agent deep reinforcement learning.

Applied Intelligence, v. 53, p. 13677–13722, 2019. Disponível em: <https://api.semanticscholar.org/CorpusID:199543559>.

[Portelas et al. 2021]PORTELAS, R. et al. Automatic curriculum learning for deep rl: a short survey. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2021. (IJCAI'20). ISBN 9780999241165.

[PyTorch 2024]PYTORCH. *Reinforcement Learning (PPO) with TorchRL Tutorial*. 2024. https://pytorch.org/rl/main/tutorials/coding_ppo.html. Acessado em: outubro de 2024.

[30]REGRAS para competição Robocup Small Size League Entry-Level (SSL-EL). 2024. https://cbr.robocup.org.br/wp-content/uploads/2024/08/sslrules.pdf. Disponível em: <https://cbr.robocup.org.br/wp-content/uploads/2024/08/sslrules.pdf>.

[Repositório oficial da RoboCup SSL Brasil 2024]REPOSITóRIO oficial da RoboCup SSL Brasil. 2024. https://github.com/robocup-ssl-br. Disponível em: <https://github.com/robocup-ssl-br>.

[RobôCIn Small Size League Extended Team Description Paper for RoboCup 2024 2024] ROBôCIN Small Size League Extended Team Description Paper for RoboCup 2024. 2024. https://ssl.robocup.org/wp-content/uploads/2024/04/2024_ETDP_RoboCIn.pdf. Disponível em: <https://ssl.robocup.org/wp-content/uploads/2024/04/2024$_ETDP_RoboCIn.pdf$>.

[Sayar et al. 2024]SAYAR, E. et al. Diffusion-based curriculum reinforcement learning. In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. [s.n.], 2024. Disponível em: <https://openreview.net/forum?id=yRhrVaDOWE>.

[Schulman et al. 2017]SCHULMAN, J. et al. *Proximal Policy Optimization Algorithms*. 2017. Disponível em: <https://arxiv.org/abs/1707.06347>.

[35]SCHWAB, D.; ZHU, Y.; VELOSO, M. M. Learning skills for small size league robocup. In: *Robot Soccer World Cup*. [s.n.], 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:52212489>.

[Shen 2024]SHEN, Y. Proximal policy optimization with entropy regularization. In: *2024 4th International Conference on Computer, Control and Robotics (ICCCR)*. [S.l.: s.n.], 2024. p. 380–383.

[Silver et al. 2017]SILVER, D. et al. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. 2017. Disponível em: <https://arxiv.org/abs/1712.01815>.

[Site oficial da RobôCin 2025]SITE oficial da RobôCin. 2025. https://www.robocin.com.br/. Disponível em: <https://www.robocin.com.br/>.

[Site oficial do Pequi Mecânico 2025]SITE oficial do Pequi Mecânico. 2025. https://www.pequi-mecanico.com.br/inicio. Disponível em: <https://www.pequi-mecanico.com.br/inicio>.

[Soviany et al. 2022]SOVIANY, P. et al. *Curriculum Learning: A Survey*. 2022. Disponível em: <https://arxiv.org/abs/2101.10382>.

[Sutton e Barto 2018]SUTTON, R. S.; BARTO, A. G. *Reinforcement learning: An introduction*. [S.l.]: MIT press, 2018.

[Technical Overview of the Small Size League 2025]TECHNICAL Overview of the Small Size League. 2025. https://ssl.robocup.org/technical-overview-of-the-small-size-league/. Acessado fev 2025. Disponível em: <https://ssl.robocup.org/technical-overview-of-the-small-size-league/>.

[TurtleRabbit 2024 SSL Team Description Paper 2024]TURTLERABBIT 2024 SSL Team Description Paper. 2024. https://ssl.robocup.org/wp-content/uploads/2024/04/2024_TDP_turtlerabbit.pdf. Disponível em: <https://ssl.robocup.org/wp-content/uploads/2024/04/2024$_TDP_turtlerabbit.pdf$>.

[Verleysen 2023]VERLEYSEN, N. *Proximal Policy Optimization (PPO) for OpenAI Gym Environments using PyTorch*. 2023. https://github.com/VerleysenNiels/PPO-pytorch-gym. Acessado em: outubro de 2024.

[Zhang et al. 2024]ZHANG, R. et al. *A Survey on Self-play Methods in Reinforcement Learning*. 2024. Disponível em: <https://arxiv.org/abs/2408.01072>.

[Zhou et al. 2022]ZHOU, Y. et al. Curml: A curriculum machine learning library. In: *Proceedings of the 30th ACM International Conference on Multimedia*. New York, NY, USA: Association for Computing Machinery, 2022. (MM '22), p. 7359–7363. ISBN 9781450392037. Disponível em: <https://doi.org/10.1145/3503161.3548549>.