

FUNDAÇÃO GETULIO VARGAS  
ESCOLA DE MATEMÁTICA APLICADA – FGV/EMAp  
CURSO DE GRADUAÇÃO EM MATEMÁTICA APLICADA

EXPLORANDO O TOTAL POTENCIAL DE UM AGENTE COM  
REINFORCEMENT LEARNING

por  
Lucas de Brito dos Reis

Rio de Janeiro  
(2019)

FUNDAÇÃO GETULIO VARGAS  
ESCOLA DE MATEMÁTICA APLICADA – FGV/EMAp  
CURSO DE GRADUAÇÃO EM MATEMÁTICA APLICADA

EXPLORANDO O TOTAL POTENCIAL DE UM AGENTE COM  
REINFORCEMENT LEARNING

”Declaro ser o único autor do presente projeto de monografia que se refere ao plano de trabalho a ser executado para continuidade da monografia e ressalto que não recorri a qualquer forma de colaboração ou auxílio de terceiros para realizá-lo a não ser nos casos e para os fins autorizados pelo professor orientador”.

-----  
Lucas de Brito dos Reis

Orientador: Renato Rocha Souza

Rio de Janeiro  
(2019)

LUCAS DE BRITO DOS REIS

EXPLORANDO O TOTAL POTENCIAL DE UM AGENTE COM  
REINFORCEMENT LEARNING

“Projeto de Monografia apresentado à Escola de Matemática Aplicada –  
FGV/EMAp como requisito parcial para continuidade ao trabalho de  
monografia.”

Aprovado em \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_.  
Grau atribuído ao Projeto de Monografia: \_\_\_\_\_ .

\_\_\_\_\_  
Professor Orientador: Renato Rocha Souza

Escola de Matemática Aplicada – FGV/EMAp

Fundação Getulio Vargas

## **Resumo**

Aprendizado por Reforço, ou Reinforcement Learning (RL) é um conjunto de algoritmos e técnicas que consistem em um agente aprender a realizar determinada tarefa por meio de recompensas atribuídas à suas ações. Em nosso estudo, determinamos um benchmark onde essas diversas técnicas podem ser comparadas. Então, dentre as etapas que consistem em uma solução de RL, decidimos estudar o impacto do Tratamento das Observações no treinamento desses algoritmos.

### **Abstract**

Reinforcement Learning (RL) is a field of study where an agent learns to accomplish goals by receiving rewards to their actions. In our study, we determine a new benchmark where these many techniques can be compared. Then, among the several steps that consist into a RL solution, we decided to study the impact of Observation Treatment on the training of such algorithms.

## Conteúdo

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
1.1	O novo Benchmark . . . . .	8
1.2	Proximal Policy Optimization . . . . .	8
<b>2</b>	<b>PESQUISA EXPLORATÓRIA DA LITERATURA</b>	<b>9</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>10</b>
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>11</b>
4.1	Sistema Operacional . . . . .	11
4.2	Gym Retro . . . . .	12
4.3	Agente em ação . . . . .	13
4.4	Tratamento das Observações . . . . .	13
4.5	Imagem Reduzida . . . . .	14
4.6	Tons de Cinza . . . . .	15
<b>5</b>	<b>RESULTADOS</b>	<b>17</b>
5.1	Random Agent . . . . .	17
5.2	Padrão . . . . .	17
5.3	Imagem Reduzida . . . . .	17
5.4	Tons de Cinza . . . . .	18
5.5	Tensorboard . . . . .	19
<b>6</b>	<b>CONCLUSÃO</b>	<b>20</b>
<b>7</b>	<b>BIBLIOGRAFIA CONSULTADA</b>	<b>21</b>

# 1 INTRODUÇÃO

Quando falamos em aprendizado por reforço, nos referimos a algoritmos que, por meio da interação constante com determinado ambiente, tomando decisões (ou ações) a cada passo, recebem um feedback positivo ou negativo para essas ações. Por meio desse feedback, esses algoritmos ajustam seus parâmetros para se aproximar dos comportamentos que geram maior recompensa positiva e se afastar daqueles que geram recompensas negativas. Essas recompensas devem estar em linha com a tarefa que o agente busca aprender, de forma que maximizar as recompensas implique em melhores resultados nessa tarefa. Após numerosas interações com dado ambiente, se espera que o agente tenha otimizado sua tomada de decisão para gerar maiores recompensas, consequentemente obtendo sucesso na tarefa em questão.

O estudo das técnicas de reinforcement learning é relativamente novo na ciência e se encontra muito em alta nos dias atuais. Por conta disso, novos artigos vêm sendo publicados e novas técnicas exploradas em ritmo acelerado. Essa constante expansão dos estudos vem com uma necessidade de podermos analisar e comparar os resultados em um mesmo ambiente de referência. Chamamos esses ambientes de benchmarks.

Atualmente, ainda existem poucos benchmarks que cumprem tal papel. Desse, a grande maioria se concentra em inserir um agente simples, com espaço de ações de dimensão pequena, atuando em um ambiente complexo, frequentemente com a presença de elementos desconhecidos. Assim, o foco em um benchmark como esse é a capacidade do agente de explorar o local onde está inserido e se adaptar a elementos desconhecidos. Apesar do valor de um algoritmo performar bem em um ambiente como esse, para muitas tarefas complexas no mundo real, o principal fator para o sucesso reside na capacidade do agente utilizar todas as ferramentas ao seu dispor para alcançar seu objetivo. [8][3].

Dito isso, decidimos propor um novo benchmark para a análise das técnicas de reinforcement learning. Optamos por inserir um agente complexo (elevada dimensão do espaço de ações) em um ambiente simples (sem elementos desconhecidos ou introduzidos de modo inesperado). O objetivo com esse benchmark é analisar a capacidade de o agente explorar e aprender a utilizar as diversas ferramentas a sua disposição para atingir seu objetivo, tirando o foco da exploração do local onde este está inserido. Acreditamos que esse benchmark pode fornecer uma melhor referência de performance em relação ao potencial do algoritmo de dominar essas atividades mais complexas.

Como algoritmos de Reinforcement Learning aprendem com a interação com o ambiente e precisamos frequentemente de centenas de milhares ou mesmo milhões de interações com o mesmo para estabilizar a performance, ambien-

tes virtuais se mostram ideias para realizar o treinamento e testar a eficácia dessas técnicas. Particularmente, jogos virtuais promovem desafios ideais para fazermos essas análises e é nesse cenário que realizaremos nosso estudo.

No mais, apesar do constante surgimento de novas técnicas e melhorias nos principais algoritmos, existem duas áreas que vem sendo pouco exploradas pela literatura recente. Essas tem uma particularidade interessante, são comuns a todas as técnicas de Reinforcement Learning. Por isso, seu estudo se faz essencial para melhoria da eficiência e performance dessas novas técnicas de maneira global. Essas áreas são o Tratamento da Observação (podendo reduzir o tempo de treinamento necessário para atingir a máxima performance) e a Função de Recompensa (podendo aumentar significativamente os resultados obtidos pelo algoritmo).

Em nosso estudo, portanto, vamos buscar demonstrar o impacto que uma melhoria nessas áreas pode ter no desempenho dos algoritmos de RL. Em vias práticas, enquanto a Função de Recompensa se baseia em variáveis que estão presentes no código do jogo, inacessíveis sem o código fonte do mesmo, o Tratamento das Observações depende apenas das imagens que alimentam nosso algoritmo. Assim, optamos por focar nosso estudo no impacto do Tratamento das Observações.

## 1.1 O novo Benchmark

Para isso, vamos utilizar o jogo 'Mortal Kombat 3 - Genesis'. Nesse jogo, nosso agente deve enfrentar um oponente que possui espaço de ações similar ao seu, buscando reduzir os pontos de vida desse a zero. O confronto se passa em um cenário de duas dimensões (sem rotação de câmera), permitindo o foco apenas no combate. O agente possui diversas opções de movimentos, ataques e golpes especiais para reduzir os pontos de vida do adversário, classificando-se assim como um agente complexo.

## 1.2 Proximal Policy Optimization

Uma das técnicas que vem mostrando os melhores resultados nas pesquisas recentes e é considerada o estado da arte, é a Proximal Policy Optimization (PPO) [10]. Essa técnica possui diversas vantagens que direcionaram nossa escolha. Primeiramente, por ser uma técnica com estrutura mais simples que outras das técnicas mais avançadas, implementações da mesma possuem menos tendências a erros. Além disso, obtém resultados equiparáveis ou superiores a qualquer outro algoritmo de RL, com a vantagem de possuir menor dependência em hiperparâmetros, que são difíceis de se ajustar.



## 2 PESQUISA EXPLORATÓRIA DA LITERATURA

A plataforma Gym, criada pela Open AI, pode ser considerada uma das primeiras tentativas de criar um padrão para que diferentes técnicas de RL fossem comparadas em um mesmo ambiente. Inicialmente com jogos de atari [1] e em seguida com o famoso jogo 'Sonic: The Hedgehog' [8], alguns benchmarks foram estabelecidos. A literatura, entretanto, ainda é recente e deixa espaço para que outros benchmarks sejam apresentados e possibilitem a análise de outros fatores no aprendizado.

Após estudarmos diversas das técnicas mais bem sucedidas de RL, como Rainbow[4], Actor Critic Methods (AC2, AC3)[5] e Deep Reinforcement Learning[7], optamos por utilizar o Proximal Policy Optimization [10] para nossa análise, em razão das vantagens já citadas anteriormente.

Existem estudos que analisam melhorias gerais para algoritmos de machine learning (melhorias que independem da técnica utilizada). Enquanto conseguimos encontrar algumas propostas promissoras para evoluir a aprendizagem utilizando as Funções de Recompensa[2] [9] [6], a literatura se mostra escassa no Tratamento das observações, apesar de existirem também estudos nessa direção [11].

### 3 METODOLOGIA

O algoritmo que serve como base para nosso artigo é o Proximal Policy Optimization (PPO). Utilizamos o algoritmo PPO2 como base para todas as tentativas de melhoria. A implementação escolhida está presente na biblioteca Stable-Baselines.

Contamos com a plataforma Gym Retro para a emulação e integração do jogo e o Tensorboard para acompanhar a evolução do agente ao longo do treino. Então iremos estudar o impacto dos diferentes tratamentos das observações, sendo esses reduzir o tamanho da imagem (Vizinho Mais Próximo) e transformar a imagem em tons de cinza.

Cada agente foi treinado utilizando 8 cpus simultaneamente, por 2500000 timesteps, equivalendo a aproximadamente 18 horas de treino sem o tratamento das observações. Utilizando uma ferramenta chamada Tensorboard, temos acesso à gráficos que mostram as recompensas e diversas variáveis ao longo de cada timestep, assim permitindo a comparação precisa do resultado dos diferentes tratamentos das observações.

## 4 DESENVOLVIMENTO

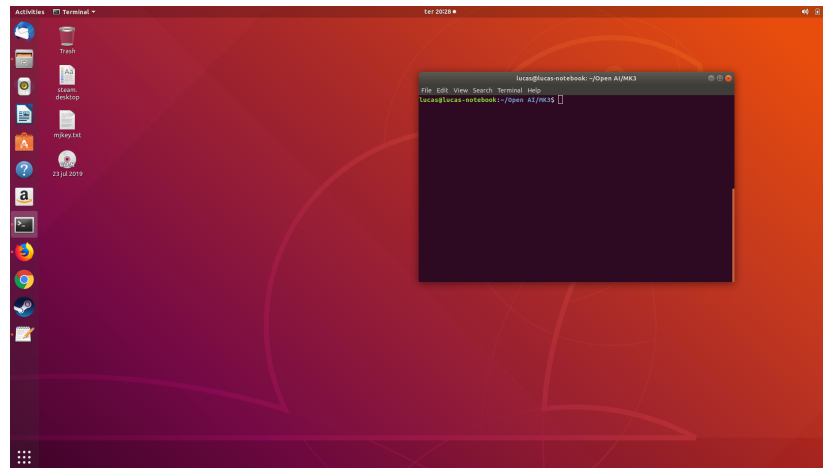
Nosso objetivo é analisar, dentro do novo benchmark, melhorias que possam ser utilizadas em algoritmos de reinforcement learning. Apesar de tomarmos o PPO como base para analisar essas mudanças, as mesmas podem ser utilizadas em conjunto com diversas técnicas atuais de RL. Olhamos em nossa pesquisa para dois pontos que são comuns aos algoritmos de RL e que têm sido pouco explorados na literatura.

O primeiro desses é o Tratamento das Observações, que consiste em simplificar as imagens alimentadas ao modelo no treinamento sem perda significativa de informação. Uma simplificação eficaz dos inputs pode reduzir consideravelmente o tempo de cada passo, resultando em um tempo menor de treinamento e possibilitando experimentações mais rápidas. Existem diversas maneiras que esse tratamento pode ser feito e focamos em analisar o impacto dessas técnicas no treinamento do algoritmo.

Além disso, estudamos as possibilidades de mudanças na Função de Recompensa (RF). Um dos objetivos comuns de pesquisa em Reinforcement Learning, é nos aproximarmos do desenvolvimento de Inteligências Artificiais Gerais (AGI). Por isso, é muito comum a utilização de funções de recompensa muito simples, que requerem o mínimo possível de especialização humana em sua confecção, ao invés de uma RF mais específica. Entretanto, uma RF adequada para o problema é crucial para o aprendizado de tarefas complexas, diminuindo significativamente o tempo de treinamento e especialmente aumentando a performance. Entretanto, para podermos utilizar essas informações, é preciso que tenhamos acesso a elas. Por não termos acesso ao código fonte do jogo utilizado, a obtenção dessas informações se mostrou uma barreira.

### 4.1 Sistema Operacional

Primeiramente, foi necessária a utilização do sistema operacional adequado para realizar o estudo. Diversas bibliotecas e funcionalidades não estavam disponíveis para o sistema operacional Windows, então nosso primeiro passo foi configurar nossa máquina com o SO Ubuntu.



## 4.2 Gym Retro

Essa plataforma emula e permite a integração de diversos jogos eletrônicos para a utilização de técnicas de RL. Por meio dessa, foi possível não só treinar nossos modelos, como ver o agente treinado em ação.



### 4.3 Agente em ação

Após completarmos o treinamento do agente, a plataforma nos permitia visualizar a interação do agente com o ambiente, o que nos permitia entender melhor o comportamento do agente e detectar pontos que poderiam ainda assim ser melhorados.



### 4.4 Tratamento das Observações

Toda imagem é composta por pixels. Usualmente, esses pixels seguem o formato RGB, contendo 3 valores de 0 à 255, sendo esses relativos as cores vermelha, verde e azul, respectivamente. A cada passo, nosso modelo era alimentado com imagens de 224x320 (71680) pixels.



Essa larga quantidade de dados faz com que o modelo precise de demasiado tempo de treinamento para atingir a performance máxima. Assim, se faz oportuna a redução do volume de dados fornecido ao modelo, sem a perda das informações cruciais para tomada de decisão. Para o tratamento dessas observações, foi necessário adaptar o código e as redes neurais para receberem imagens com dimensão diferente da original.

#### 4.5 Imagem Reduzida

A primeira abordagem é a redução da imagem original. Existem diversas maneiras de reduzir essa imagem e optamos pelo método intitulado Vizinhos Mais Próximos. Esse método tem a vantagem de não realizar a interpolação de cores, assim não inserindo novas cores que não estavam presentes na imagem original.



Esse método nos permitiu reduzir o tamanho da imagem em 16 vezes, o que impactou significativamente o tempo de treinamento.

#### 4.6 Tons de Cinza

Outra técnica para simplificar a imagem original, é a conversão para tons de cinza. Removendo as cores da figura, temos apenas um terço dos dados originais (cada pixel passa ter somente o valor de saturação, de 0 à 255).



Apesar da redução no volume de dados ser relevante, podemos notar que a redução na quantidade de combinações possíveis para um pixel é expressivamente maior, já que no lugar de  $256^3$  (16777216) possibilidades, temos agora apenas 256. Além disso, essa técnica tem maior adaptabilidade a novos cenários, já que esses muitas vezes acompanham uma total mudança das cores de fundo presentes na imagem.



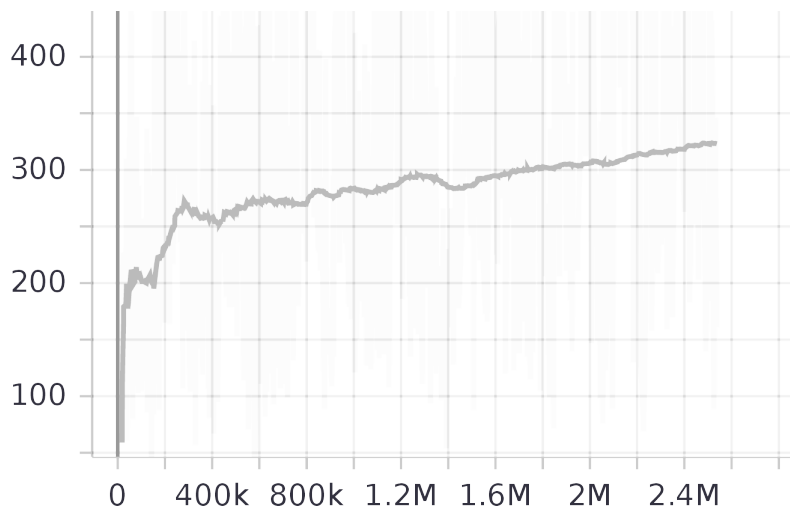
## 5 RESULTADOS

### 5.1 Random Agent

Tomando o agente aleatório como referência, verificamos que sua taxa de vitória contra a inteligência artificial presente no jogo foi inferior a 10%. Esse agente tomava ações completamente aleatórias a cada passo, tendo sua utilidade somente para comparação com os agentes treinados com PPO.

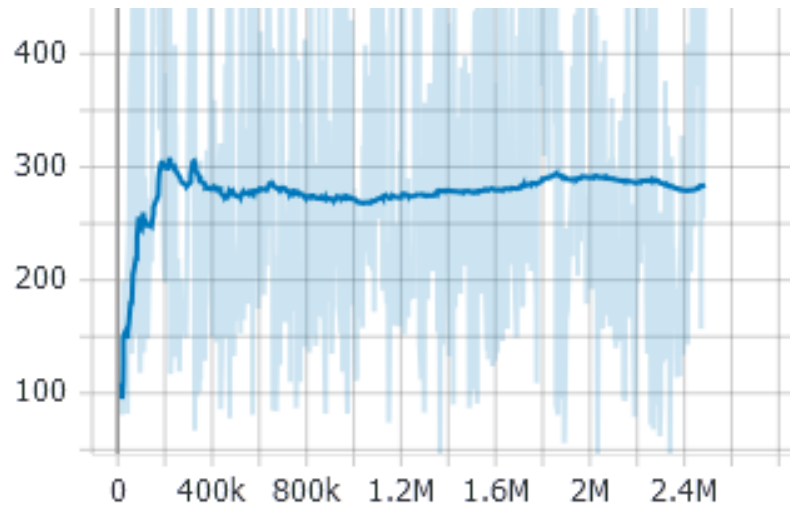
### 5.2 Padrão

Utilizando o algoritmo para treinar o agente no nosso benchmark, conseguimos uma taxa de vitórias de aproximadamente 40%, muito superior à obtida pelo agente aleatório. Após aproximadamente 1.000.000 de timesteps, o agente não demonstrou melhorias significativas. Foi possível notar que, ao vencer o primeiro confronto, avançando assim de fase, o agente demonstrou mais dificuldade de enfrentar o próximo oponente. Isso pode se dar ao menor tempo de treinamento nas fases subsequentes, que possuem um inimigo e cenários diferentes, quanto a maior dificuldade da IA inimiga conforme o progresso das fases.



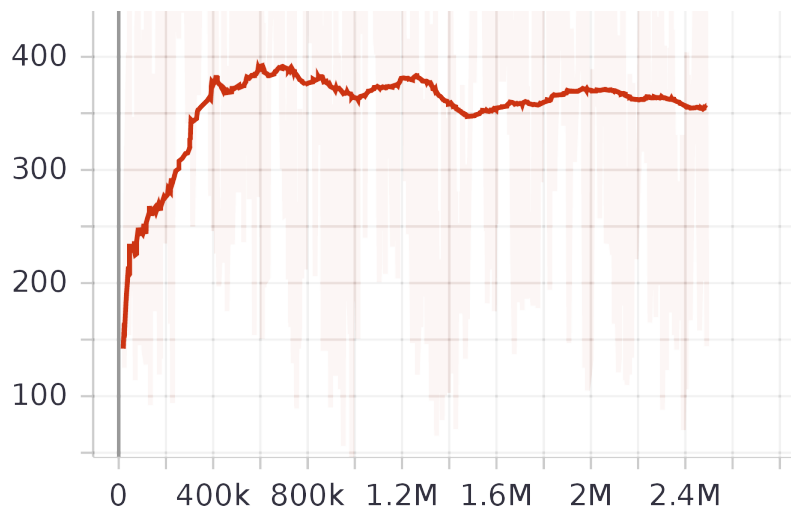
### 5.3 Imagem Reduzida

A imagem reduzida teve o maior impacto no tempo de treinamento, sendo esse reduzido em 16 vezes. Os resultados finais atingidos não mostraram diferenças notáveis em relação ao agente original.



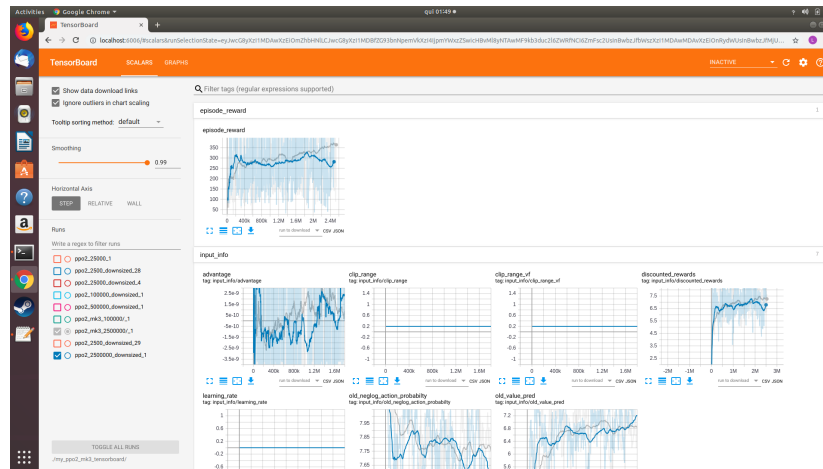
#### 5.4 Tons de Cinza

O algoritmo levou aproximadamente 65% do tempo de treinamento original. Apesar da melhoria pouco significativa no tempo de treinamento, seus resultados foram superior até mesmo ao agente treinado com as imagens no formato original. Isso mostra que um bom tratamento dos dados pode inclusive diminuir os ruídos presentes nas observações, sendo um passo crucial para a performance ótima.



## 5.5 Tensorboard

Com o tensorboard, que nos fornece visualizações gráficas das variáveis relativas ao treinamento, foi possível analisarmos a evolução do nosso agente ao longo dos timesteps. Suavizando as curvas, pudemos verificar e comparar a performance de cada um dos nossos agentes.



## 6 CONCLUSÃO

Foi possível notar que os resultados finais não foram prejudicados pelo Tratamento das Observações. Percebemos inclusive uma melhora no resultado relativo ao tratamento da imagem utilizando Tons de Cinza. Podemos intuir que esse resultado se deve a melhor adaptabilidade à diferentes cenários e ruídos, que facilitou o avanço do agente nas fases seguintes, onde teve menor tempo de treinamento. Também pudemos notar vendo os agentes em ação e comparando seu comportamento com o de experts humanos, que a demasiada simplicidade da Função de Recompensa prejudicou severamente o resultado final, nos indicando um próximo passo para estudos futuros.

## 7 BIBLIOGRAFIA CONSULTADA

- [1] Greg Brockman et al. “Openai gym”. Em: *arXiv preprint arXiv:1606.01540* (2016).
- [2] Aleksandra Faust, Anthony Francis e Dar Mehta. “Evolving Rewards to Automate Reinforcement Learning”. Em: *arXiv preprint arXiv:1905.07628* (2019).
- [3] John Foley et al. “ToyBox: Better Atari Environments for Testing Reinforcement Learning Agents”. Em: *arXiv preprint arXiv:1812.02850* (2018).
- [4] Matteo Hessel et al. “Rainbow: Combining improvements in deep reinforcement learning”. Em: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.
- [5] Vijay R Konda e John N Tsitsiklis. “Actor-critic algorithms”. Em: *Advances in neural information processing systems*. 2000, pp. 1008–1014.
- [6] Matheus Ribeiro Furtado de Mendonça et al. “Evolution of reward functions for reinforcement learning applied to stealth games”. Em: (2016).
- [7] Volodymyr Mnih et al. “Playing atari with deep reinforcement learning”. Em: *arXiv preprint arXiv:1312.5602* (2013).
- [8] Alex Nichol et al. “Gotta learn fast: A new benchmark for generalization in rl”. Em: *arXiv preprint arXiv:1804.03720* (2018).
- [9] Deepak Pathak et al. “Curiosity-driven exploration by self-supervised prediction”. Em: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 16–17.
- [10] John Schulman et al. “Proximal policy optimization algorithms”. Em: *arXiv preprint arXiv:1707.06347* (2017).
- [11] Niklas Wahlström, Thomas B Schön e Marc Peter Deisenroth. “From pixels to torques: Policy learning with deep dynamical models”. Em: *arXiv preprint arXiv:1502.02251* (2015).