Motivação intrínseca para aprendizado de manipulação robótica com recompensas esparsas

Bryan L. M. de Oliveira

Orientadora:
Profa. Dra. Telma W. de L. Soares

INSTITUTO DE INFORMÁTICA
UFG

UFG
UNIVERSIDADE FEDERAL DE GOIÁS

Agenda

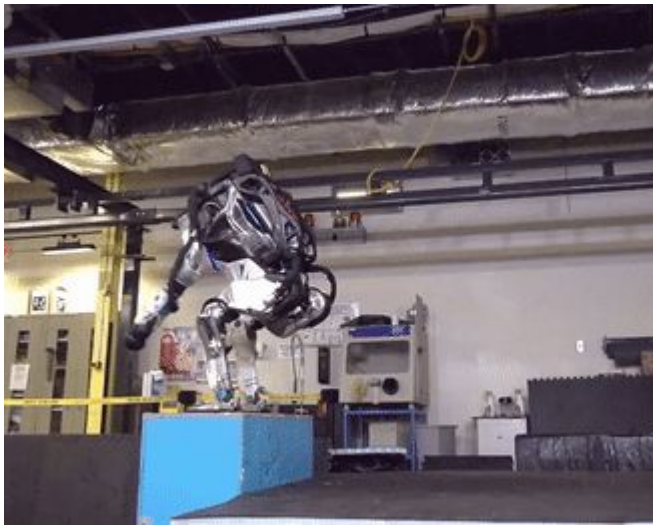
1. Introdução
2. Fundamentos
3. Metodologia
4. Resultados
5. Conclusões

1 Introdução

Introdução

- Comportamentos complexos são difíceis de programar

Figura 1.1: Robô Atlas executando um *backflip*.

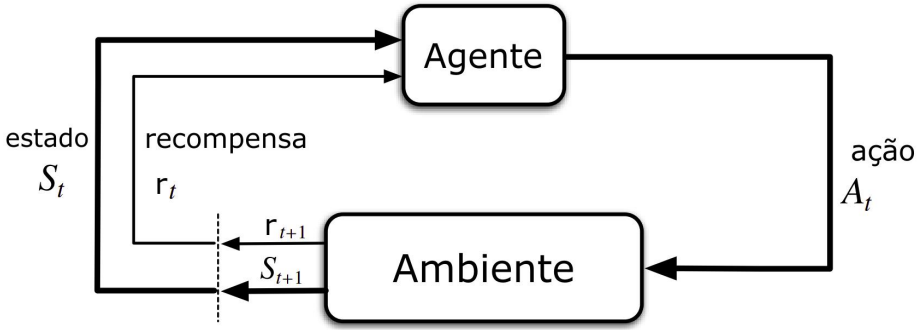


Fonte: Boston Dynamics (2017).

Introdução

- Comportamentos complexos são difíceis de programar
- Aprendizado por reforço

Figura 1.2: Esquema de um sistema de aprendizado por reforço.



Fonte: Sutton & Barto (1998).

Introdução

- Comportamentos complexos são difíceis de programar
- Aprendizado por reforço
- Problema: funções de recompensa

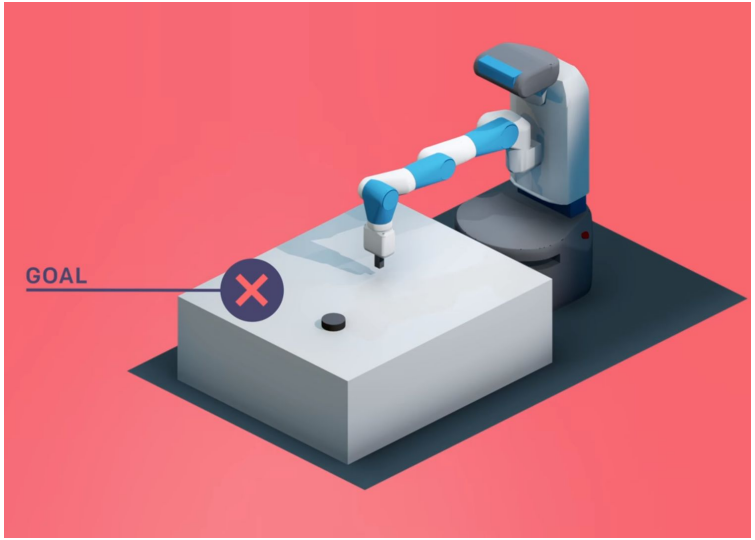
Figura 1.3: Função de recompensa sendo explorada no jogo CoastRunners.

Fonte: OpenAI (2016).

Introdução

- Comportamentos complexos são difíceis de programar
- Aprendizado por reforço
- Problema: funções de recompensa
- Solução: recompensas esparsas

Figura 1.4: Recompensa esparsa em ambientes de manipulação robótica.



Fonte: OpenAI (2018).

Introdução

- Comportamentos complexos são difíceis de programar
- Aprendizado por reforço
- Problema: funções de recompensa
- Solução: recompensas esparsas
- Problema: exploração

Figura 1.5: Sequência de ações muito específica para ser executada aleatoriamente.

Fonte: Montezuma's Revenge (1983).

Introdução

- Comportamentos complexos são difíceis de programar
- Aprendizado por reforço
- Problema: funções de recompensa
- Solução: recompensas esparsas
- Problema: exploração
- Solução: motivação intrínseca

Figura 1.6: Agente com motivação intrínseca resolvendo o jogo Montezuma's Revenge.

Fonte: Montezuma's Revenge (1983).

Proposta

- Aplicar motivação intrínseca em ambientes de manipulação robótica com recompensa esparsa
    - Analisar impactos em relação ao mesmo algoritmo de treinamento sem motivação intrínseca

2 | Fundamentos

Redes Neurais

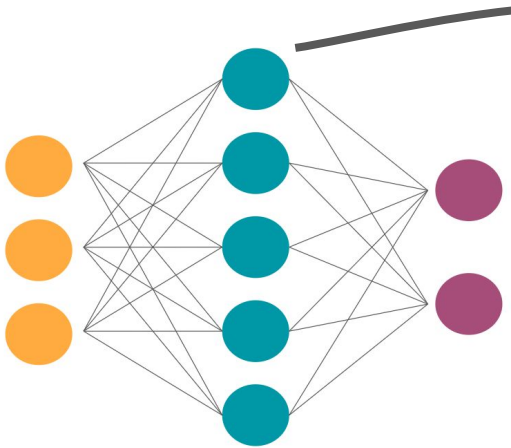


Figura 2.1: Esquema de uma rede neural com uma camada escondida.

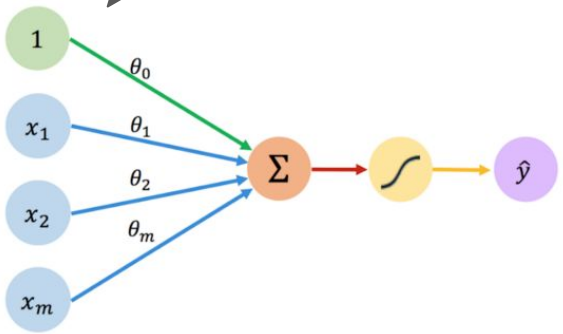


Figura 2.2: Esquema de um neurônio no contexto de redes neurais.

$$\hat{y} = \sigma(w^T x + b)$$

Onde x é o vetor com valores da camada de **entrada** ou saída da **camada anterior**.

Aprendizado por Reforço (AR)

- Sub-área do Aprendizado de Máquina
- Pode ser modelado como um Processo de Decisão de Markov (MDP)

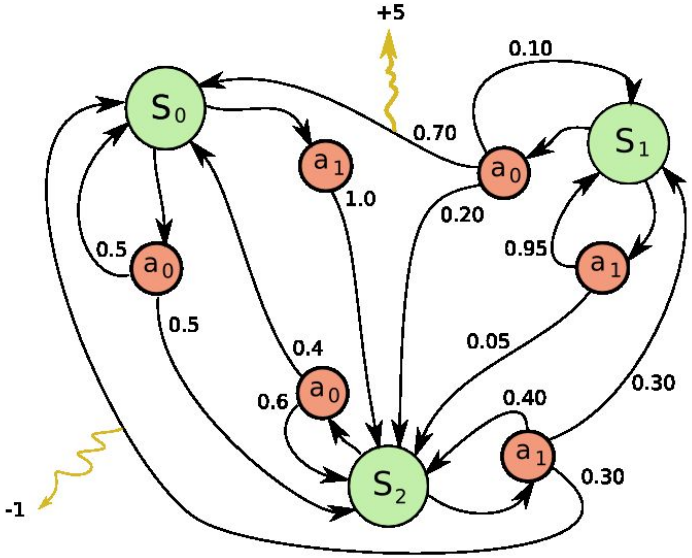


Figura 2.3: Esquema de um Processo de Decisão de Markov.

Aprendizado por Reforço (AR)

- Seleção de ações através da **política**:

$$\pi(a|s) = P(A_t = a|S_t = s)$$

- **Valor** de um estado, onde γ é um fator de desconto:

$$V(s) = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} r_i\right] \forall s \in S$$

- Exploração:
  - ε-greedy
  - Ruído Gaussiano
  - Entropia: $H(\pi(s_t;\theta))$

Estimativa de Vantagem Generalizada (GAE)

- Função de **vantagem** com visão de *n* passos:

$$A_t^{(n)}(s,a) = r_t + \gamma V(s_{t+1}) + \ldots + \gamma^n V(s_{t+n}) - V(s_t)$$

- Estimativa de Vantagem Generalizada, onde λ balanceia o viés:

$$A_t^{GAE(\gamma,\lambda)} = (1-\lambda)(A_t^{(1)} + \lambda A_t^{(2)} + \lambda^2 A_t^{(3)} + \ldots)$$

Proximal Policy Optimization (PPO)

- Proporção entre a nova política e a política antiga:

$$d_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$$

- Função de atualização limitada do PPO:

$$L^{CLIP}(\theta) = \mathbb{E}_t[min(d_t(\theta)A_{\pi_\theta}, clip(d_t(\theta), 1-\varepsilon, 1+\varepsilon)A_{\pi_\theta})]$$

Onde $A_{\pi\theta}$ é uma função de vantagem: $A_{\pi\theta} = r + \gamma V(s_{t+1}) - V(s_t)$

Proximal Policy Optimization

Algoritmo 2.1: PPO com função de atualização limitada

Entrada: parâmetros da política inicial θ_0, limiar de corte ε

para $k = 0, 1, 2, \ldots$ **faça**

Colete um conjunto de trajetórias D_k com política $\pi_k = \pi(\theta_k)$

Estime a função de vantagem $A_t^{GAE(\gamma,\lambda)}$

Calcule a atualização da política

$$\theta_{k+1} = arg\max_{\theta} L^{CLIP+VF+H}(\theta_k)$$

executando N passos do gradiente ascendente, onde

$$L^{CLIP+VF+H}(\theta_k) = \mathbb{E}_t[L^{CLIP} - L^{VF} + cH(\pi(s_t;\theta))]$$

Motivação Intrínseca

- Modelo de Futuro ***f*** dá uma aproximação $\hat{s}_{t+1}$ do próximo estado s_{t+1}, dados o estado atual s_t e a ação a_t:

$$\hat{s}_{t+1} = f(s_t, a_t; \theta_F)$$

- A função de custo do modelo é definida:

$$L_F(s_{t+1}, \hat{s}_{t+1}) = \frac{1}{2}\|\hat{s}_{t+1} - s_{t+1}\|_2^2$$

- Recompensa total para o estado atual, onde **η** é um fator de escala:

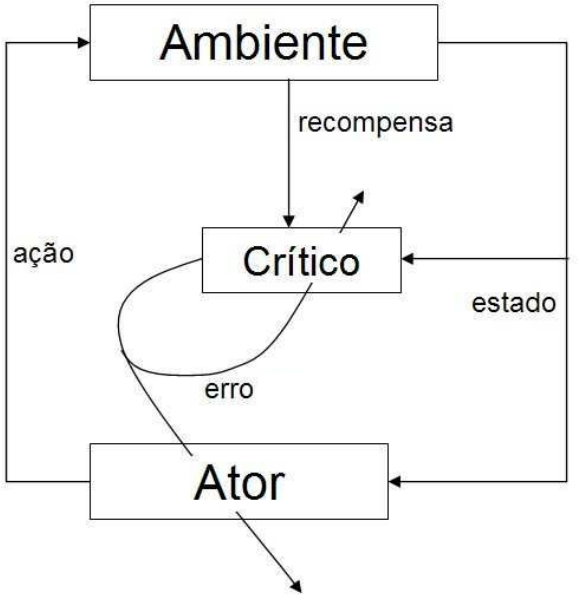
$$r_t = r_t^e(s_t, a, s_{t+1}) + \eta L_F$$

3 Metodologia

PPO com motivação intrínseca

- Modelo Ator-Crítico
  - Ator: política
  - Crítico: função de valor *V(s)*
  - Entrada: 13~28 dimensões
  - 2 camadas escondidas:
    - 128 neurônios
    - Ativação ReLU
  - Saída do ator: 4 dimensões

Figura 3.1: Sistema com modelo ator-crítico.



Fonte: Renan U. B. Ferreira (2006).

PPO com motivação intrínseca

Algoritmo 3.1: PPO com Motivação Intrínseca

Entrada: parâmetros da política inicial θ_0, limiar de corte ε

para $k = 0, 1, 2, \ldots$ **faça**

Colete um conjunto de trajetórias D_k com política $\pi_k = \pi(\theta_k)$

para *cada tupla* (s_t, a_t, r_t, s_{t+1}) *em* D_k **faça**

Calcule o erro do modelo de futuro como mostra a Equação 2-12

Calcule e atualize a recompensa total r_t como mostra a Equação 2-13

Estime a função de vantagem $A_t^{GAE(\gamma,\lambda)}$

Calcule a atualização da política

$$\theta_{k+1} = arg\max_{\theta} L^{CLIP+VF+H}(\theta_k)$$

executando N passos do gradiente ascendente, onde

$$L^{CLIP+VF+H}(\theta_k) = \mathbb{E}_t[L^{CLIP} - L^{VF} + cH(\pi(s_t;\theta))]$$

PPO com motivação intrínseca

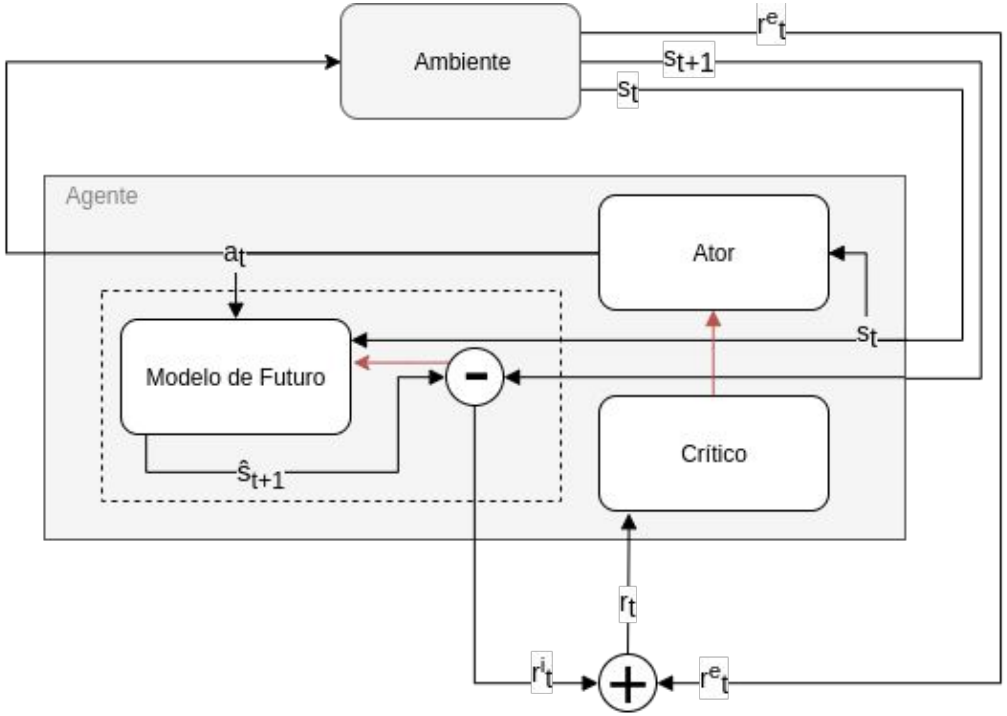


Figura 3.2: Agente com motivação intrínseca

PPO sem motivação intrínseca (baseline)

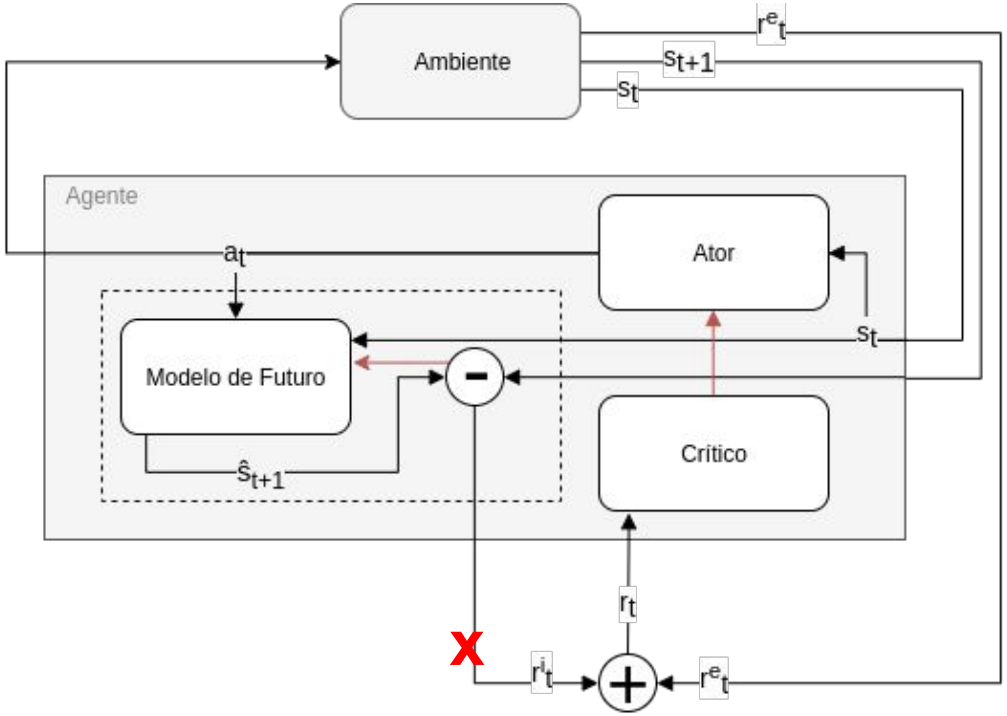
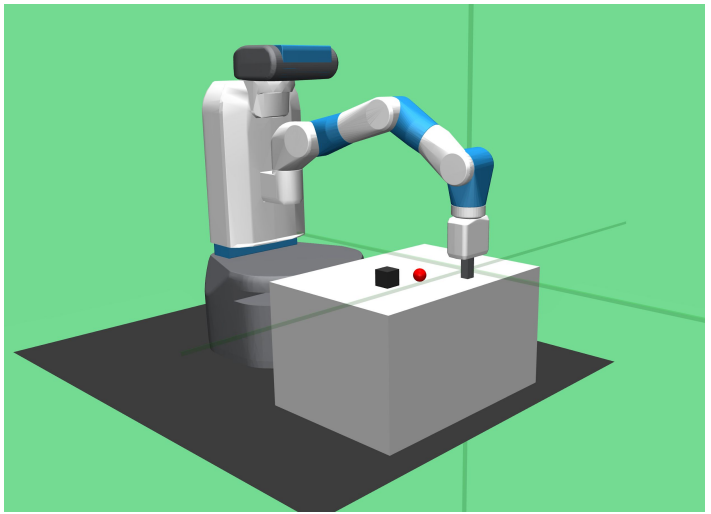


Figura 3.3: Agente sem motivação intrínseca

Gym

- Framework de benchmark para algoritmos de AR
- Ambientes escolhidos:
  - Fetch Reach: **alcançar** um alvo
  - Fetch Push: **empurrar** um bloco a um alvo
  - Fetch Pick and Place: **levar** um bloco a um alvo
- Observação serializada

Figura 3.4: Ambiente Fetch Push, do Gym



Fonte: OpenAI (2019).

Testes

- Hipótese: motivação intrínseca leva a exploração de comportamentos complexos em ambientes de manipulação robótica.
- PPO *baseline* vs. PPO + Motivação Intrínseca
- Métricas
  - Razão de sucessos
  - Recompensa intrínseca média
  - Entropia da política
  - Divergência KL entre as políticas a cada atualização

4 Resultados

Fetch Reach

10M iterações

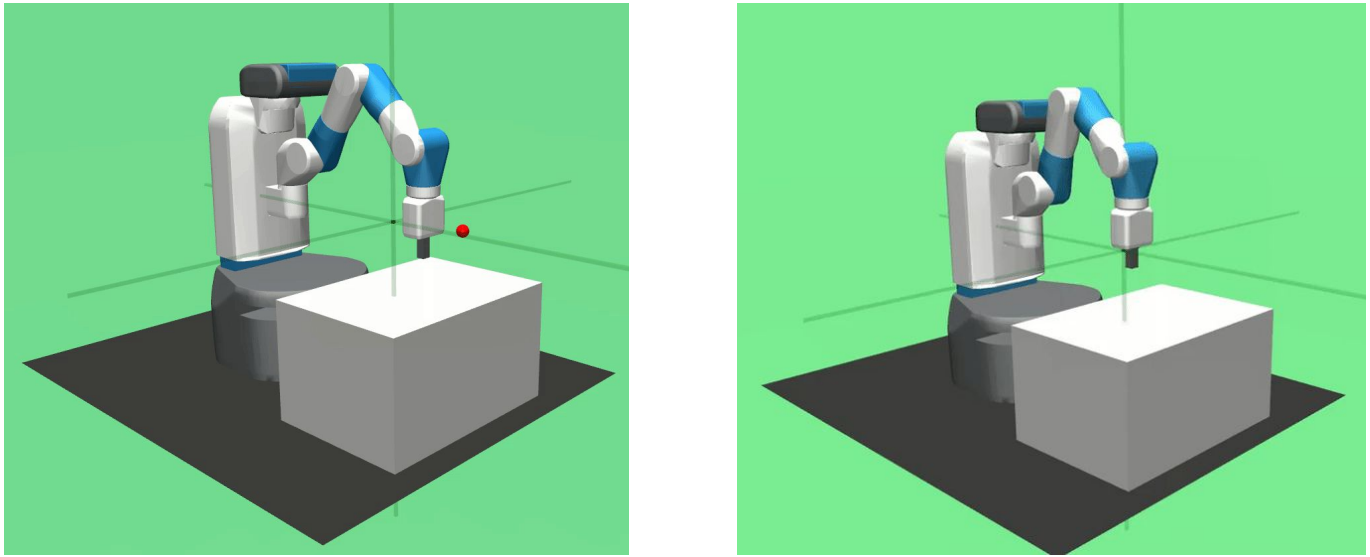


Figura 4.1: Agente PPO *baseline* (esquerda) vs. agente PPO + motivação intrínseca (direita)

Fetch Reach

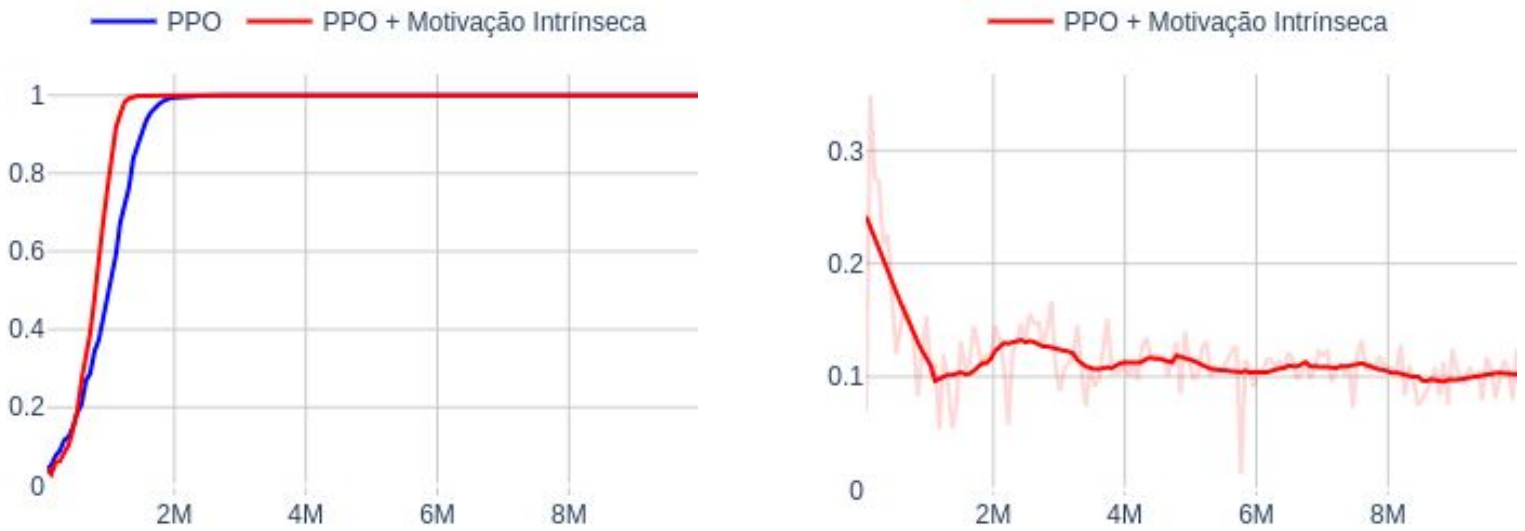


Figura 4.2: Razão de sucessos (esquerda) e recompensa intrínseca (direita).

Fetch Reach

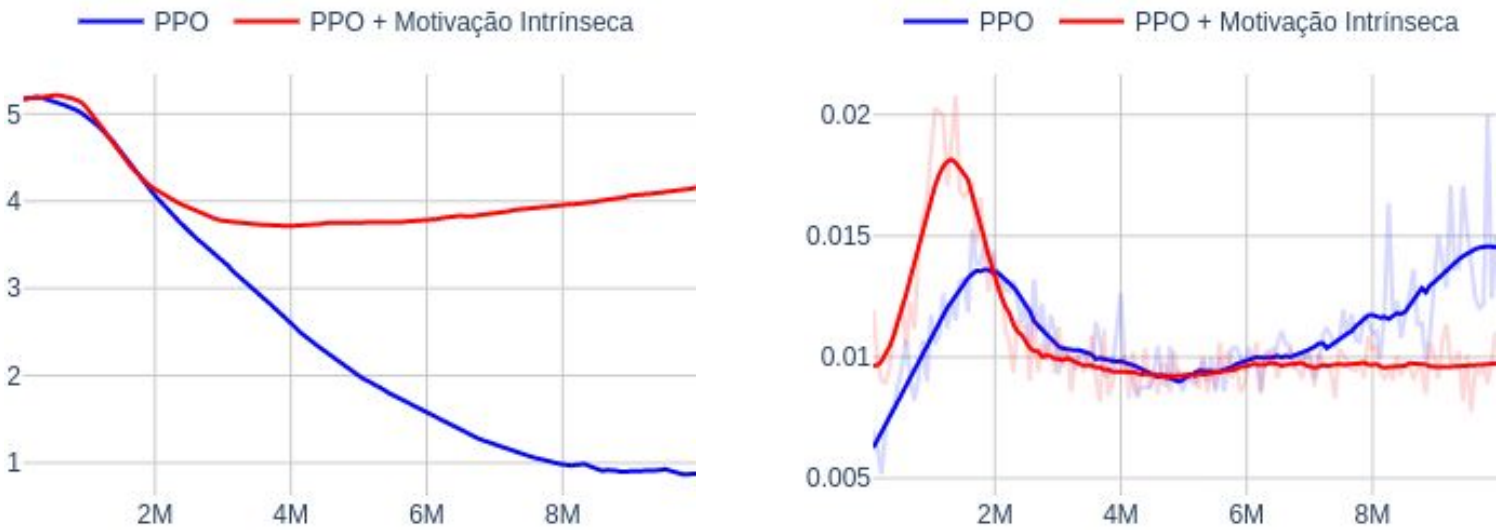


Figura 4.3: Entropia da política (esquerda) e divergência KL entre atualizações (direita).

Fetch Push

30M iterações

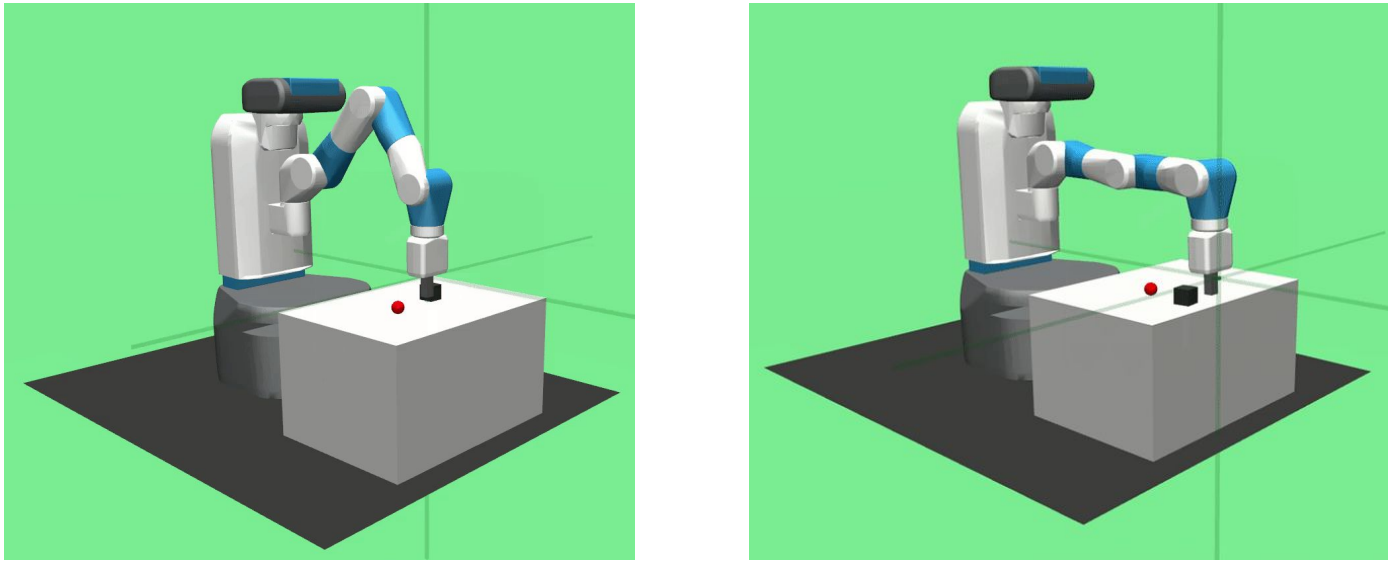


Figura 4.4: Agente PPO *baseline* (esquerda) vs. agente PPO + motivação intrínseca (direita)

Fetch Push

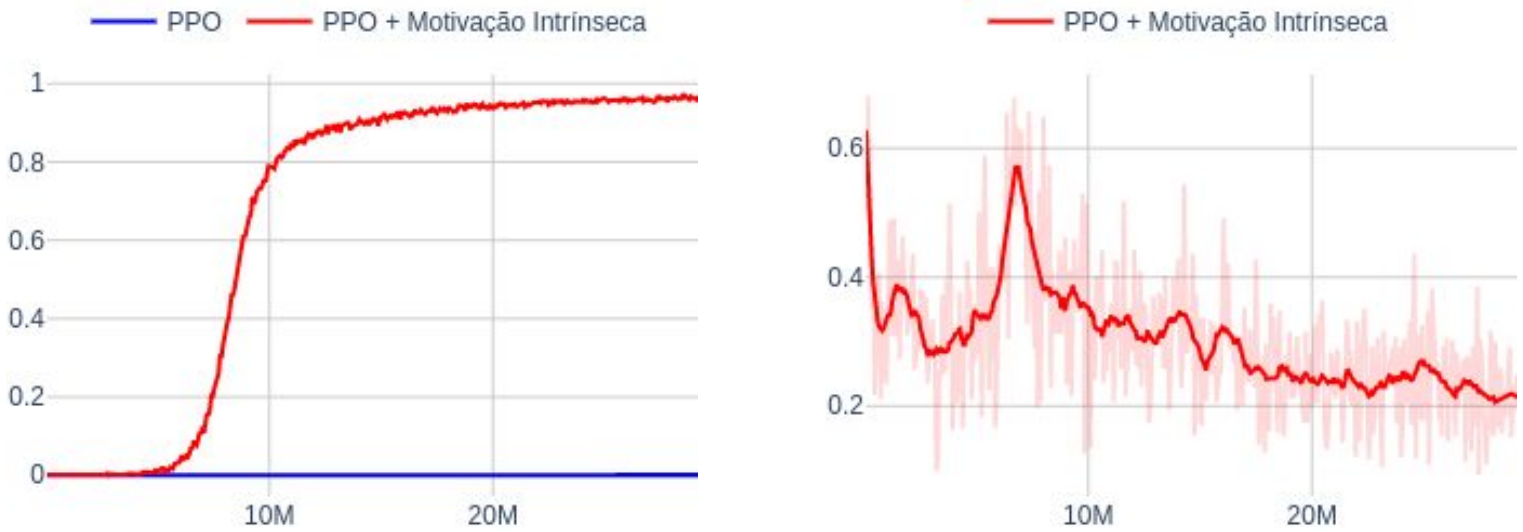


Figura 4.5: Razão de sucessos (esquerda) e recompensa intrínseca (direita).

Fetch Push

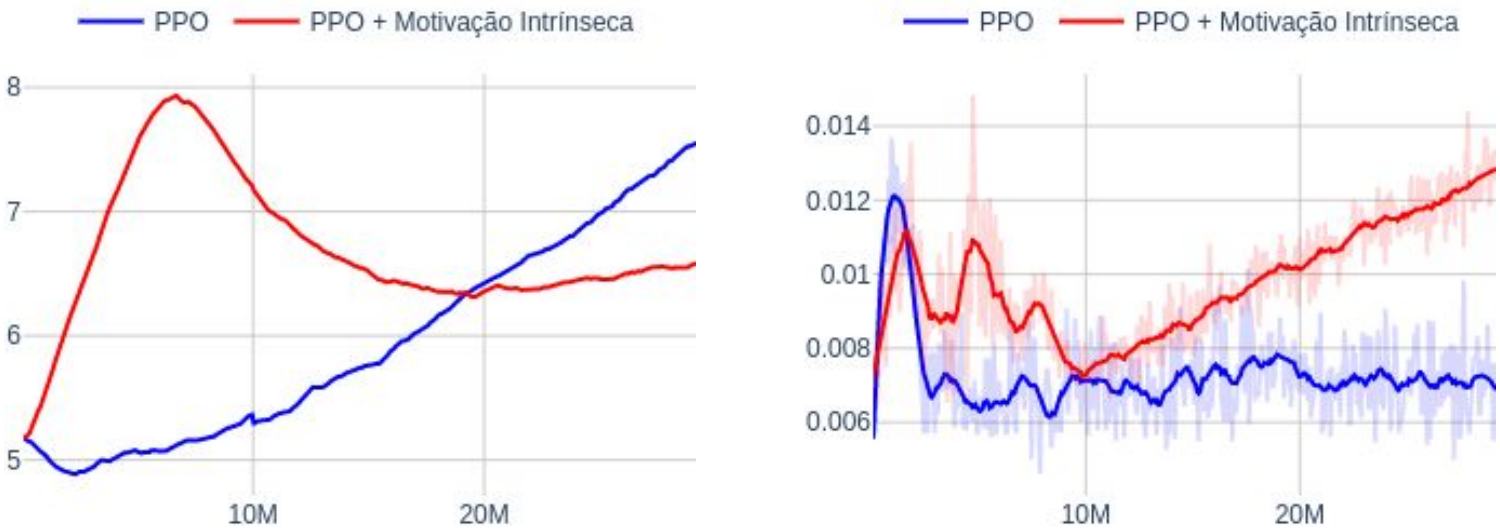


Figura 4.6: Entropia da política (esquerda) e divergência KL entre atualizações (direita).

Fetch Push

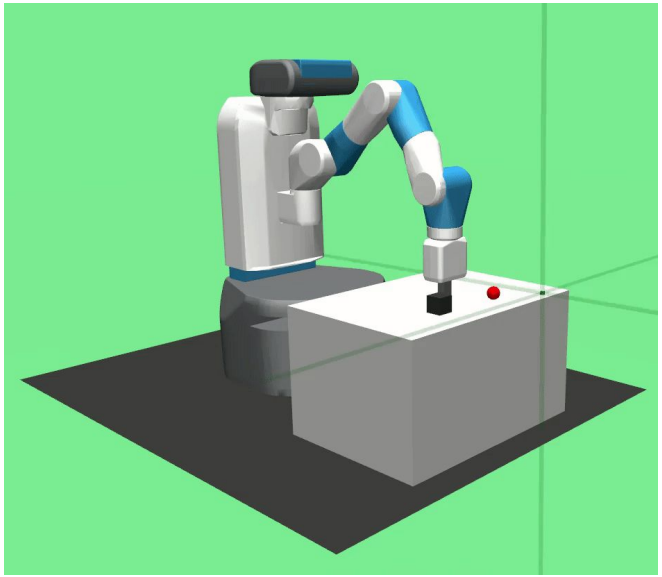


Figura 4.7: Agente com motivação intrínseca em aproximadamente **5M iterações**.

Fetch Pick and Place

30M iterações

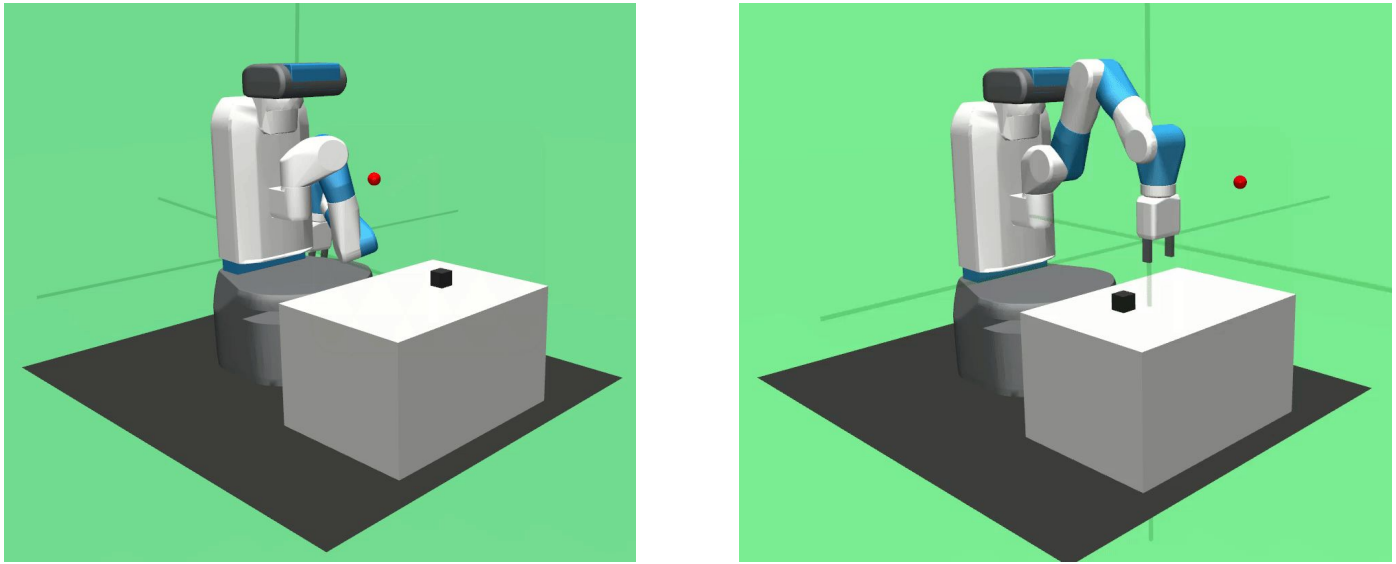


Figura 4.8: Agente PPO *baseline* (esquerda) vs. agente PPO + motivação intrínseca (direita)

Fetch Pick and Place

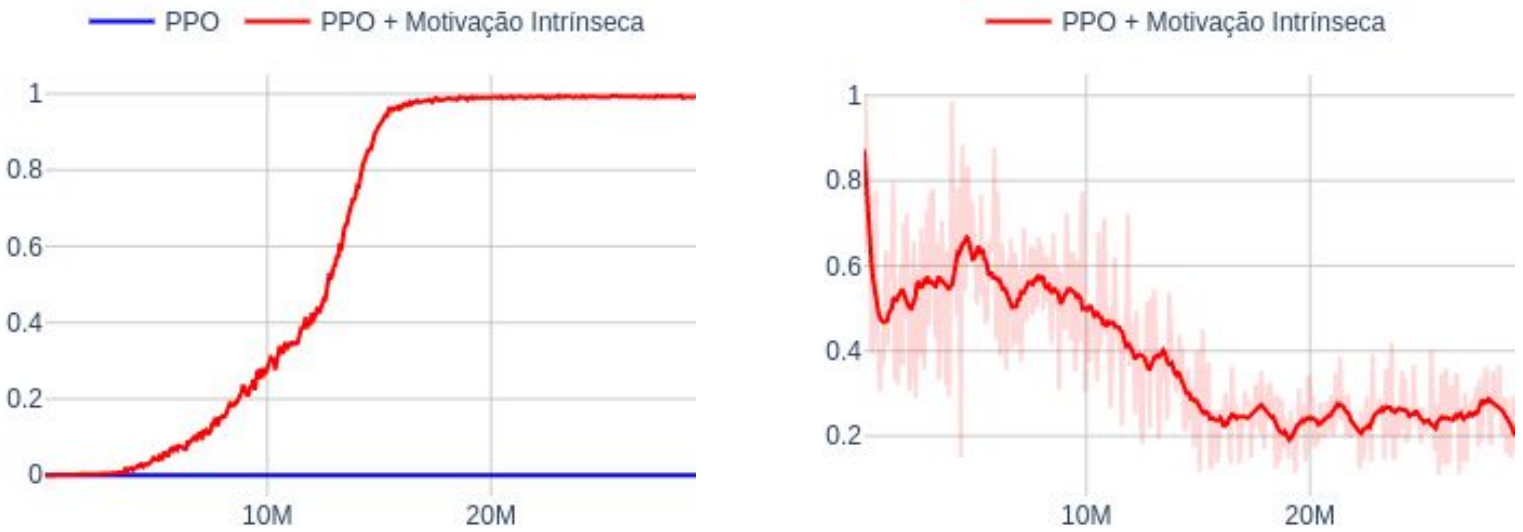


Figura 4.9: Razão de sucessos (esquerda) e recompensa intrínseca (direita).

Fetch Pick and Place

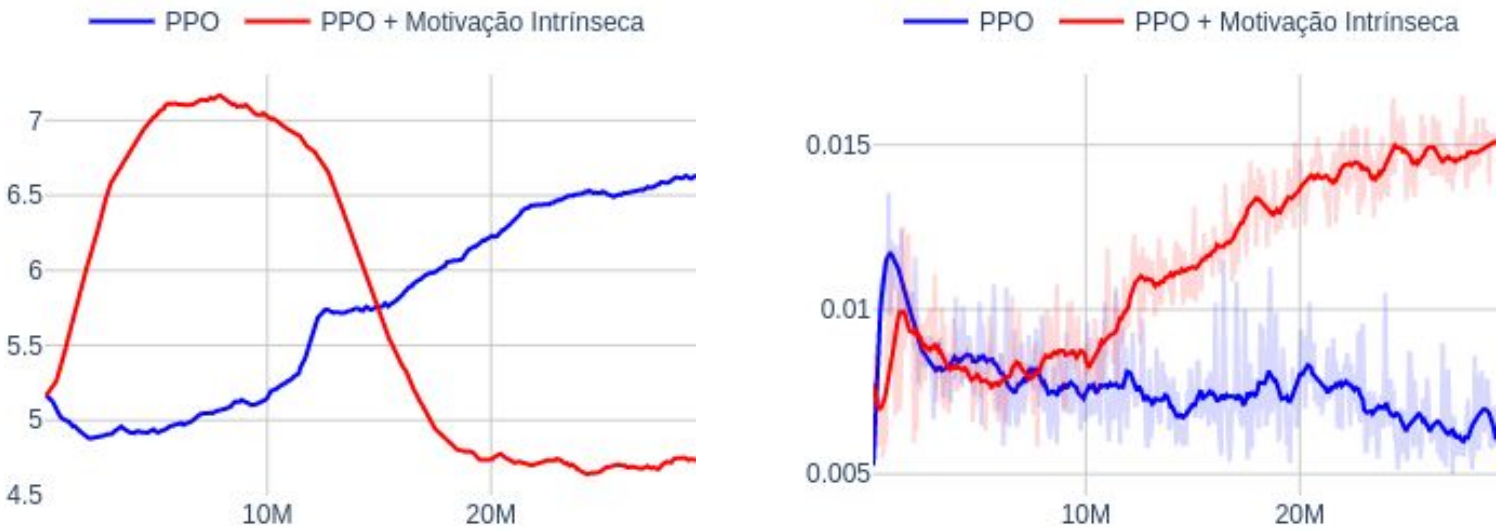


Figura 4.10: Entropia da política (esquerda) e divergência KL entre atualizações (direita).

Fetch Pick and Place

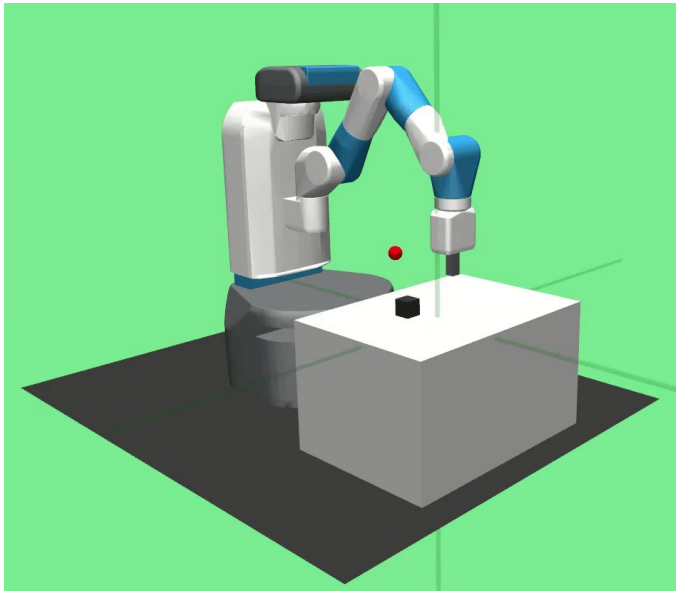


Figura 4.11: Agente com motivação intrínseca em aproximadamente **15M iterações**.

Considerações Gerais

Algoritmo	Fetch Reach	Fetch Push	Fech Pick and Place
PPO *baseline*	100%	0%	0%
PPO + motivação intrínseca	100%	96%	99%

Tabela 4.1: Razão de sucessos dos algoritmos em cada ambiente.

5 Conclusões

Conclusões

- Motivação intrínseca incentiva políticas exploratórias complexas
  - Onde a exploração não direcionada não consegue
- Informações sobre objetos influenciados pelo agente no ambiente são cruciais
- Tendência a políticas exploratórias mesmo após convergência (crescente entropia)

Trabalhos Futuros

- Investigar impacto na capacidade de generalização em ambientes **multitarefa** e **meta-aprendizado**
- Investigar impacto na capacidade **transferência** de aprendizado
- Aplicações em sistema **multi-agente**
- Aplicações em ambientes cuja observação é através de ***pixels***

Obrigado

bryanoliveira@inf.ufg.br

Dúvidas ou sugestões?

FUNAPE
Fundação de Apoio à Pesquisa - UFG

INSTITUTO DE INFORMÁTICA
UFG

UFG
UNIVERSIDADE FEDERAL DE GOIÁS