Rapport IAR TD3

Léa Movsessian, 28624266
Max Maiche, 21312637

Sorbonne Université

2024

Introduction

Dans ce rapport, nous évaluons une implémentation personnalisée de l'algorithme TD3 (*Twin Delayed Deep Deterministic Policy Gradient*) sur l'environnement `LunarLanderContinuous-v2`. Nous comparons notre implémentation avec deux bibliothèques populaires : *Stable Baselines 3* et *Tianshou*. L'objectif est de comparer les courbes d'apprentissage et d'analyser les éventuelles différences dans les performances.

Méthodologie

Nous avons d'abord implémenté l'algorithme TD3 en utilisant les hyperparamètres que nous avons trouvés dans la documentation de diverses bibliothèques. Ensuite, nous avons exécuté notre version de TD3 sur `LunarLanderContinuous-v2` avec 10 seeds différentes pour capturer la variabilité des performances. De plus, nous avons utilisé les implémentations TD3 de *Stable Baselines 3* et *Tianshou* pour effectuer des comparaisons similaires en termes de courbes d'apprentissage et de stabilité.

Les hyperparamètres principaux sont les suivants :

- Taux d'apprentissage de l'actor : 10^{-3}
- Taux d'apprentissage du critic : 10^{-3}
- Facteur de discount γ : 0.99
- Tau (facteur de mise à jour lente des cibles) : 0.05
- Noise de la politique : 0.1
- Batch size : 64

Nous avons ensuite produit des courbes d'apprentissage basées sur la moyenne des récompenses accumulées au cours du temps et avons également analysé la loss de l'actor au fil des itérations.

Résultats avant ajustement

Courbes d'apprentissage de notre implémentation

Nous avons tracé les courbes d'apprentissage en termes de récompense moyenne pour chaque épisode. La figure 1 montre les résultats pour notre implémentation TD3 avec 10 seeds différentes. Comme on peut le voir, la performance fluctue avec des récompenses moyennes atteignant environ -100 après 1 million d'itérations.

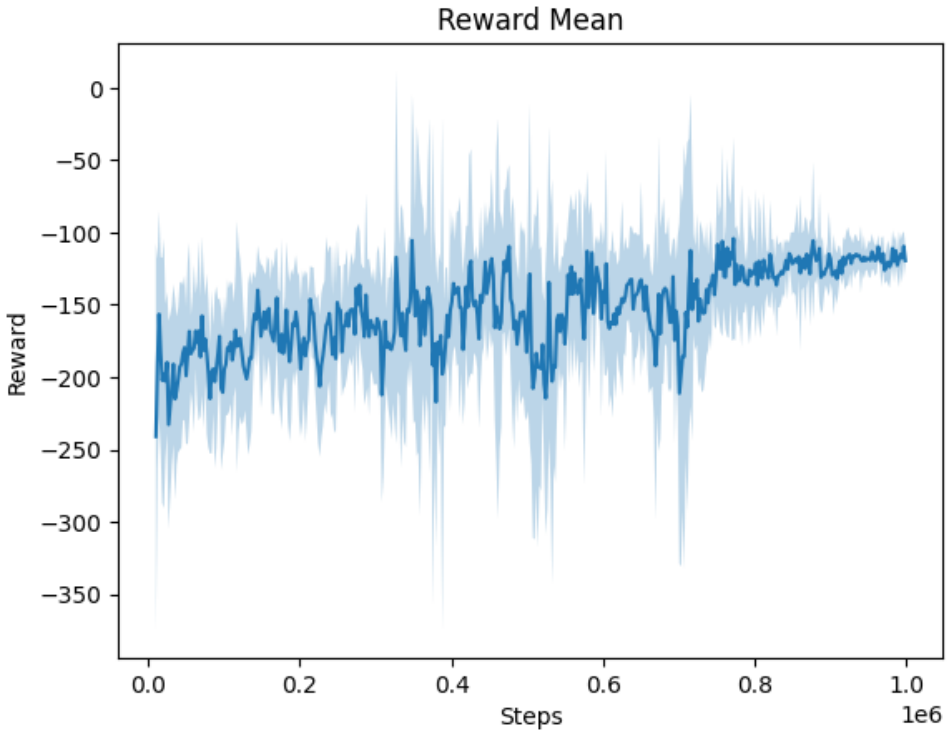

Figure 1: Courbe d'apprentissage de notre implémentation TD3 sur `LunarLanderContinuous-v2`.

Comparaison avec *Stable Baselines 3* et *Tianshou*

Les courbes d'apprentissage obtenues à partir de *Stable Baselines 3* et *Tianshou* sont présentées respectivement dans les figures 2 et 3. Nous avons également inclus un graphique comparatif (Figure 4) qui montre la récompense moyenne sur SB3 et BBRL. Nous constatons que *Stable Baselines 3* atteint une meilleure récompense moyenne plus rapidement, tandis que *Tianshou* atteint un plateau vers 200 000 steps. Pour Tianshou, le OffPolicyTrainer effectue une moyenne de reward à chaque epoch, ce qui rend les données plus lisses mais empêche une comparaison directe (sur un même graphe) avec les deux autres algorithmes.

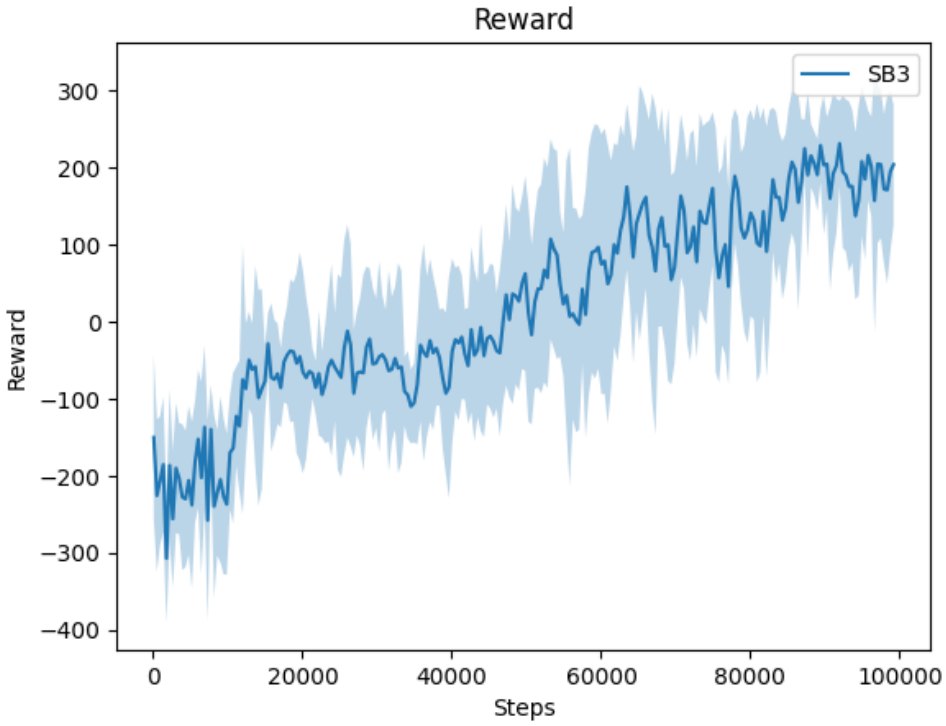

Figure 2: Courbe d'apprentissage de TD3 avec *Stable Baselines 3*.

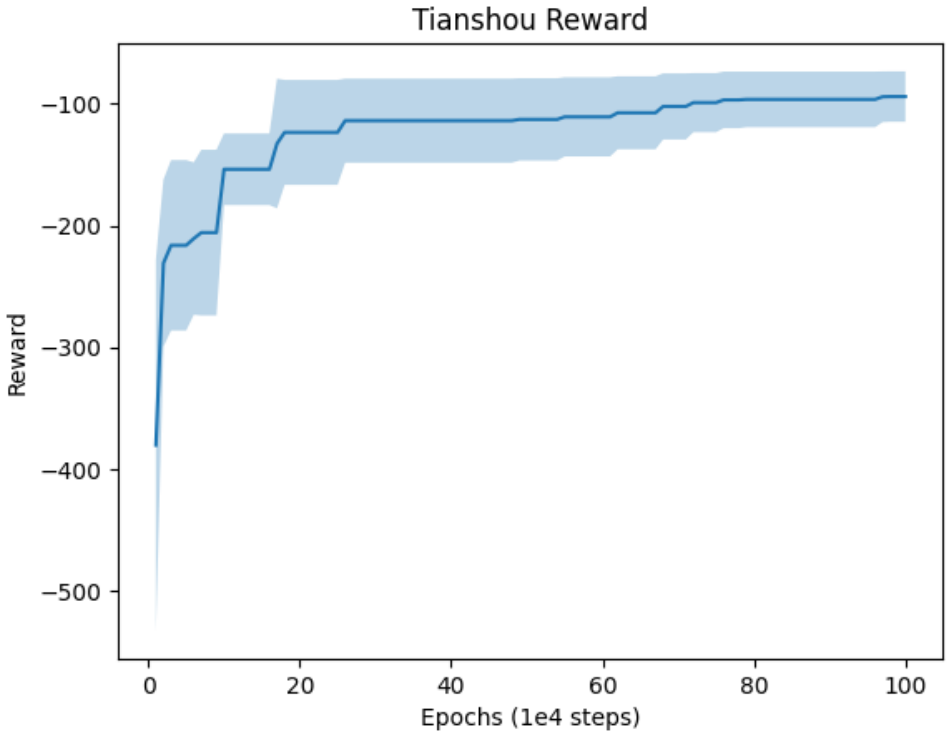

Figure 3: Courbe d'apprentissage de TD3 avec *Tianshou*.

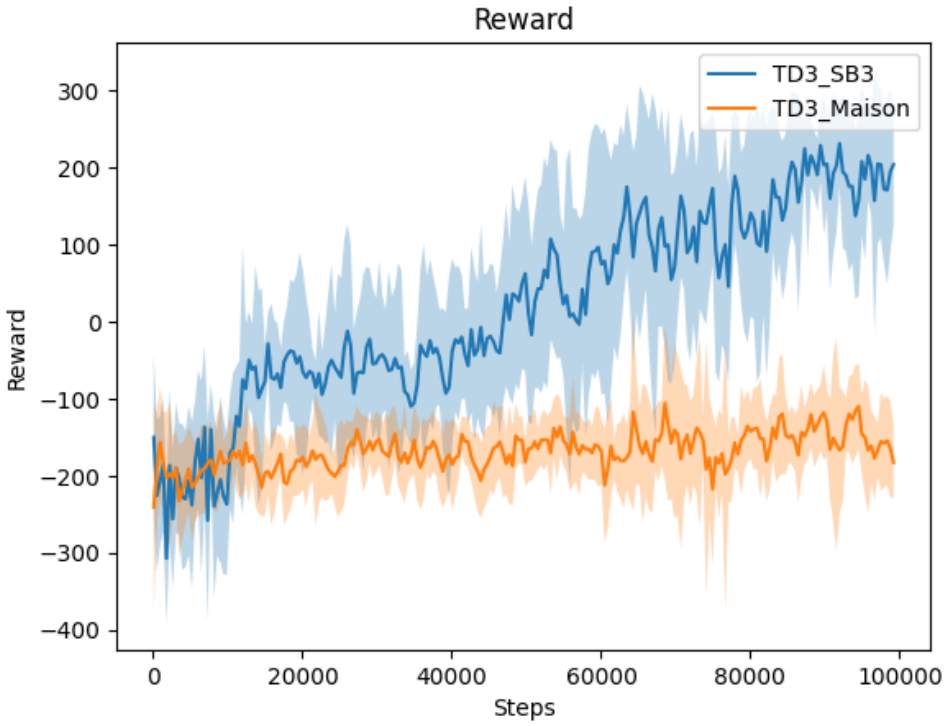

Figure 4: Comparaison des courbes d'apprentissage entre notre implémentation et *Stable Baselines 3*.

Conclusion

Dans ce projet, nous avons comparé notre implémentation de TD3 avec celles de *Stable Baselines 3* et *Tianshou*. Après avoir observé des différences notables dans les performances, nous avons identifié certaines divergences dans les hyperparamètres et les stratégies d'implémentation.

Dépôt Git

Voici le lien de notre dépôt Github public : Github repository