Rapport micro projet IAR

Étudiants :
Baris KAFTANCIOGLU : 28711733
Paul-Tiberiu IORDACHE : 28706827

Encadré par :
Olivier SIGAUD

9 octobre 2024

Table des matières

1 Introduction

Le but de ce rapport est de comparer notre version de l'algorithme TD3, implémentée dans la librairie BBRL, avec d'autres versions de l'algorithme implémentées dans différentes librairies. De plus, nous avons choisi de mener nos études dans les même environnements, à savoir LunarLanderContinuous-v2 et BipedalWalker-v3.

Avant de commencer l'étude, nous tenons à préciser que le code est disponible sur notre dépôt GitHub : https://github.com/PaulTiberiu/TD3_study.

2 Comparaison de notre algorithme TD3 avec la version des librairies SB3 et CleanRL

2.1 Réglage des hyperparamètres

Nous avons décidé de comparer notre algorithme avec les hyperparamètres qui sont considérés comme les plus performants dans SB3. La source de nos hyperparamètres s'inspire fortement de ceux proposés par les développeurs de Hugging Face, parce que ces hyperparamètres sont assez adaptés dans les environnements que nous avons choisi. Le lien est le suivant : https://huggingface.co/sb3/td3-LunarLanderContinuous-v2.

Après avoir identifié certaines équivalences entre les hyperparamètres des librairies, nous avons finalement fixé nos hyperparamètres :

- Learning rate : 0.001
- Buffer size : 200000
- Discount factor : 0.98
- Learning starts : 10000
- Noise : 0.1
- Actor size : [400, 300]
- Critic size : [400, 300]
- Batch size : 64

Ensuite, nous tenons à préciser que certains hyperparamètres n'ont pas d'équivalent dans l'autre librairie. Nous avons essayé de trouver le meilleur compromis pour pouvoir comparer des apprentissages les plus proches possibles, mais normalement les hyperparamètres qui permettent cela sont ceux que nous avons définis ci-dessus. Un autre critère important dans nos choix a été le temps d'exécution de chaque run, que nous avons fixé à environ 20-30 minutes.

Pour pouvoir garantir ce temps d'exécution, en SB3 et CleanRL, nous avons décidé de lancer l'algorithme pendant 100000 timesteps. De l'autre côté, en BBRL, nous avons choisi de tester pendant 11000 époques, avec un eval_interval à 2000 et un n_steps de 100.

SB3 et CleanRL génèrent des courbes d'apprentissage en fonction des timesteps, tandis que BBRL fonctionne différemment en utilisant des époques et des intervalles d'évaluation, ainsi que des courbes basées sur les training steps. Pour établir une comparaison,

nous avons calculé le nombre total de timesteps dans BBRL à l'aide de la formule total_timesteps = n_steps * max_epochs. Les training steps sont définis par training_steps = (n_steps * max_epochs - learning_starts) / eval_interval. Cependant, en raison des différences de fonctionnement entre les librairies, il est impossible de comparer directement les résultats en utilisant une même unité sur l'axe des abscisses. Par conséquent, nous analyserons la récompense en fonction des timesteps dans SB3, des training steps dans BBRL et des episodes en CleanRL, car la courbe moyenne d'apprentissage a comme abscisse le nombre moyen des epiodes et pas les timesteps.

Dans la suite de notre analyse, nous avons décidé de réaliser un test statistique en faisant varier dix fois la seed et en calculant la moyenne des récompenses afin d'obtenir une courbe d'apprentissage plus stable, permettant de mieux observer les différences lors du processus d'apprentissage des deux versions de l'algorithme.

2.2 Comparaison dans l'environnement LunarLanderContinuous-v2

2.2.1 Performance de notre algorithme TD3 en BBRL

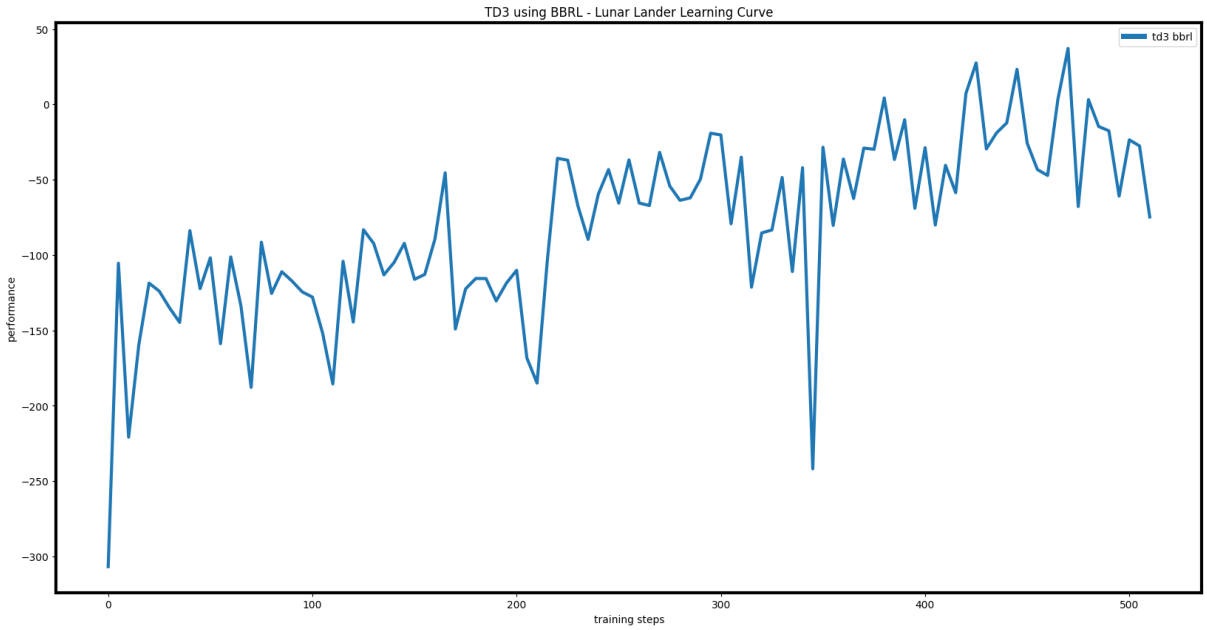

FIGURE 1 – Moyenne des courbes d'apprentissage en BBRL sur 10 runs avec 10 seeds différentes dans l'environnement LunarLanderContinuous-v2.

Comme nous pouvons le voir dans la figure 1, il est évident que, même après avoir fait une moyenne sur 10 runs avec des seeds différentes, la courbe présente des variations au cours de l'apprentissage. Cela peut être dû à un nombre insuffisant d'epochs ou à un réglage imparfait des hyperparamètres. De plus, la variance entre les récompenses peut également s'expliquer par le fait que l'algorithme TD3, dans cet environnement, dépend fortement du bon choix d'hyperparamètres, et s'ils ne sont pas assez fins, ils peuvent entraîner de l'instabilité pendant l'entraînement.

De plus, étant donné que nous avons choisi de tester notre algorithme dans la librairie BBRL avec des hyperparamètres optimisés pour la librairie SB3, et compte tenu des différences de conception entre ces librairies, des instabilités durant l'apprentissage sont inévitables. Ainsi, une étude plus approfondie des hyperparamètres dans BBRL aurait pu produire de meilleurs résultats.

Malgré tous ces inconvénients, nous pouvons tout de même constater que l'algorithme converge et parvient, vers la fin de l'apprentissage, à atteindre des valeurs autour de 20. Ainsi, comme précisé plus haut, un réglage plus fin des hyperparamètres ainsi qu'un temps d'apprentissage plus long auraient permis à l'algorithme d'obtenir de meilleurs résultats.

2.2.2 Performance de l'algorithme TD3 en SB3

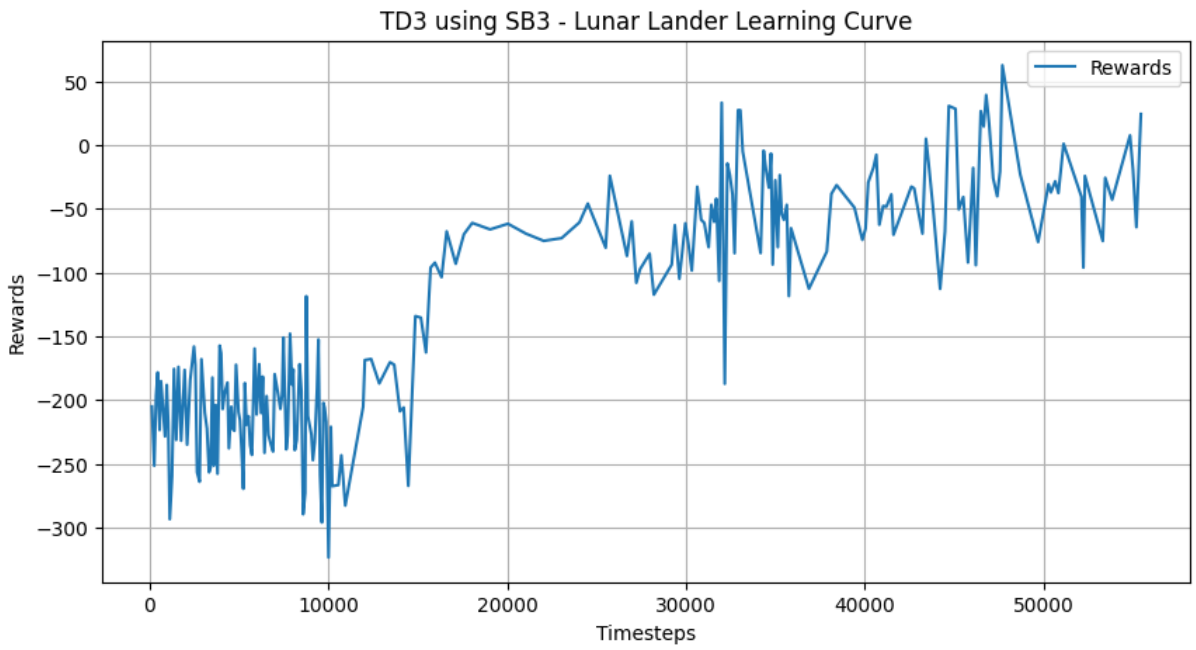

FIGURE 2 – Moyenne des courbes d'apprentissage en SB3 sur 10 runs avec 10 seeds différentes dans l'environnement LunarLanderContinuous-v2.

Dans la figure 2, nous pouvons déjà constater l'importance accrue de l'hyperparamètre learning starts dans la librairie SB3. Comme nous l'avons fixé à 10000, nous voyons immédiatement qu'avant 10000, l'algorithme explore et ne converge pas encore. Après cette valeur, nous observons une convergence très rapide vers des valeurs autour de -50.

En comparant avec la courbe d'apprentissage de BBRL, nous constatons que cet algorithme converge plus rapidement entre 100000 et 200000 timesteps, avant d'avoir une croissance plus lente, comme observé dans la courbe d'apprentissage de BBRL. En revanche, dans SB3, nous remarquons des instabilités moins prononcées entre les récompenses, ce qui indique que les hyperparamètres sont mieux adaptés à cette librairie, comme c'est le cas.

Pour la convergence finale, nous pouvons tirer la même conclusion que dans BBRL, car les récompenses se trouvent également autour de 20. Comme nous l'avons précisé

précédemment, l'idéal dans SB3 aurait été de lancer sur 300000 timesteps, selon les développeurs de Hugging Face pour obtenir des meilleurs résultats, mais cela aurait triplé notre temps d'exécution. De plus, nous pensons qu'une valeur plus élevée pour le batch size, comme 128 ou 256, aurait donné de meilleurs résultats, car cela aurait permis des mises à jour plus stables et réduit les instabilités pendant l'apprentissage. En revanche, une convergence plus lente aurait été attendue, ce qui aurait évidemment nécessité un plus grand nombre de timesteps.

Pour conclure, malgré les différences observées dans nos courbes d'apprentissage, nous pouvons finalement dire qu'à la fin de l'apprentissage, dans les deux librairies, nous obtenons des valeurs similaires, ce qui démontre que notre implémentation de TD3 dans BBRL est correcte et peut être utilisée comme les versions de TD3 des autres librairies.

2.2.3 Performance de l'algorithme TD3 en CleanRL

Pour disposer d'une source de comparaison supplémentaire, nous avons décidé de comparer notre algorithme TD3 sous BBRL avec une autre librairie d'apprentissage par renforcement, en l'occurrence CleanRL. Tout d'abord, il est important de préciser que cette étude a été réalisée avec les mêmes hyperparamètres (définis au début de la section 2.1).

Le fonctionnement de cette bibliotheque est similaire a SB3 et les courbes d'apprentissage sont faites par rapport au nombre de timesteps. Par contre, durant le 100000 timesteps, CleanRL effecute entre 200 et 1000 episodes et la courbe moyenne d'apprentissage, comme avant, faite sur 10 seeds differentes, a comme abscisse le nombre moyen des episodes.

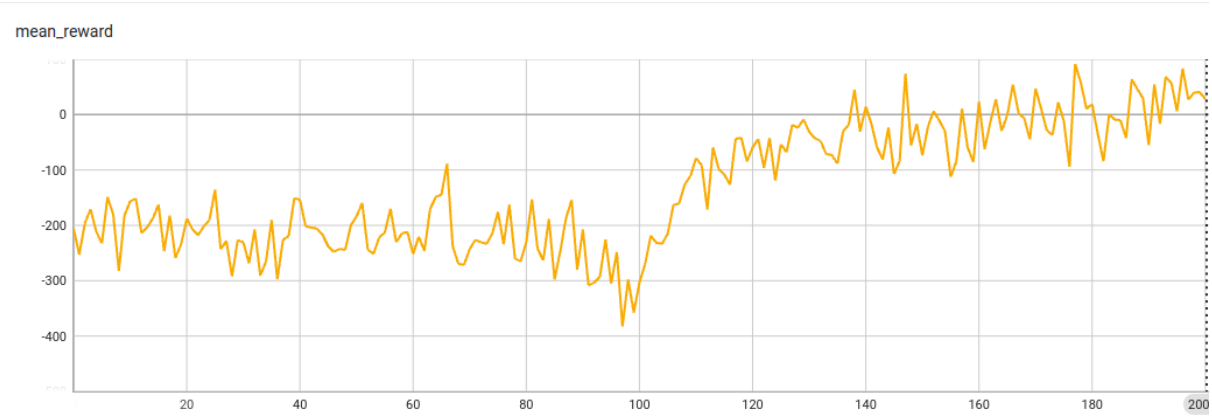

FIGURE 3 – Moyenne des courbes d'apprentissage en CleanRL sur 10 runs avec 10 seeds différentes dans l'environnement LunarLanderContinuous-v2.

Comme pour BBRL et contrairement à SB3, au début de l'apprentissage, nous observons que l'importance de l'hyperparamètre learning starts n'est pas suffisamment élevée. De plus, une augmentation brusque, similaire à celle observée avec SB3, apparaît vers le 100e épisode, suivie d'une croissance linéaire. Par ailleurs, en termes de progression, cette librairie est plus proche de SB3 que de BBRL, qui présente une croissance plus lente au cours de l'apprentissage.

Comme dans les deux librairies précédentes, on observe qu'à la fin de l'apprentissage, les rewards convergent vers une valeur de 20. Cela constitue donc un autre argument en

faveur du fait que notre version de TD3 dans BBRL est correctement implémentée et peut produire de bons résultats.

2.3 Comparaison dans un nouvel environnement : BipedalWalker-v3.

Afin de mieux comparer les algorithmes, nous avons décidé de les tester dans un nouvel environnement, BipedalWalker-v3. Dans cet environnement, l'objectif est de faire marcher un robot à deux pattes. Pour réaliser cette tâche, nous avons utilisé les algorithmes TD3 implémentés dans BBRL, SB3 et CleanRL.

2.3.1 Performance de notre algorithme TD3 en BBRL

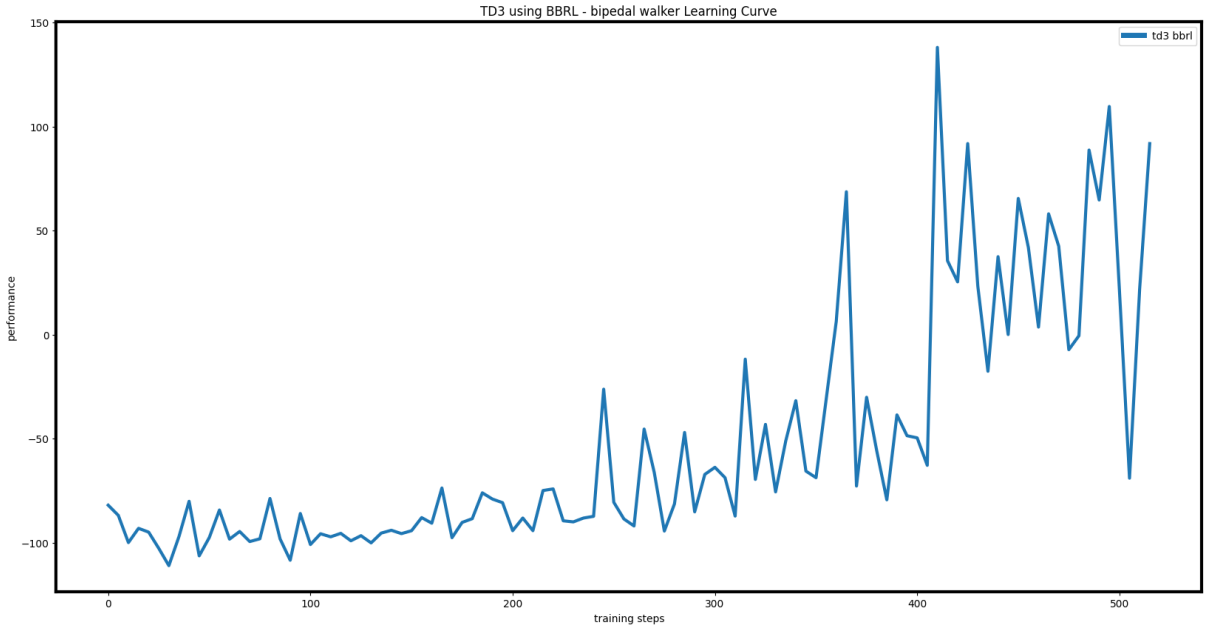

FIGURE 4 – Moyenne des courbes d'apprentissage en BBRL sur 10 runs avec 10 seeds différentes dans l'environnement BipedalWalker-v3.

Au début de l'entraînement, les performances sont assez faibles. C'est une phase typique où l'agent n'a pas encore appris à marcher correctement. On observe une stabilité dans les premières étapes, ce qui est habituel lorsque l'agent explore l'environnement et teste différentes actions qui peuvent être moins performantes.

Les pics et les chutes brusques peuvent indiquer que l'agent est toujours en train de stabiliser sa politique d'apprentissage. Il se peut qu'il ait trouvé une bonne stratégie pour certaines parties de l'environnement, mais qu'il soit encore en train d'explorer d'autres aspects, ce qui provoque des baisses de performance.

Pour éviter ces instabilités, il est nécessaire d'ajuster les hyperparamètres. Une autre cause possible de cette instabilité pourrait être la durée de la phase d'apprentissage.

2.3.2 Performance de notre algorithme TD3 en SB3

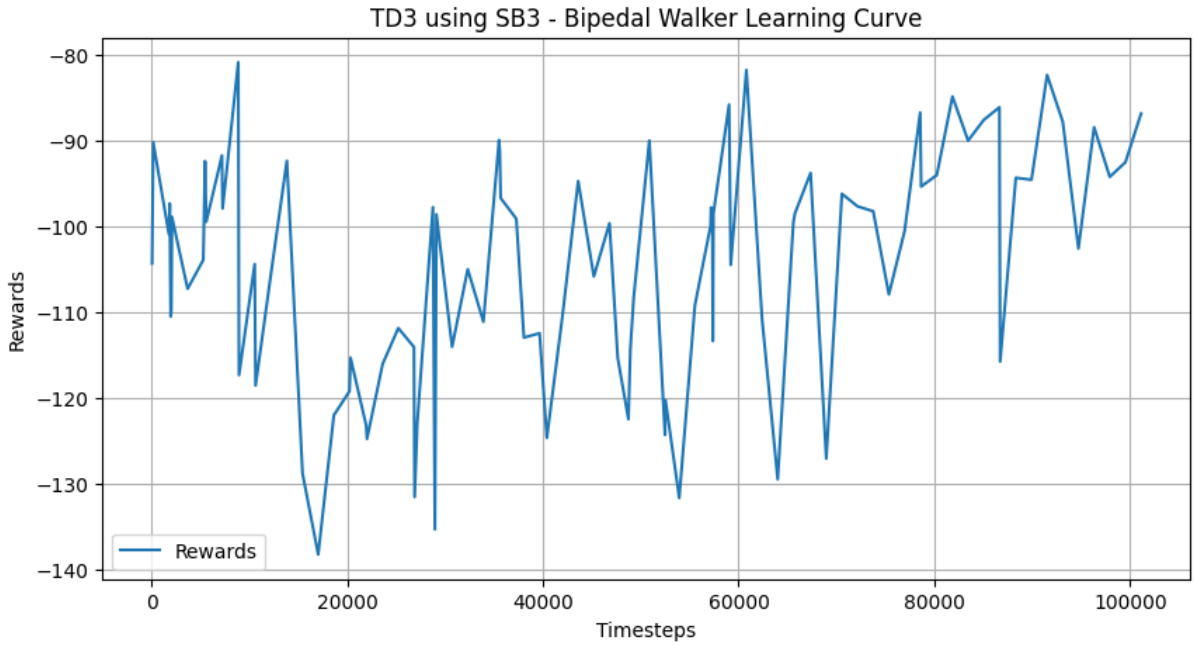

FIGURE 5 – Moyenne des courbes d'apprentissage en SB3 sur 10 runs avec 10 seeds différentes dans l'environnement BipedalWalker-v3.

On observe des fluctuations importantes dans les récompenses tout au long de la courbe, ce qui montre que l'agent explore encore et a du mal à converger vers une stratégie optimale. Bien qu'il puisse parfois obtenir de meilleures récompenses, ses performances restent loin d'être optimales (une récompense proche de zéro ou positive serait idéale).

Même si l'agent parvient parfois à obtenir de meilleures récompenses, elles sont encore loin d'être optimales.

Encore une fois, les hyperparamètres jouent un rôle crucial dans l'apprentissage par renforcement. L'algorithme de SB3 semble plus sensible aux hyperparamètres que celui de BBRL.

En conclusion, cette courbe montre que l'agent TD3 utilisant BBRL s'adapte mieux à la tâche du BipedalWalker et parvient à obtenir des récompenses positives bien plus rapidement que celui utilisant Stable-Baselines3. Les fortes variations observées en phase avancée suggèrent qu'il peut encore être optimisé pour plus de stabilité, mais les performances générales sont prometteuses.

2.3.3 Performance de l'algorithme TD3 en CleanRL

Dans le cas de cet environnement, durant le 100000 timesteps, CleanRL effecute en moyenne 120 et 500 episodes et la courbe moyenne d'apprentissage, comme avant, faite sur 10 seeds differentes, a comme abscisse le nombre moyen des episodes.

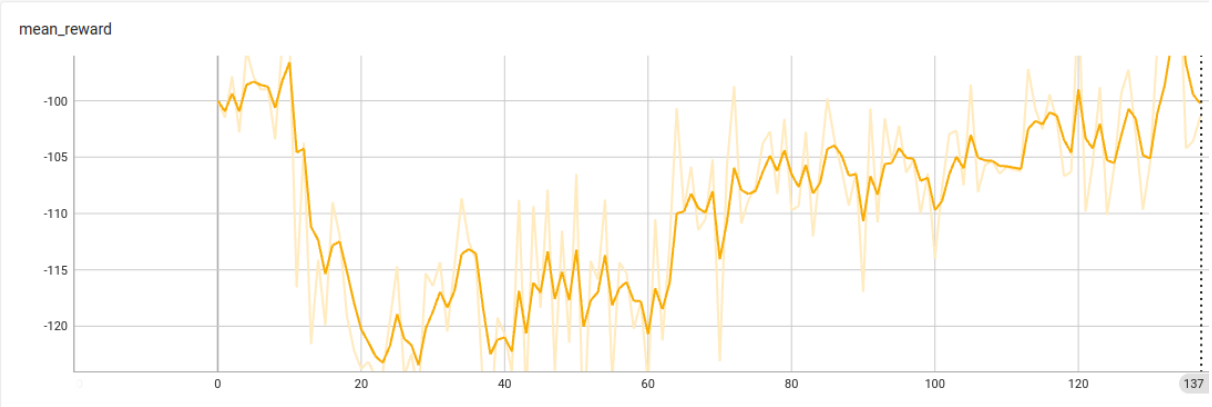

FIGURE 6 – Moyenne des courbes d'apprentissage en CleanRL sur 10 runs avec 10 seeds différentes dans l'environnement BipedalWalker-v3.

Concernant TD3 de CleanRL, nous observons des performances relativement similaires à celles de SB3. Cependant, la courbe d'apprentissage apparaît légèrement plus line. Une fois de plus, cela souligne l'importance des hyperparamètres, car notre apprentissage n'a pas réussi à converger vers les valeurs optimales.

Contrairement à BBRL, qui s'est bien adapté à ce nouvel environnement avec les mêmes hyperparamètres, nous pouvons à nouveau souligner l'importance des hyperparamètres ainsi que les différences de conception entre les bibliothèques d'apprentissage par renforcement. Ainsi, dans le cas de l'environnement BipedalWalker, BBRL montre une meilleure adaptation avec le jeu d'hyperparamètres donné.

En conclusion, bien que les algorithmes soient identiques, les différences de conception entre les bibliothèques et les variations des hyperparamètres peuvent avoir un impact significatif sur la performance d'un algorithme d'apprentissage dans un environnement donné. Ainsi, certains algorithmes provenant de certaines bibliothèques peuvent se révéler mieux adaptés que le même algorithme d'une autre bibliothèque.

3 Conclusions

Ce rapport a mis en évidence l'importance cruciale des hyperparamètres et des choix de librairies dans l'implémentation d'algorithmes d'apprentissage par renforcement, notamment l'algorithme TD3. À travers nos expériences dans les environnements BipedalWalker-v3 et LunarLanderContinuous-v2, nous avons observé des variations significatives de performance selon les configurations choisies.

Nos résultats montrent que même des algorithmes identiques peuvent se comporter différemment en fonction de leur environnement d'exécution, soulignant la nécessité d'une attention particulière lors du réglage des hyperparamètres et du choix de la librairie. Cette étude met également en lumière l'importance d'une évaluation rigoureuse et comparative des implémentations afin de tirer parti des meilleures pratiques dans le domaine de l'apprentissage par renforcement.