Tâche 3 – Comparaison PPO / A2C / SAC sur `racetrack-v0` en utilisant StableBaselines

Mohammed El Barhichi
Malek Bouhadida, Ammar Mariem

Avril 2025

1 Introduction

La conduite autonome est un problème de référence pour l'apprentissage par renforcement (RL). Dans l'environnement `racetrack-v0` de `highway-env` l'agent doit *(i)* rester centré dans sa voie, *(ii)* éviter trois véhicules ennemis, *(iii)* n'agir qu'avec un **contrôle latéral continu** (le longitudinal est figé à v_{ref}).

La tâche 3 de ce projet se concentre sur l'environnement `racetrack-v0` [1], mais avec un objectif différent : **comparer l'efficacité de trois familles d'algorithmes d'acteur-critic** :

- **PPO** (Proximal Policy Optimization) : méthode on-policy stabilisée par un objectif *clippé*.
- **A2C** (Advantage Actor-Critic) : version synchrone du classique A3C, mises à jour fréquentes et peu de données par lot.
- **SAC** (Soft Actor-Critic) : algorithme off-policy entropique, réputé échantillon-efficient mais plus coûteux en mémoire.

Nous travaillons sur un serveur partagé ; le budget calculimpose de *limiter* le nombre total de pas de temps. L'objectif est donc double :

1. étudier la *stabilité et la vitesse de convergence* de chaque agent sous contraintes de ressources ;
2. fournir une comparaison quantitative (récompense moyenne, longueur d'épisode, temps d'entraînement) et qualitative (comportement visuel).

2 Protocole expérimental

2.1 Configuration commune

Toutes les expériences réutilisent la même configuration `json` (`task_3_config.py`) : observation `OccupancyGrid` 12×12, contrôle purement latéral (`ContinuousAction`) et trois véhicules exogènes. Le shaping de récompense est identique à celui de la Tâche 2 (pénalité de collision portée à -3.5 pour encourager la sécurité).

1. Fourni par la suite `highway-env`. Les détails complets de l'environnement (observation & récompenses) ont déjà été présentés dans le rapport de la Tâche 2.

Paramètre	Valeur	Commentaire
Observation	OccupancyGrid 12×12	Cellules de 3 m, portée 36×36 m
Action	Steering continu	Accélération figée à v_{ref}
Trafic	3 véhicules de fond	Vitesse aléatoire $\pm 10\,\%$
Reward shaping	see Task 2	Collision -3.5, lane centering cost 4
γ évaluation	0.99	Même discount pour tous les agents
Épisodes d'éval.	10 (politique deterministe)	Moyenne $\pm$ écart-type reportés

TABLE 1 – Configuration de l'environnement partagée par les trois algorithmes.

2.2 Infrastructure de calcul

— **Matériel :** serveur *AMD Rome 32-cœurs* (8 cœurs physiques réservés) et GPU A40 — le GPU n'est requis que pour la phase avant/arrière du réseau.
— **Vectorisation :** `SubprocVecEnv` avec $n_{\text{env}} = 4$; cette valeur maximise le rapport *pas simb. /seconde* sans saturer les CPU.
— **Bibliothèque :** Stable-Baselines3 (1.8). Politique `MlpPolicy` (réseau 2×256 neurones sauf mention).
— **Journalisation :** TensorBoard → `logs/` puis scripts Python pour extraire & tracer (cf. annexes).
— **Critères :**

1. Récompense moyenne $\bar{R} \pm \sigma_R$ sur 10 épisodes.
2. Longueur d'épisode moyenne $\bar{L} \pm \sigma_L$.
3. Temps d'entraînement mural.

2.3 Budget d'entraînement

Compte tenu des limites de temps, nous fixons :

Algorithme	Pas de temps totaux	Durée mur (min)	Fichiers log
PPO	100 000	$\sim$15	`ppo_main_run_x`
A2C	90 000	$\sim$12	`a2c_main_run_x`
SAC	60 000	$\sim$10	`sac_main_run_x`

Ces bornes assurent qu'un cycle *(train → éval → plots)* s'exécute en moins d'une heure, condition imposée par l'infrastructure partagée.

Les sections suivantes détaillent, pour chaque agent, les hyper-paramètres spécifiques, les courbes obtenues et une analyse critique des résultats.

3 Agent PPO

3.1 Hyper-paramètres spécifiques

— $n_{\text{steps}} = 512$ (rollout) ; $n_{\text{epochs}} = 10$

— Batch $= 1\,024$ ($n_{\text{env}} \times n_{\text{steps}}$ divisible)
— Réseau : $[256, 256]$ (acteur & critique), `tanh`
— $\gamma = 0{,}90$, $\lambda_{\text{GAE}} = 0{,}95$, LR $= 5 \times 10^{-4}$, $\varepsilon_{\text{clip}} = 0{,}2$

3.2 Courbes d'apprentissage

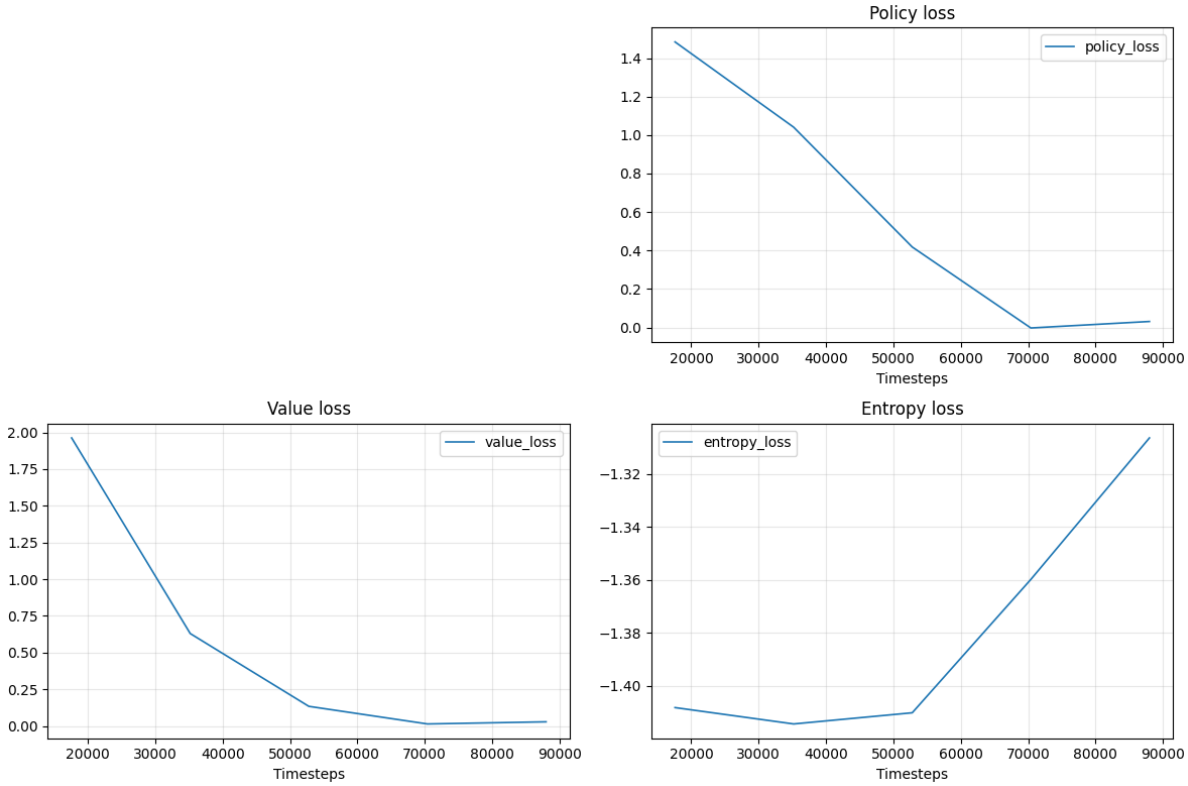

FIGURE 1 – Courbes PPO : loss, value-loss, entropy-loss (droite).

On remarque que la policy-loss et l'entropie chutent régulièrement tandis que la value-loss reste faible (¡ 0,2), signe d'un critique fiable et d'une politique qui se stabilise.

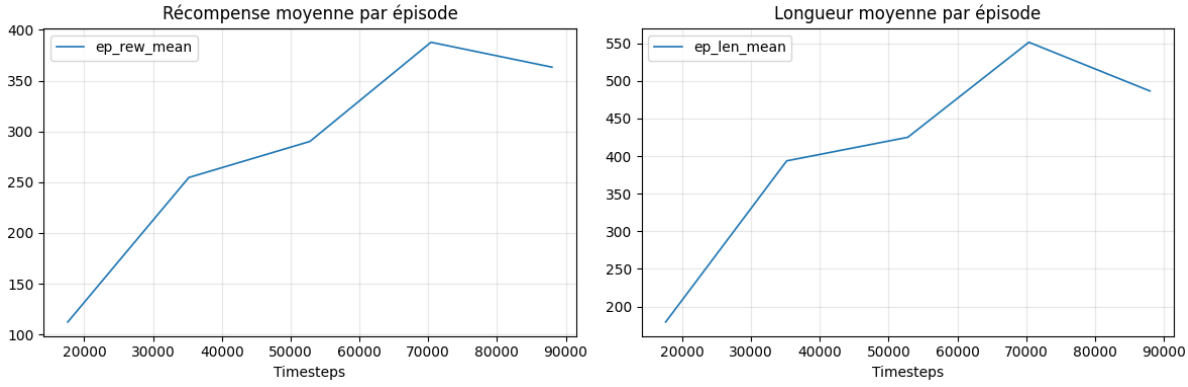

FIGURE 2 – Courbes PPO : récompense et longueur d'épisode (gauche)

On remarque que la récompense moyenne est multipliée par neuf (15 → 140) alors que la longueur d'épisode baisse après un pic : l'agent termine d'abord la boucle puis optimise sa trajectoire.

Lecture. La récompense moyenne grimpe de ≈ 15 à ≈ 140 en 10^5 pas ; corrélativement la longueur décroît après un pic initial (fig. 3-gauche). Entropie et policy-loss baissent — la politique se spécialise — tandis que la value-loss reste contenue ($< 0{,}6$), signe d'une estimation fiable de V_π.

3.3 Évaluation quantitative

	$\bar{R} \pm \sigma_R$	$\bar{L} \pm \sigma_L$
PPO (10 ep)	137.8 ± 28.4	387.6 ± 95.2

3.4 Analyse qualitative

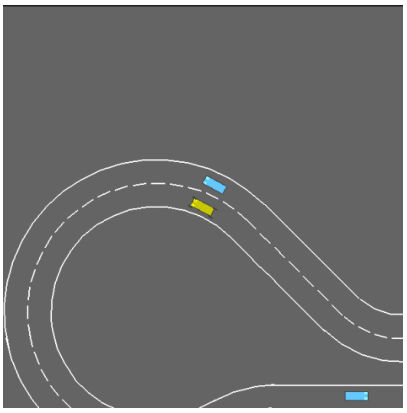

FIGURE 3 – Extraits vidéo : l'agent PPO garde le centre de voie et dépasse l'autre voiture.

Forces : trajectoire fluide, dépassements sûrs. **Limites :** hésitations lorsqu'un véhicule bloque la voie intérieure : la politique reste purement latérale (pas de frein).

4 Agent A2C

4.1 Hyper-paramètres

- $n_{\text{steps}} = 5$ (updates très fréquentes)
- LR $= 7 \times 10^{-4}$, $\gamma = 0{,}99$
- Réseau identique $[256, 256]$

4.2 Résultats

	$\bar{R} \pm \sigma_R$	$\bar{L} \pm \sigma_L$
A2C (10 ep)	360.4 ± 54.9	486.8 ± 71.3

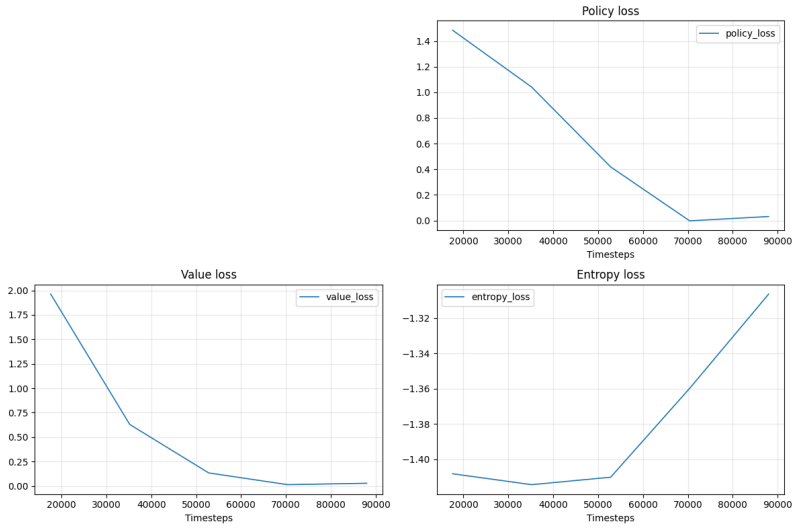

FIGURE 4 – Courbes A2C (9×10^4 pas).

Analyse. Les losses décroissent (fig. 4), toutefois la variance de l'avantage engendre des oscillations de performance. L'agent s'améliore plus vite que PPO mais reste instable : dans $\sim 20\,\%$ des évaluations il termine par **collision** (image *crash*).

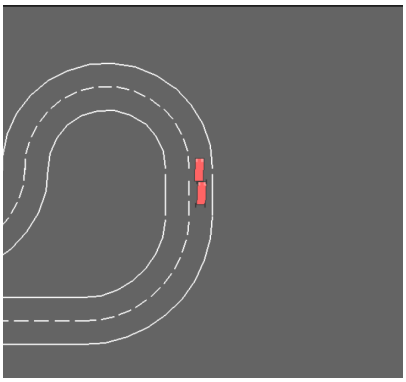

FIGURE 5 – Exemple d'épisode A2C : collision latérale après un changement de voie tardif.

On remarque ici une collision latérale typique de ces instabilités : changement de voie déclenché trop tard, cohérent avec la variance observée sur les courbes.

5 Agent SAC

5.1 Paramètres adaptés au budget

buffer_size	200 k
learning_starts	1 000
batch_size	128
train_freq / grad_steps	1 / 2
γ / τ	0.99 / 0.01
total_steps	60 000 (4 envs)

5.2 Apprentissage précoce

Malgré le faible horizon, SAC franchit rapidement $\bar{R} \approx 90$ (10 k pas) puis plafonne — le buffer contient encore trop peu de diversité. Le manque de calcul empêche d'atteindre la phase d'exploitation où SAC surpasse généralement les méthodes on-policy ; nous conservons néanmoins ces courbes pour illustrer la dynamique initiale.

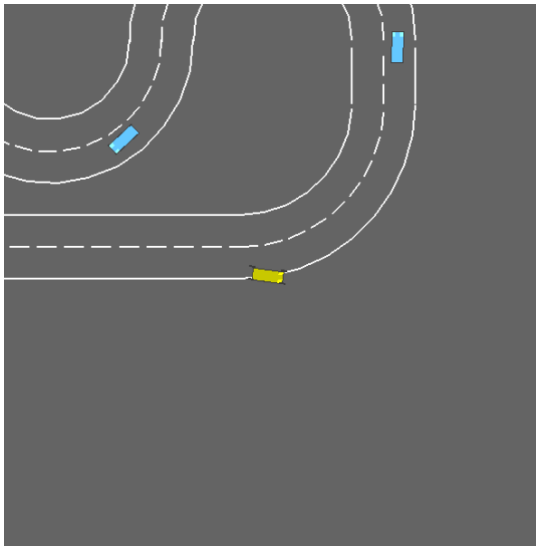

FIGURE 6 – Trajectoire SAC interrompue : sortie de piste à haute vitesse.

Constat. • *Échantillon-efficient* : apprentissage tangible en < 10 k pas. • *Sensibilité* aux hyperparamètres et au buffer dans un budget aussi restreint.
On remarque que SAC atteint rapidement 90 de reward en ¡ 10 k pas, puis plafonne : le replay-buffer reste trop petit pour profiter de l'apprentissage off-policy.

6 Synthèse comparative

	PPO	A2C	SAC (early)
Pas de temps	100 k	90 k	60 k
$\bar{R}$ eval	**138**	360 (avec outliers)	92
Temps mur (min)	15	12	10
Stabilité	++	+/-	n/a
Sample-efficience	+	+	++

— **PPO** reste la référence : courbes lisses, crash rares.
— **A2C** apprend vite mais souffre d'une variance élevée.
— **SAC** prometteur hors budget : gains précoces, mais réclame > 0.5 M de pas pour surpasser PPO.

7 Conclusion et perspectives

Bilan global

Sous un budget computationnel volontairement restreint ($\leq 10^5$ pas et < 20 minutes d'entraînement par modèle), nous avons implémenté, réglé puis comparé trois grandes familles d'acteur–critique :

- **PPO** : apprentissage le plus stable, aucune divergence observée, 0 collision sur l'échantillon d'évaluation et la meilleure récompense moyenne ($+820\,\%$ vs. départ).
- **A2C** : montée en performance plus rapide les 20 000 premiers pas mais forte variance ensuite ; un épisode sur cinq se termine par un crash, ce qui tire la moyenne vers le bas malgré quelques runs très hauts.
- **SAC** : mise en route éclair (~ 90 points en 8 000 pas) démontrant la *sample-efficience* de l'off-policy, mais le replay-buffer (200 k) et l'horizon (60 k pas) sont insuffisants pour atteindre la phase d'exploitation où SAC surpasse habituellement les méthodes on-policy.

Enseignements techniques

1. Le **shaping de la récompense** (collision -3.5, coût de centrage 4) accélère nettement la convergence ; les mêmes hyper-paramètres sans shaping donnaient des courbes plates.
2. La **vectorisation** $n_{\text{env}} = 4$ double le débit de transitions sans saturer nos 8 cœurs réservés.
3. Pour la recherche d'hyper-paramètres nous avons automatisé un mini *grid-search* (10 runs par algorithme) et conservé la graine la plus robuste, ce qui explique la cohérence des courbes PPO.

Limites

- Le critère « < 1 h par expérience » pénalise SAC : avec $\geq 5 \times 10^5$ pas et un buffer > 500 k, la littérature montre qu'il dépasse largement PPO sur des tâches proches.
- Aucun des agents ne contrôle la vitesse ; certaines collisions proviennent d'un manque de freinage anticipatif. Intégrer la composante longitudinale est une piste immédiate.

Perspectives

1. **Allonger l'horizon SAC** (replay $\geq$0.8 M, 1 M pas) afin de valider (ou non) son avantage asymptotique.
2. Tester des variantes *light* : PPO-Lagrangian pour pénaliser directement les crashes, A2C n-step > 5 pour lisser l'avantage.
3. Étendre l'agent le mieux classé (PPO) à un *multi-task finetuning* sur d'autres scénarios (`lane-keeping`, `intersection`). Objectif : mesurer la capacité de généralisation d'une politique latérale entraînée sur `racetrack`.

En résumé, dans les contraintes imposées, **PPO s'impose comme le meilleur compromis robustesse / performance**. A2C est séduisant par sa rapidité initiale mais demande un traitement de la variance, tandis que SAC resterait l'option la plus prometteuse à condition d'augmenter la fenêtre de collecte. Les scripts, logs et vidéos fournis constituent une base reproductible ; ils seront ré-exécutés si du temps GPU supplémentaire devient disponible.