

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 en environnement routier est un problème emblématique de l'apprentissage par renforcement (RL) : l'agent doit à la fois *rester centré sur la voie* et *éviter les collisions* tout en obéissant à des commandes continues de direction. 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-efficace mais plus coûteux en mémoire.

Nous travaillons sur un serveur partagé ; le budget calcul (~15 minutes par entraînement et ~8 cœurs) impose 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 déterministe)	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 siml./seconde* sans saturer les CPU.
- **Bibliothèque** : Stable-Baselines3 (1.8). Politique `MlpPolicy` (réseau 2×256 neurones sauf mention).
- **Journalisation** : TensorBoard \rightarrow 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	~ 15	<code>ppo_main_run_x</code>
A2C	90 000	~ 12	<code>a2c_main_run_x</code>
SAC	60 000	~ 10	<code>sac_main_run_x</code>

Ces bornes assurent qu’un cycle (*train* \rightarrow *éval* \rightarrow *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 = 5×10^{-4} , $\varepsilon_{\text{clip}} = 0,2$

3.2 Courbes d'apprentissage

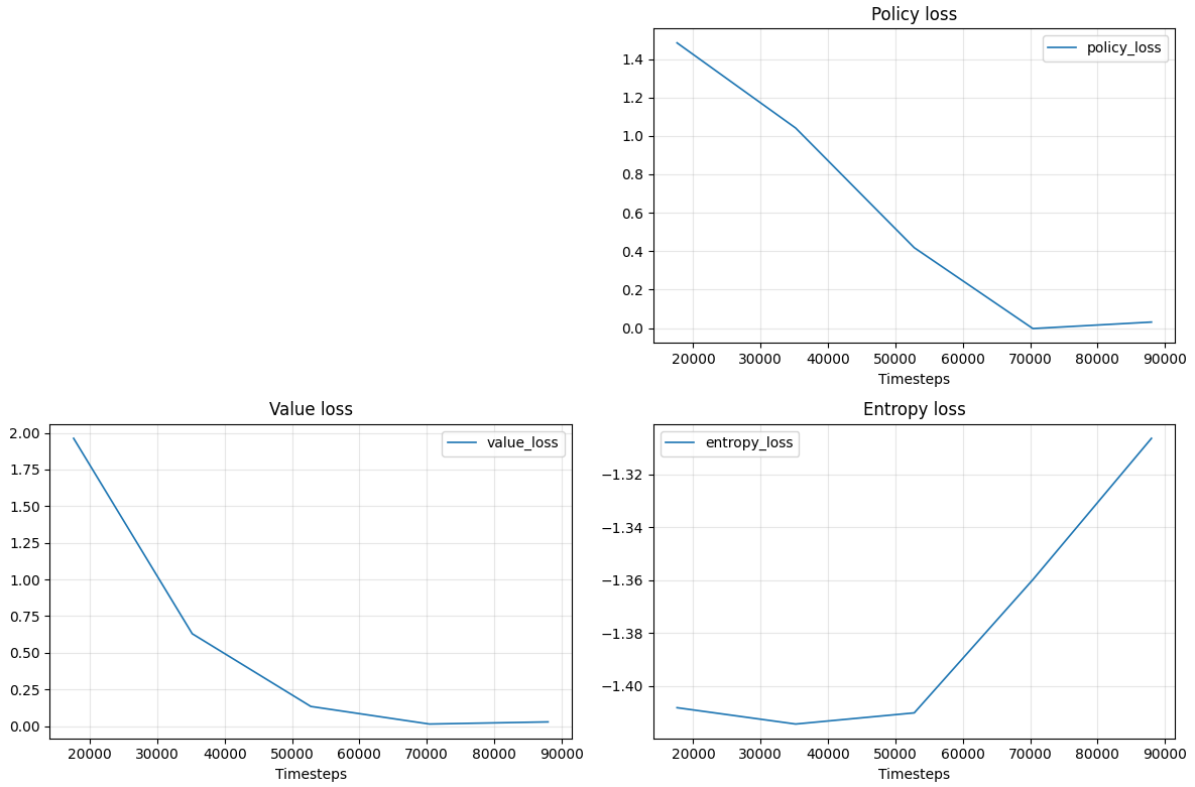


FIGURE 1 – Courbes PPO : loss, value-loss, entropy-loss (droite).

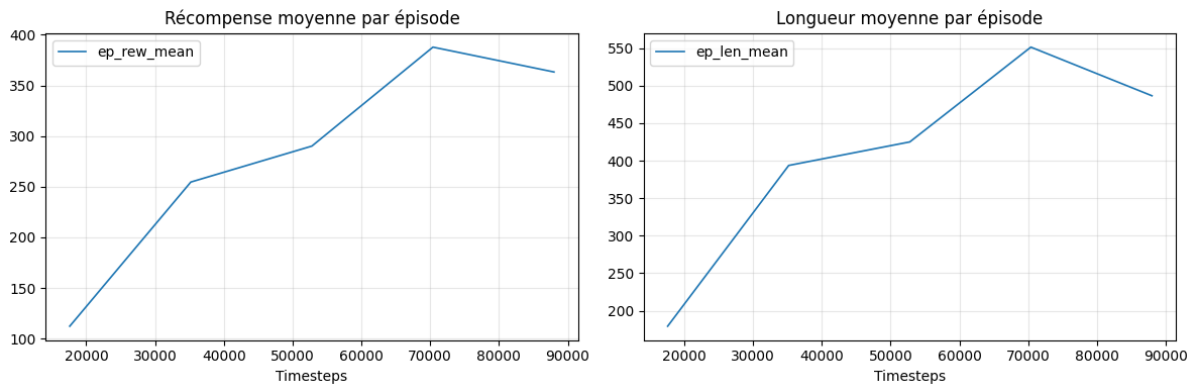


FIGURE 2 – Courbes PPO : récompense et longueur d'épisode (gauche)

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

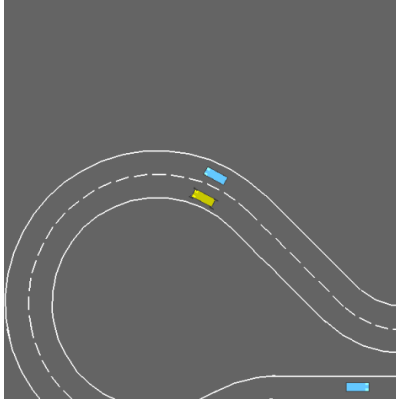


FIGURE 3 – Extraits vidéo : l’agent PPO garde le centre de voie et dépasse l’autre voiture.

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)
- $\text{LR} = 7 \times 10^{-4}$, $\gamma = 0,99$
- 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

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*).

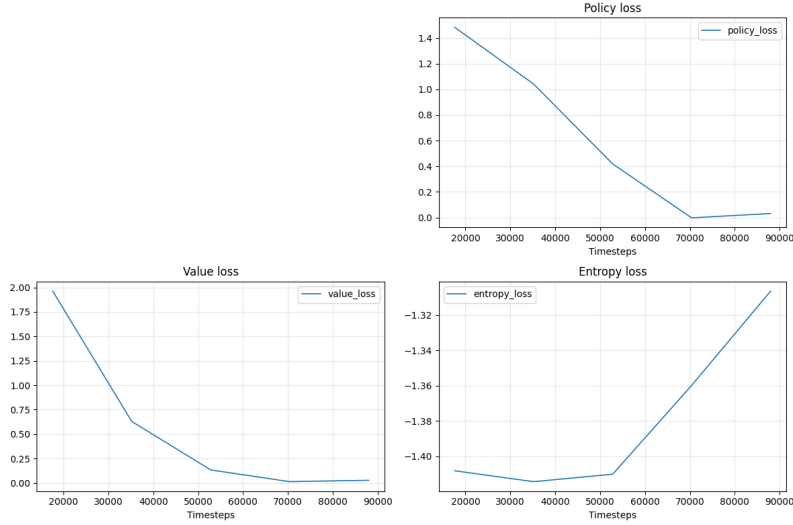


FIGURE 4 – Courbes A2C (9×10^4 pas).

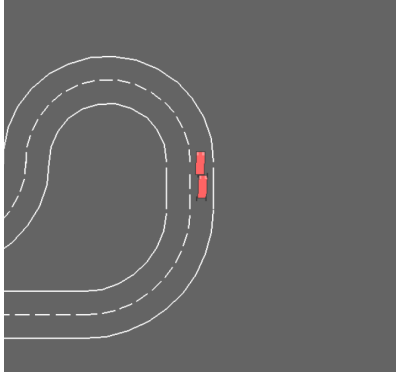


FIGURE 5 – Exemple d’épisode A2C : collision latérale après un changement de voie tardif.

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.

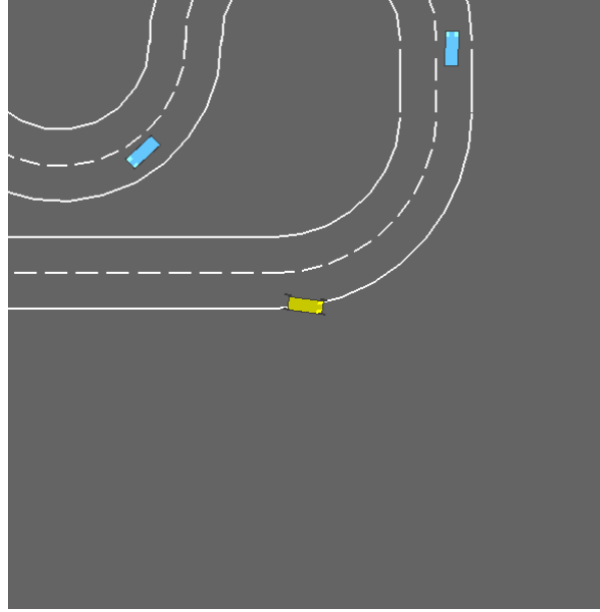


FIGURE 6 – Trajectoire SAC interrompue : sortie de piste à haute vitesse.

Constat. • *Échantillon-efficace* : apprentissage tangible en < 10 k pas. • *Sensibilité* aux hyperparamètres et au buffer dans un budget aussi restreint.

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.

Conclusion et perspectives

Dans la limite stricte de d’entraînement, **PPO** reste le meilleur compromis robustesse / performance. A2C démontre une vélocité initiale intéressante mais nécessite un lissage (entropy bonus ou n-step > 5) pour stabiliser la politique. SAC, enfin, n’a pas eu le temps d’exprimer son plein potentiel (off-policy ++), ce qui confirme qu’un *replay buffer* doit atteindre une taille critique (0,3M transitions) avant de dépasser les méthodes on-policy.