Apprentissage par Renforcement
Tennis Atari

Comparaison de DQN, PPO et A2C
Environnement ALE/Tennis-v5

Présentation de l'environnement

- Jeu Atari Tennis (ALE/Tennis-v5)
- Agent contrôle un joueur de tennis
- Objectif : marquer des points contre l'adversaire
- Environnement visuel complexe basé sur des images

Caractéristiques techniques

- Observations : images RGB
- Prétraitement :
  - Resize 84x84
  - Grayscale
  - Frame stacking (4 frames)
- Observation finale : (4, 84, 84)
- Récompenses : +1 point marqué / -1 point encaissé

Actions possibles de l'agent

- 18 actions discrètes
- NOOP, FIRE
- Déplacements : Haut, Bas, Gauche, Droite
- Combinaisons mouvement + FIRE

L'agent doit apprendre :
- Positionnement
- Anticipation de la balle
- Timing des frappes

Algorithme DQN – Pourquoi ?

- L’algorithme est conçu pour des environnements à actions discrètes et à observations visuelles.
- CNN pour apprendre directement à partir des pixels.
- Adapté aux environnements Atari.
- Les transitions sont stockées dans un replay buffer pour stabiliser l’apprentissage

DQN – Paramètres clés

- **Learning rate (1e-4) :** Paramètre critique pour la stabilité. Une valeur trop élevée provoque des divergences, tandis que 1e-4 est un compromis éprouvé sur Atari pour apprendre à partir de pixels sans instabilité majeure.

- **Replay buffer size (200 000) :** Détermine la diversité des expériences utilisées pour l'apprentissage. Cette taille est adaptée à un budget de 300k timesteps, permettant d'éviter à la fois le sur-apprentissage sur des transitions récentes

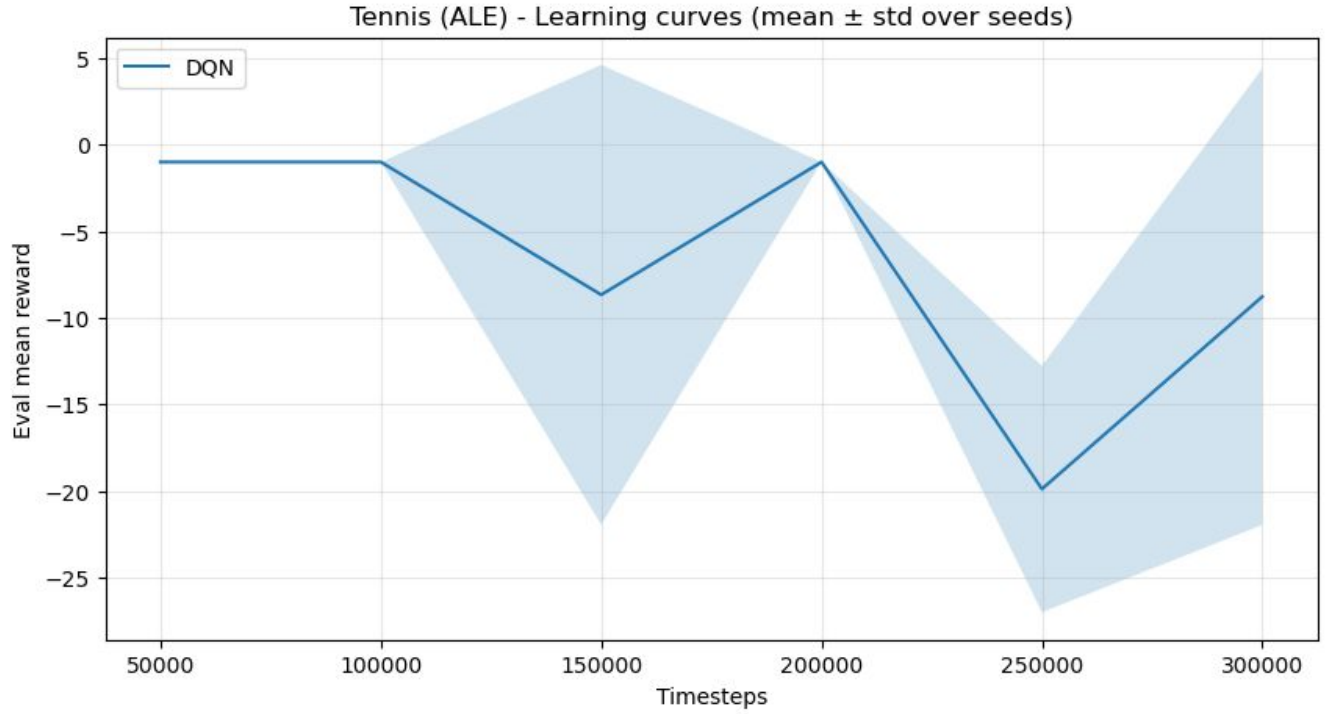
- **Exploration fraction (0.20) :** Paramètre central pour DQN. Une part significative du budget est dédiée à l'exploration afin de découvrir des stratégies pertinentes dans un environnement complexe et adversarial.

- **Device = cuda** : Exploitation du GPU pour accélérer l'entraînement du réseau convolutionnel, indispensable pour un apprentissage sur données visuelles.

Courbe d’apprentissage DQN

	algo	seed	final_mean	final_std	learn_time_s	eval_time_s	seed_time_s
0	DQN	0	-23.35	0.852936	2237.873586	41.750625	2285.037679
1	DQN	1	-4.00	5.805170	2443.208861	669.852178	3114.169228
2	DQN	2	-1.00	0.000000	2327.080521	610.380564	2938.416210

	algo	final_mean_avg	final_mean_std	runs	learn_time_min
0	DQN	-9.45	12.130849	3	38.934239



Algorithme PPO – Pourquoi ?

- Algorithme on-policy robuste
- Très stable grâce au clipping
- Bon compromis performance / simplicité
- Référence moderne en policy gradient

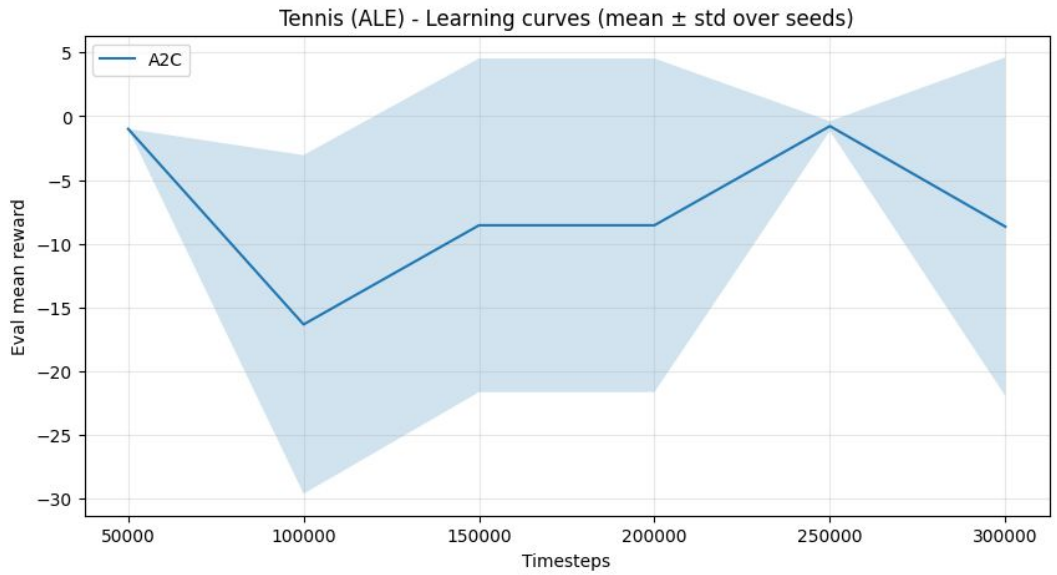
PPO – Paramètres clés

- Learning rate : 2.5e-4
- Rollout : 128 steps
- 4 epochs d'optimisation
- Clip range : 0.1 (stabilité)
- Entropy coef : 0.01 (exploration)

```
(  algo  seed  final_mean  final_std  learn_time_s  eval_time_s  seed_time_s
 0  PPO     0        -1.0        0.0    861.191553   519.817437  1402.684401
 1  PPO     1        -1.0        0.0    851.346390   538.259212  1411.092105
 2  PPO     2        -1.0        0.0    859.742607   529.549992  1406.403417,
   algo  final_mean_avg  final_mean_std  runs  learn_time_min
 0  PPO            -1.0             0.0     3       14.290447)
```

Algorithme A2C – Pourquoi ?

- Méthode actor-critic synchrone
- Apprentissage parallèle multi-environnements
- Convergence rapide
- Bon compromis simplicité / efficacité



A2C – Paramètres clés

- Learning rate : 7e-4
- n_steps : 8 (updates fréquentes)
- 4 environnements parallèles
- RMSprop (optimiseur stable)
- Advantage normalisé

Configuration expérimentale

- 300 000 timesteps d'entraînement
- 3 seeds différents
- Évaluations régulières
- Même preprocessing pour tous les algorithmes

Interprétation des résultats

- DQN : bonne efficacité en données mais apprentissage plus lent
- PPO : apprentissage stable et régulier
- A2C : convergence rapide grâce au parallélisme
- Variance réduite avec plusieurs seeds

Comparaison globale

- DQN : efficace mais instable au début
- PPO : meilleur compromis stabilité / performance
- A2C : rapide mais parfois moins stable

Choix de l'algorithme dépend du compromis : stabilité, vitesse et ressources

Conclusion

- Les trois algorithmes apprennent à jouer au Tennis Atari
- PPO se distingue par sa stabilité
- A2C profite du parallélisme
- DQN reste une référence historique

Projet complet de comparaison RL sur Atari