# Apprentissage par Renforcement Profond

Moez Krichen

# Apprentissage par Renforcement Profond

Moez Krichen

Laboratoire ReDCAD, Université de Sfax, Tunisie
moez.krichen@redcad.org

**Résumé.** L'apprentissage par renforcement profond (DRL) est une technique puissante pour apprendre des politiques pour des tâches de prise de décision complexes. Dans cet article, nous fournissons un aperçu du DRL, y compris ses composants de base, ses algorithmes et techniques clés, ainsi que ses applications dans des domaines tels que la robotique, les jeux et la conduite autonome. Nous discutons également des défis et des limitations du DRL, tels que l'inefficacité des échantillons et les préoccupations de sécurité, et identifions certaines des directions prometteuses pour la recherche future en DRL, telles que la méta-apprentissage, l'apprentissage par renforcement hiérarchique et la combinaison de DRL avec des techniques formelles. Dans la deuxième partie de l'article, nous discutons de plusieurs applications importantes du DRL, notamment l'apprentissage par transfert, l'apprentissage par renforcement multi-agent et l'apprentissage par renforcement explicatif. Nous explorons également la combinaison de DRL avec des techniques formelles, une zone de recherche prometteuse pour assurer la sécurité et la fiabilité des applications DRL. Enfin, nous identifions certaines des limitations et des problèmes ouverts en DRL, notamment l'efficacité des échantillons, les préoccupations de sécurité et de scalabilité. Pour aider les praticiens à appliquer efficacement le DRL dans leur travail, nous fournissons des recommandations pour commencer par des problèmes simples, choisir des algorithmes et des architectures appropriés, prêter attention à la sécurité et à l'éthique, collaborer avec des experts et rester à jour avec les dernières recherches dans le domaine. Nous concluons en soulignant l'impact potentiel du DRL dans un large éventail d'applications et en mettant l'accent sur la nécessité de prendre en compte les implications éthiques et sociétales du DRL de manière réfléchie.

## 1 Introduction

Le domaine de l'intelligence artificielle (IA) connu sous le nom d'apprentissage automatique (ML) se concentre sur la création de modèles et d'algorithmes qui peuvent apprendre à partir de données sans être explicitement programmés (5; 19). Les techniques de ML permettent aux ordinateurs de comprendre les motifs et les relations dans les données et de les utiliser pour faire des prédictions ou des jugements sur des

données préalablement inconnues. La reconnaissance d'images et de sons, le traitement du langage naturel (NLP), les systèmes de recommandation, la détection de fraudes et la maintenance prédictive sont autant d'exemples de l'utilisation de techniques de ML. Le type et la quantité de données étiquetées utilisées pour entraîner le modèle déterminent si l'algorithme de ML est supervisé, non supervisé ou d'apprentissage par renforcement.

Les réseaux de neurones artificiels (ANNs) avec plusieurs couches de nœuds interconnectés sont la base de l'apprentissage profond (DL), un sous-ensemble du ML motivé par la structure et le fonctionnement du cerveau humain (3; 1). Les modèles DL peuvent apprendre automatiquement des représentations de données complexes en extrayant progressivement des caractéristiques de niveau supérieur à partir de celles de niveau inférieur. Les algorithmes de DL ont démontré des performances de pointe dans diverses disciplines, notamment la vision par ordinateur, le NLP, la reconnaissance vocale et les jeux. Les modèles DL peuvent être entraînés avec des volumes massifs de données étiquetées et se mettre à l'échelle pour gérer des données extrêmement haute dimensionnalité et hétérogènes, telles que les photos, les vidéos et le texte. L'apprentissage non supervisé peut être réalisé en entraînant des modèles DL à reconstruire ou à produire des données, ou à regrouper et afficher des données dans des espaces de faible dimension.

La discipline de l'apprentissage par renforcement (RL) se concentre sur l'enseignement aux agents comment prendre une série de décisions dans un environnement pour maximiser un signal de récompense. Depuis sa première introduction dans les années 1980, le cadre RL a été utilisé pour résoudre une variété de problèmes, notamment les jeux, la robotique et les systèmes de contrôle (44). Les algorithmes RL traditionnels reposent souvent sur des caractéristiques artisanales et une approximation linéaire de fonction, ce qui limite leur capacité à gérer des tâches complexes avec des espaces d'état et d'action de grande dimension.

L'apprentissage profond par renforcement (DRL) est apparu comme une solution à ce problème en utilisant des réseaux de neurones profonds (DNNs) pour représenter la politique et la fonction de valeur de l'agent. Les DNN peuvent apprendre automatiquement des représentations hiérarchiques de données d'entrée complexes, telles que des images, des sons et du langage naturel, ce qui les rend bien adaptés pour gérer des espaces d'états et d'actions de grande dimension (41). La combinaison de RL et de DL a conduit à des avancées significatives dans le domaine, avec de nombreuses applications atteignant des performances de pointe.

Les dernières années ont vu une explosion d'intérêt pour le DRL, avec de nombreuses applications dans l'industrie et le monde universitaire. Par exemple, le DRL a été utilisé pour entraîner des robots à effectuer des tâches telles que la saisie d'objets et la navigation dans des environnements complexes (7), à jouer à des jeux tels que Go et les jeux Atari avec des performances surhumaines (40; 32), et à générer des descriptions de langage naturel d'images (46). Le succès du DRL dans ces domaines et d'autres encore a stimulé la recherche de nouveaux algorithmes, architectures et applications de DRL.

Dans ce travail, nous fournissons un aperçu du DRL, de ses applications, de ses limitations et des problèmes ouverts, en mettant l'accent sur certaines des zones de recherche les plus prometteuses dans ce domaine. Nous commençons dans la Section

2 en décrivant certains des algorithmes DRL les plus populaires, tels que les réseaux Q profonds, les techniques d'acteur-critique et les techniques de gradient de politique. Nous discutons également des avancées récentes en meta-learning, en RL hiérarchique et en apprentissage par imitation.

Dans la Section 3, nous discutons de certaines des applications les plus prometteuses de DRL dans divers domaines, notamment la robotique, les jeux, le NLP et les systèmes de recommandation. Nous explorons également la combinaison de DRL avec d'autres techniques, telles que l'apprentissage par transfert et l'apprentissage multi-tâches, pour améliorer l'efficacité et l'efficience du DRL (Section 4). Nous approfondissons également l'apprentissage RL multi-agent dans la Section 5 et l'apprentissage RL explicatif dans la Section 6.

Dans la Section 8, nous passons en revue les principales limitations et les problèmes ouverts dans le domaine du DRL, notamment l'inefficacité des échantillons, les préoccupations de sécurité et la mise à l'échelle. Nous discutons également de certaines des directions les plus prometteuses pour la recherche future en DRL, notamment la combinaison de DRL avec des techniques formelles (Section 7), l'apprentissage RL explicatif et l'apprentissage RL multi-agent.

Pour aider les praticiens à appliquer efficacement le DRL dans leur travail, nous fournissons des recommandations dans la Section 9, qui incluent des conseils sur la résolution de problèmes simples, le choix d'algorithmes et d'architectures appropriés, l'attention portée à la sécurité et à l'éthique, et la collaboration avec des experts.

Enfin, dans la Section 10, nous résumons l'article et présentons quelques extensions possibles pour la recherche future dans ce domaine passionnant et en constante évolution. Nous soulignons l'impact potentiel du DRL dans un large éventail d'applications, tout en mettant en évidence la nécessité de considérer attentivement les implications éthiques et sociétales des applications de DRL. Dans l'ensemble, nous pensons que le DRL a le potentiel de révolutionner de nombreux secteurs et domaines, et nous sommes impatients de voir les progrès et le développement continu de ce domaine passionnant.

# 2 Algorithmes DRL

Les algorithmes DRL combinent les principes de RL avec les réseaux de neurones profonds pour apprendre des politiques et des fonctions de valeur complexes à partir de données d'entrée de haute dimension. Dans cette section, nous donnons une introduction à certains des algorithmes DRL les plus largement utilisés, tels que les techniques d'acteur-critique, les techniques PG et les réseaux de Q-profond (DQN).

## 2.1 DQN

DQN [1] est un type d'algorithme RL basé sur la valeur qui utilise un réseau de neurones profonds pour approximer la fonction de valeur d'action optimale, $Q^*(st, act)$, qui représente la récompense cumulative attendue obtenue en effectuant l'action *act* dans l'état *st* et en suivant la politique optimale par la suite. Pour augmenter la stabilité et la convergence, la méthode DQN utilise un réseau cible et une mémoire de lecture.

1. DQN=Réseaux Q-Profonds

La mémoire de lecture stocke les transitions $(st_t, act_t, r_t, st_{t+1})$ dans un tampon de lecture, à partir duquel des lots de transitions sont échantillonnés uniformément au hasard pour mettre à jour le réseau Q. Afin de fournir des cibles plus stables pour les mises à jour de Q-learning, le réseau cible est une copie distincte du réseau Q qui est périodiquement mis à jour avec les poids du réseau Q. DQN a été appliqué à une large gamme de tâches, notamment la lecture de jeux Atari (32) et le contrôle de systèmes robotiques (24; 45).

## 2.2 Techniques AC

Les techniques AC [2] (11) sont un type d'algorithme RL basé sur la politique qui utilise deux réseaux de neurones : un réseau d'acteur qui apprend une politique stochastique $\pi(act|st)$, et un réseau de critique qui mesure la fonction de valeur $V(st)$. Alors que le réseau de critique utilise l'apprentissage de différence temporelle (TD) pour estimer la fonction de valeur, le réseau d'acteur utilise le théorème PG pour mettre à jour les paramètres de la politique dans la direction de la récompense anticipée. Trois sous-catégories de méthodes acteur-critique existent : acteur seul, acteur seul sur la politique et acteur seul hors politique. Les techniques sur la politique, telles que A2C et A3C (31), mettent à jour la politique en fonction de la politique actuelle, tandis que les techniques hors politique, telles que DDPG (24) et TD3 (6), utilisent une politique comportementale distincte pour générer les données utilisées pour mettre à jour la politique cible. Les techniques d'acteur seul, telles que PPO, mettent à jour la politique sans estimer explicitement la fonction de valeur.

## 2.3 Techniques PG

Les techniques PG [3] sont un groupe d'algorithmes RL qui optimisent directement les paramètres de la politique pour maximiser la récompense attendue. Le théorème PG fournit un moyen de calculer le gradient de la récompense attendue par rapport aux paramètres de la politique, qui peut être utilisé pour mettre à jour la politique en utilisant la descente de gradient stochastique (SGD) [4]. Les techniques PG peuvent être subdivisées en deux sous-catégories : les techniques VPG [5] et les techniques TRPO [6]. Les techniques VPG mettent à jour la politique en utilisant le gradient de première ordre de la récompense attendue, tandis que TRPO (39) utilise une contrainte de région de confiance pour garantir que la mise à jour de la politique ne s'écarte pas trop de la politique actuelle, ce qui améliore l'efficacité del'échantillonnage et la convergence.

## 2.4 Techniques PPO

PPO [7] est un algorithme RL basé sur la politique qui a été développé en 2017. PPO est une variante de la technique PG traditionnelle qui utilise un objectif de substitution

2. Techniques AC=Techniques d'acteur-critique
3. Techniques PG=Techniques de Gradient de Politique
4. SGD=Descente de Gradient Stochastique
5. VPG=Techniques de Gradient de Politique Vanille
6. TRPO=Optimisation de Politique à Région de Confiance
7. PPO=Optimisation de Politique Proximale

tronqué pour mettre à jour le réseau de politique. L'objectif de substitution tronqué limite la taille de la mise à jour de la politique pour chaque itération, ce qui contribue à éviter de grands changements de politique qui pourraient déstabiliser le processus d'apprentissage. L'un des avantages clés de PPO est son efficacité d'échantillonnage. PPO est capable d'obtenir de bonnes performances avec moins d'échantillons que les anciens algorithmes PG tels que l'optimisation de politique à région de confiance (TRPO). PPO est un algorithme basé sur la politique qui ne nécessite pas de mémoire de lecture distincte car il apprend directement à partir de la politique existante.

## 2.5 Techniques SAC

SAC [8] est un autre algorithme RL basé sur la politique qui a été proposé par Haarnoja et al. en 2018 (8). SAC est un algorithme hors politique qui utilise un objectif d'entropie maximale pour encourager l'exploration et un réseau d'acteur stochastique pour échantillonner des actions. L'objectif d'entropie maximale encourage la politique à prendre des actions qui ne sont pas seulement optimales mais aussi diverses, ce qui peut aider à améliorer l'exploration et à conduire à de meilleures performances à long terme. SAC utilise également un réseau de critique pour estimer la fonction de valeur état-action. Contrairement aux techniques de Q-learning traditionnelles, qui utilisent une seule estimation de la fonction Q, SAC utilise un ensemble de fonctions Q pour estimer la fonction de valeur. L'utilisation de plusieurs fonctions Q contribue à réduire le biais de surestimation et à améliorer la stabilité du processus d'apprentissage.

## 2.6 Techniques TD3

TD3 [9] est un algorithme RL basé sur la politique qui a été proposé par Fujimoto et al. en 2018 (6). TD3 est un algorithme hors politique qui utilise deux réseaux de critique pour estimer la fonction de valeur état-action et un mécanisme de mise à jour différée pour stabiliser l'apprentissage. Les deux réseaux de critique sont entraînés indépendamment, ce qui contribue à réduire le biais de surestimation et à améliorer la précision des estimations de la valeur. Pour augmenter la stabilité du processus d'apprentissage, TD3 applique également une technique de lissage de politique cible. En ajoutant du bruit à la politique cible pendant l'apprentissage, la technique de lissage de politique cible atténue la sensibilité du processus d'apprentissage aux changements minimes de la politique.

Dans l'ensemble, ces avancées récentes dans le domaine de la DRL ont montré une grande promesse en améliorant les performances et l'efficacité des agents RL. La tâche particulière à accomplir, ses besoins, les ressources de traitement disponibles et la méthode choisie déterminent finalement la décision finale. À mesure que le domaine continue d'évoluer, il est probable que de nouveaux algorithmes et techniques continueront à émerger pour faire avancer l'état de l'art en DRL.

8. SAC=Acteur-Critique Souple
9. TD3=Acteur-Critique Profond à Retardement Double

# 3 Applications de DRL

La DRL a démontré un excellent potentiel dans une gamme d'applications, allant de la robotique et des jeux à la NLP et au-delà. Dans cette section, nous décrirons certaines des applications les plus prometteuses de la DRL, ainsi que les défis qui se posent lors de l'application de ces techniques à des problèmes réels.

## 3.1 Robotique

La robotique est l'un des domaines les plus passionnants dans lesquels la DRL peut être utilisée. Les agents RL peuvent être utilisés pour contrôler des robots qui effectuent des tâches complexes dans des environnements réels. Par exemple, la RL a été utilisée pour entraîner des robots à effectuer des tâches telles que la saisie d'objets, la manipulation d'outils et la navigation dans des environnements complexes (22; 7).

Cependant, l'application de la RL à la robotique pose plusieurs défis. Un défi majeur est le besoin d'efficacité d'échantillonnage, car les interactions réelles avec un robot peuvent être lentes et coûteuses. Pour relever ce défi, les chercheurs ont développé des techniques telles que l'apprentissage par transfert, où un agent pré-entraîné est affiné pour une nouvelle tâche, et l'apprentissage basé sur la simulation, où l'agent apprend à partir d'environnements simulés avant d'être déployé dans le monde réel (9).

## 3.2 Jeux

La DRL a également été appliquée avec succès aux jeux. Les agents RL se sont distingués dans des jeux tels que Go, Chess et autres, en particulier. Ces jeux posent un défi unique pour les agents RL, car ils impliquent une planification à long terme et une réflexion stratégique (40). L'un des principaux avantages de l'utilisation de la RL pour les jeux est qu'elle peut apprendre à partir de l'auto-apprentissage, où l'agent joue contre lui-même et s'améliore au fil du temps. Cela permet à l'agent d'apprendre à partir d'une grande quantité d'expérience sans avoir besoin de supervision humaine (42).

## 3.3 NLP

La DRL a également montré des promesses en NLP. Les agents RL peuvent être utilisés pour apprendre à générer du texte, répondre à des questions et effectuer d'autres tâches de NLP. Par exemple, la RL a été utilisée pour former des chatbots capables d'interagir avec les utilisateurs de manière naturelle et engageante (23). Cependant, l'application de la RL à la NLP pose également plusieurs défis. Un défi est le besoin d'interprétabilité, car il peut être difficile de comprendre comment l'agent prend des décisions. Un autre défi est le besoin d'efficacité d'échantillonnage, car la génération de texte peut être un processus lent et coûteux. Pour relever ces défis, les chercheurs ont développé des techniques telles que le façonnage de récompenses, où la fonction de récompense est conçue pour encourager un comportement souhaitable, et l'apprentissage de programme, où l'agent est formé sur des tâches progressivement plus difficiles (47; 33).

### 3.4 Test de boîte noire d'applications Android

L'article (10) étudie si la RL peut remplacer les algorithmes métaheuristiques conçus par l'homme dans les tests logiciels basés sur la recherche (SBST) et propose un nouveau cadre appelé GunPowder. Les auteurs reformulent le SUT [10] en tant qu'environnement RL et entraînent un agent DDQN [11] avec DNN pour produire automatiquement des données de test qui optimisent les critères de test structurels. Les auteurs effectuent une recherche empirique modeste pour tester leur approche et constatent que l'agent peut apprendre des algorithmes métaheuristiques SBST et atteindre unecouverture de branche de 100

### 3.5 Tests logiciels basés sur la recherche

Le travail (38) a développé et évalué ARES, une technique de Deep Reinforcement Learning (RL) pour tester des applications Android en boîte noire. Les auteurs utilisent le Deep RL et les réseaux neuronaux pour explorer l'espace d'état important des applications Android lors des tests. ARES surpasse Time Machine et Q-Testing en termes de couverture et de découverte de failles. Les activités enchaînées et bloquées dans les applications Android rendent le Deep RL particulièrement performant, selon les auteurs. Pour éviter le coût de calcul de l'optimisation des applications réelles, les auteurs ont développé FATE, un outil pour affiner les hyperparamètres de l'algorithme Deep RL sur des applications simulées. FATE peut réduire considérablement le temps et le coût de l'ajustement des hyperparamètres, selon les études des auteurs. Le Deep RL et les réseaux neuronaux peuvent automatiser l'exploration de l'espace d'état énorme, améliorant ainsi les tests en boîte noire des applications Android. Les auteurs démontrent que le Deep RL peut évaluer les applications Android dans des environnements d'exploration complexes. Le travail pourrait aider la communauté de l'ingénierie logicielle à améliorer les tests d'applications Android.

## 4 Le transfert d'apprentissage en RL

Le transfert d'apprentissage est une technique d'apprentissage automatique dans laquelle la connaissance acquise pour une tâche est transférée à une tâche connexe. En RL, le transfert d'apprentissage peut être utilisé pour accélérer l'apprentissage d'une nouvelle tâche en exploitant la connaissance acquise à partir d'une tâche connexe.

L'utilisation de réseaux neuronaux pré-entrainés comme point de départ pour l'apprentissage d'une nouvelle tâche est l'une des méthodes les plus prometteuses pour le transfert d'apprentissage en RL. Le réseau pré-entrainé peut être affiné pour la nouvelle tâche, permettant à l'agent d'apprendre plus rapidement et avec moins d'exemples d'entraînement. Par exemple, un réseau pré-entrainé qui a appris à jouer à un jeu peut être affiné pour jouer à un jeu connexe, permettant à l'agent d'apprendre plus rapidement qu'en partant de zéro.

10. SUT=Logiciel sous test
11. DDQN=Réseaux Q profonds doubles

Une autre approche pour le transfert d'apprentissage en RL consiste à utiliser la connaissance acquise à partir d'un ensemble de tâches connexes pour apprendre une nouvelle tâche. Cette approche est connue sous le nom d'apprentissage multi-tâches et implique la formation de l'agent sur plusieurs tâches connexes simultanément. L'idée est que l'agent apprendra à partager des connaissances entre les tâches, ce qui lui permettra d'apprendre plus rapidement et avec moins d'exemples d'entraînement pour chaque tâche individuelle.

Le transfert d'apprentissage a été appliqué avec succès à une variété de problèmes RL. Par exemple, le transfert d'apprentissage a été utilisé pour apprendre à des agents à jouer à plusieurs jeux Atari en utilisant un seul réseau neuronal (30). Le transfert d'apprentissage a également été utilisé pour apprendre à des agents à jouer à Pong en utilisant la connaissance acquise à partir de la jouerie de Breakout (34).

Cependant, le transfert d'apprentissage en RL pose également plusieurs défis. Un défi majeur est la nécessité de trouver une tâche connexe qui peut fournir une connaissance utile pour la nouvelle tâche. Un autre défi est la nécessité d'équilibrer la quantité de connaissance transférée de la tâche connexe avec la quantité d'apprentissage spécifique à la nouvelle tâche.

À mesure que le domaine du RL continue d'évoluer, il est probable que de nouvelles techniques et approches de transfert d'apprentissage émergeront pour faire avancer l'état de l'art.

# 5 RL multi-agent

Le RL multi-agent (RLMA) est un sous-domaine du RL qui traite des agents qui interagissent entre eux dans un environnement partagé. En RLMA, les agents doivent apprendre à coopérer ou à rivaliser avec d'autres agents pour atteindre leurs objectifs. Le RLMA devient de plus en plus important en tant qu'outil pour résoudre des problèmes complexes du monde réel, tels que la gestion du trafic, l'optimisation de la chaîne d'approvisionnement et la réponse aux catastrophes. Dans ces scénarios, plusieurs agents doivent coordonner leurs actions pour atteindre un objectif commun.

L'un des principaux défis du RLMA est le problème de coordination : comment s'assurer que les agents apprennent à coordonner efficacement leurs actions. Il existe plusieurs approches pour résoudre ce problème, notamment l'entraînement centralisé avec exécution décentralisée (ECED), l'entraînement décentralisé avec exécution centralisée (EDCE) et l'entraînement et l'exécution entièrement décentralisés.

Dans l'ECED, un agent centralisé est entraîné avec accès aux états et actions de tous les agents. Pendant l'exécution, chaque agent exécute son action en fonction des recommandations de l'agent centralisé. Dans l'EDCE, chaque agent est entraîné indépendamment, mais pendant l'exécution, un agent centralisé coordonne leurs actions. L'entraînement et l'exécution entièrement décentralisés impliquent que chaque agent apprend et agit de manière indépendante.

Le RLMA a été utilisé pour résoudre une variété de problèmes, tels que le football robotique (43), la gestion du trafic (36) et l'optimisation de la chaîne d'approvisionnement (35). Ces dernières années, le RL profond a été utilisé pour entraîner des agents en RLMA, ce qui a conduit à des améliorations significatives des performances.

# 6 RL Explicable

Une branche du RL appelée RL Explicable (RLX) est préoccupée par la création d'algorithmes qui peuvent fournir des explications pour leurs décisions et actions. Le but de RLX est de rendre le RL plus transparent et compréhensible, permettant aux humains de mieux comprendre et faire confiance aux décisions prises par les agents RL.

RLX devient de plus en plus important car le RL est de plus en plus utilisé dans des applications du monde réel, telles que les voitures autonomes et les soins de santé. Dans ces applications, il est crucial de pouvoir comprendre comment l'agent RL prend des décisions, surtout lorsque ces décisions ont des conséquences importantes pour la sécurité et le bien-être humains.

Les compromis entre performance et explicabilité sont l'un des principaux problèmes avec RLX. Les algorithmes RL hautement explicables peuvent sacrifier leur performance pour fournir des explications, tandis que les algorithmes hautement performants peuvent être difficiles à expliquer.

Il existe plusieurs approches à RLX, notamment les techniques basées sur des règles, les techniques basées sur des modèles et les techniques agnostiques aux modèles. Les techniques basées sur des règles utilisent des règles explicites pour guider le processus de prise de décision et fournir des explications pour les actions prises. Les techniques basées sur des modèles utilisent des modèles conçus pour être interprétables, tels que les arbres de décision ou les modèles linéaires. Les techniques agnostiques aux modèles se concentrent sur la génération d'explications pour les modèles boîte noire, tels que les réseaux de neurones profonds, en analysant les entrées et sorties du modèle.

De nombreux problèmes ont été abordés avec RLX, notamment les soins de santé (37), la finance (4) et la robotique (2). Dans les soins de santé, RLX a été utilisé pour prédire les résultats des patients et fournir des explications pour les prédictions, permettant aux médecins de mieux comprendre et faire confiance aux prédictions de l'algorithme. En finance, RLX a été utilisé pour prendre des décisions d'investissement et fournir des explications pour les décisions, permettant aux investisseurs de mieux comprendre et faire confiance aux recommandations de l'algorithme. En robotique, RLX a été utilisé pour permettre aux humains de mieux comprendre et interagir avec les robots, les rendant plus utiles et efficaces dans une variété d'applications.

À mesure que le domaine de RLX continue d'évoluer, il est probable que de nouvelles techniques et approches émergeront pour faire progresser l'état de l'art.

# 7 Combinaison de DRL et de techniques formelles

Les techniques formelles sont des techniques mathématiques pour spécifier, analyser et vérifier les systèmes logiciels et matériels (16; 13; 20; 28; 14). Elles sont souvent utilisées pour garantir la correction et la sécurité des systèmes critiques, tels que ceux utilisés dans l'aérospatiale, les dispositifs médicaux et les véhicules autonomes (27; 25). Les techniques formelles reposent généralement sur la logique et le raisonnement, et elles peuvent être utilisées pour prouver la correction d'un système par rapport à une spécification donnée (21; 15).

Les algorithmes DRL peuvent parfois apprendre des politiques qui violent les contraintes de sécurité ou ne satisfont pas certaines exigences. Cela est particulièrement préoccupant dans les applications critiques pour la sécurité, où les conséquences d'une défaillance peuvent être catastrophiques. Pour résoudre ce problème, il y a un intérêt croissant pour la combinaison de DRL et de techniques formelles (17; 29). L'idée est d'utiliser des techniques formelles pour spécifier les contraintes de sécurité et les exigences, puis d'utiliser DRL pour apprendre des politiques qui satisfont ces contraintes et exigences (12; 26).

Une approche pour combiner le DRL et les techniques formelles est d'utiliser la vérification de modèle, qui est une technique de vérification formelle qui vérifie si un système donné satisfait une spécification donnée (18; 19). La vérification de modèle peut être utilisée pour vérifier qu'une politique apprise satisfait les contraintes de sécurité ou d'autres exigences. Une autre approche consiste à utiliser des spécifications formelles pour guider le processus d'apprentissage en DRL. Par exemple, une spécification formelle peut être utilisée comme fonction de récompense dans l'algorithme RL, encourageant l'agent à apprendre une politique qui satisfait les exigences spécifiées dans la spécification.

La combinaison de DRL et de techniques formelles est encore un domaine de recherche relativement nouveau, et il existe de nombreuses questions et défis ouverts. Un défi consiste à développer des techniques qui peuvent être mises à l'échelle pour les systèmes larges et complexes. Un autre défi consiste à s'assurer que les politiques apprises sont non seulement sûres, mais aussi optimales et efficaces.

## 8 Limitations et enjeux ouverts

Malgré les énormes avancées réalisées en DRL ces dernières années, certains enjeux demeurent et certaines limitations n'ont pas encore été résolues. L'inefficacité de l'échantillonnage est l'un des principaux inconvénients de la DRL. Les algorithmes de DRL efficaces nécessitent souvent beaucoup de données pour apprendre, ce qui peut être prohibitif dans les applications pratiques. Il est donc nécessaire de développer des algorithmes de DRL capables d'apprendre à partir de moins d'échantillons, par exemple des algorithmes de méta-apprentissage qui peuvent apprendre à apprendre à partir d'expériences passées.

Une autre limitation de la DRL est liée aux préoccupations de sécurité. Les agents de RL peuvent parfois apprendre à exploiter l'environnement de manière inattendue et potentiellement dangereuse, ce qui suscite des préoccupations en matière de sécurité. Il est donc nécessaire de développer des algorithmes de DRL capables de garantir un comportement sûr et fiable, par exemple des algorithmes qui intègrent des contraintes de sécurité dans le processus d'apprentissage.

La scalabilité est également un problème majeur en DRL. Les algorithmes de DRL traditionnels peuvent devenir excessivement lents ou exiger des ressources mémoire considérables à mesure que le nombre d'états et d'actions dans l'environnement augmente. Il est donc nécessaire de développer des algorithmes de DRL capables de s'adapter à des environnements plus vastes et complexes, par exemple des algorithmes de RL hiérarchique capables d'apprendre à plusieurs niveaux d'abstraction.

Tab. 1 – *Limitations de la DRL*

| **Limitation** | **Description** |
|---|---|
| Inefficacité de l'échantillonnage | Les algorithmes de DRL nécessitent souvent de grandes quantités de données pour apprendre des politiques efficaces, ce qui peut être prohibitif dans les applications du monde réel. |
| Préoccupations de sécurité | Les agents de RL peuvent parfois apprendre à exploiter l'environnement de manière inattendue et potentiellement dangereuse, ce qui suscite des préoccupations en matière de sécurité. |
| Scalabilité | À mesure que le nombre d'états et d'actions dans l'environnement augmente, les algorithmes de DRL traditionnels peuvent devenir excessivement lents ou exiger des ressources mémoire considérables. |
| Interprétabilité et explicabilité | À mesure que les agents de RL deviennent plus complexes et plus puissants, il devient de plus en plus difficile de comprendre et d'interpréter leur comportement. |

Enfin, l'interprétabilité et l'explicabilité sont des enjeux importants en DRL. À mesure que les agents de RL deviennent plus complexes et plus puissants, il devient de plus en plus difficile de comprendre et d'interpréter leur comportement. Il est donc nécessaire de développer des algorithmes de DRL capables de fournir des explications pour leurs décisions et actions, permettant ainsi aux humains de mieux comprendre et faire confiance au comportement des agents de RL. Un résumé de ces limitations est présenté dans le Tableau 1.

Malgré ces limitations et enjeux ouverts, il existe plusieurs directions prometteuses pour la recherche future en DRL. Le méta-apprentissage, qui consiste à apprendre à apprendre à partir d'expériences passées, est une approche prometteuse pour résoudre le problème d'inefficacité de l'échantillonnage en DRL. Le RL hiérarchique, qui implique l'apprentissage à plusieurs niveaux d'abstraction, est une approche prometteuse pour résoudre le problème de scalabilité en DRL. L'apprentissage curriculaire, qui consiste à augmenter progressivement la difficulté de la tâche d'apprentissage, est une approche prometteuse pour accélérer l'apprentissage dans la DRL.

En ce qui concerne les préoccupations de sécurité, des algorithmes de DRL ont été proposés pour garantir un comportement sûr et fiable. Par exemple, les algorithmes de RL contraints permettent de spécifier des contraintes de sécurité qui doivent être respectées par l'agent de RL. De même, les algorithmes de RL inverse permettent de déduire la politique à partir des comportements observés, ce qui peut être utilisé pour détecter les comportements indésirables.

Enfin, en ce qui concerne l'interprétabilité et l'explicabilité, des travaux récents ont développé des approches pour expliquer le comportement des agents de RL, notamment en utilisant des méthodes de visualisation et d'interprétation. Par exemple, les méthodes de saliency map peuvent être utilisées pour mettre en évidence les parties de

l'entrée qui ont le plus d'influence sur la sortie de l'agent de RL.

En somme, malgré les enjeux et les défis à relever, la DRL continue d'offrir de nombreuses opportunités pour résoudre des problèmes complexes dans divers domaines, notamment en robotique, en jeux vidéo, en finance, et dans de nombreux autres domaines.

# 9 Recommandations pour les praticiens

Il est crucial pour les praticiens de se tenir au courant des avancées les plus récentes et des meilleures pratiques de l'industrie alors que la DRL continue de s'étendre et de trouver de nouvelles applications. Voici quelques suggestions pour les professionnels qui souhaitent utiliser la DRL dans leur travail.

## 9.1 Commencer avec des problèmes simples

La DRL peut être un outil puissant pour résoudre des problèmes complexes de prise de décision, mais elle peut aussi être difficile à appliquer efficacement. Les praticiens qui découvrent la DRL peuvent trouver utile de commencer par des problèmes simples et de progressivement travailler sur des problèmes plus complexes. Des problèmes simples peuvent comprendre des tâches telles que contrôler un bras robotique simple, jouer à des jeux simples, ou naviguer dans un labyrinthe simple. En commençant par des problèmes simples, les praticiens peuvent acquérir de l'expérience avec la DRL et développer une intuition pour son fonctionnement et comment l'appliquer efficacement dans différents contextes. Une fois que les praticiens sont à l'aise avec des problèmes simples, ils peuvent progressivement travailler sur des problèmes plus complexes, tels que la conduite autonome, le diagnostic médical ou la manipulation de robots.

## 9.2 Choisir des algorithmes et architectures appropriés

Il existe de nombreux algorithmes et architectures DRL différents, chacun ayant ses propres forces et faiblesses. Les praticiens doivent tenir compte des exigences de leur problème et choisir des algorithmes et architectures appropriés pour leurs besoins spécifiques. Par exemple, si le problème implique des espaces d'action continus, des algorithmes tels que DDPG ou TD3 peuvent être plus appropriés que des algorithmes tels que Q-learning ou SARSA, conçus pour des espaces d'action discrets. De même, si le problème implique des espaces d'entrée de haute dimension, des architectures telles que les CNNs [12] ou les RNNs [13] peuvent être plus appropriées que des réseaux de neurones feedforward simples. En choisissant soigneusement les algorithmes et architectures, les praticiens peuvent s'assurer qu'ils utilisent les techniques les plus appropriées pour leur problème spécifique.

12. CNN = Réseau de neurones convolutifs
13. RNN = Réseau de neurones récurrents

### 9.3 Prêter attention à la sécurité et à l'éthique

Alors que la DRL est de plus en plus utilisée dans des applications critiques pour la sécurité, telles que la conduite autonome et le diagnostic médical, il est important que les praticiens prêtent une attention particulière à la sécurité et aux considérations éthiques. Les praticiens doivent s'assurer que les politiques apprises sont sûres, transparentes et interprétables, et qu'elles ne perpétuent pas de biais ou de discrimination. Cela peut impliquer l'utilisation de techniques telles que l'entraînement adversarial pour s'assurer que les politiques apprises sont robustes aux attaques adverses, ou l'utilisation de techniques telles que la pensée contrefactuelle pour s'assurer que les politiques apprises ne perpétuent pas de biais ou de discrimination. De plus, les praticiens doivent être conscients des implications éthiques de leur travail et doivent considérer l'impact potentiel de leurs applications sur la société et l'environnement.

### 9.4 Collaborer avec des experts

La DRL est un domaine hautement interdisciplinaire qui nécessite une expertise en informatique, en mathématiques et souvent une connaissance spécifique du domaine. Les praticiens qui découvrent la DRL peuvent trouver utile de collaborer avec des experts dans ces domaines pour s'assurer qu'ils utilisent des techniques appropriées et qu'ils abordent les défis pertinents. Par exemple, les praticiens travaillant sur des applications de diagnostic médical peuvent collaborer avec des experts médicaux pour s'assurer que leurs modèles sont cliniquement pertinents et précis. De même, les praticiens travaillant sur des applications de conduite autonome peuvent collaborer avec des experts en transport pour s'assurer que leurs modèles sont sûrs et efficaces dans des scénarios de conduite réels. En collaborant avec des experts, les praticiens peuvent s'assurer qu'ils utilisent les techniques les plus appropriées et abordent les défispertinents dans leur domaine spécifique.

### 9.5 Se tenir au courant des dernières avancées

La DRL est un domaine en constante évolution avec de nouveaux développements et techniques qui émergent régulièrement. Les praticiens doivent faire un effort pour se tenir au courant des dernières avancées et participer à des conférences et des ateliers pour apprendre de nouvelles techniques et les meilleures pratiques. Cela peut impliquer la lecture d'articles de recherche, l'assistance à des présentations et des séminaires, ou la participation à des forums et des groupes de discussion en ligne. En restant à jour avec les dernières recherches, les praticiens peuvent s'assurer qu'ils utilisent les techniques les plus à la pointe et contribuent au développement continu du domaine.

## 10 Conclusion

Nous avons également discuté de certaines des applications clés de la DRL, notamment l'apprentissage par transfert, le RL multi-agent et le RL explicatif, et nous avons souligné certains des défis et des limites de la DRL, tels que l'inefficacité de l'échantillonnage, les préoccupations de sécurité et la scalabilité. En outre, nous avons exploré

la combinaison de la DRL avec des techniques formelles, un domaine de recherche prometteur pour garantir la sécurité et la fiabilité des applications de DRL.

Pour aider les praticiens à appliquer efficacement la DRL dans leur travail, nous avons fourni des recommandations pour commencer par des problèmes simples, choisir les algorithmes et architectures appropriés, prêter attention à la sécurité et à l'éthique, collaborer avec des experts et rester à jour avec les dernières recherches dans le domaine. En suivant ces recommandations, les praticiens peuvent appliquer efficacement la DRL dans une large gamme d'applications et contribuer au développement continu de ce domaine passionnant et en constante évolution.

Dans l'ensemble, la DRL est un domaine en constante évolution avec un potentiel significatif d'impact dans une large gamme d'applications. À mesure que le domaine continue d'évoluer, il est probable que de nouvelles techniques et approches émergeront pour faire avancer l'état de l'art et permettre de nouvelles applications de la DRL.

Cependant, il est important de noter que à mesure que les algorithmes de DRL deviennent plus puissants et complexes, il est nécessaire de s'assurer qu'ils sont sûrs, fiables et transparents. Cela nécessite non seulement des avancées dans les algorithmes et les techniques sous-jacentes, mais également une réflexion attentive sur les implications éthiques et sociétales des applications de DRL.

Nous espérons que cet article offre une introduction utile au domaine de la DRL, ses applications, ses limites et ses enjeux ouverts, et qu'il inspire de nouvelles recherches et développements dans ce domaine passionnant et en constante évolution.

# Références

[1] Hamoud Alshammari, Karim Gasmi, Moez Krichen, Lassaad Ben Ammar, Mohamed Osman Abdelhadi, Ammar Boukrara, and Mahmood A Mahmood. Optimal deep learning model for olive disease diagnosis based on an adaptive genetic algorithm. *Wireless Communications and Mobile Computing*, 2022 :1–13, 2022.

[2] Benjamin Beyret, Ali Shafti, and A Aldo Faisal. Dot-to-dot : Explainable hierarchical reinforcement learning for robotic manipulation. In *2019 IEEE/RSJ International Conference on intelligent robots and systems (IROS)*, pages 5014–5019. IEEE, 2019.

[3] Wadii Boulila, Maha Driss, Eman Alshanqiti, Mohamed Al-Sarem, Faisal Saeed, and Moez Krichen. Weight initialization techniques for deep learning algorithms in remote sensing : Recent trends and future perspectives. *Advances on Smart and Soft Computing : Proceedings of ICACIn 2021*, pages 477–484, 2022.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33 :1877–1901, 2020.

[5] Lontsi Saadio Cedric, Wilfried Yves Hamilton Adoni, Rubby Aworka, Jérémie Thouakesseh Zoueu, Franck Kalala Mutombo, Moez Krichen, and Charles Lebon Mberi Kimpolo. Crops yield prediction based on machine learning models : case of west african countries. *Smart Agricultural Technology*, page 100049, 2022.

[6] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[7] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3389–3396. IEEE, 2017.

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic : Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[9] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Int. conference on machine learning*, pages 2555–2565. PMLR, 2019.

[10] Junhwi Kim, Minhyuk Kwon, and Shin Yoo. Generating test input with deep reinforcement learning. In *Proceedings of the 11th International Workshop on Search-Based Software Testing*, pages 51–58, 2018.

[11] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

[12] Moez Krichen. *Model-based testing for real-time systems.* PhD thesis, PhD thesis, PhD thesis, Universit Joseph Fourier (December 2007), 2007.

[13] Moez Krichen. A formal framework for conformance testing of distributed real-time systems. In *International Conference On Principles Of Distributed Systems*, pages 139–142. Springer, 2010.

[14] Moez Krichen. *Contributions to model-based testing of dynamic and distributed real-time systems.* PhD thesis, École Nationale d'Ingénieurs de Sfax (Tunisie), 2018.

[15] Moez Krichen, Omar Cheikhrouhou, Mariam Lahami, Roobaea Alroobaea, and Afef Jmal Maâlej. Towards a model-based testing framework for the security of internet of things for smart city applications. In *Smart Societies, Infrastructure, Technologies and Applications : First International Conference, SCITA 2017, Jeddah, Saudi Arabia, November 27–29, 2017, Proceedings 1*, pages 360–365. Springer International Publishing, 2018.

[16] Moez Krichen, Mariam Lahami, and Qasem Abu Al-Haija. Formal methods for the verification of smart contracts : A review. In *2022 15th International Conference on Security of Information and Networks (SIN)*, pages 01–08. IEEE, 2022.

[17] Moez Krichen, Afef Jmal Maâlej, and Mariam Lahami. A model-based approach to combine conformance and load tests : an ehealth case study. *International Journal of Critical Computer-Based Systems*, 8(3-4) :282–310, 2018.

[18] Moez Krichen, Seifeddine Mechti, Roobaea Alroobaea, Elyes Said, Parminder Singh, Osamah Ibrahim Khalaf, and Mehedi Masud. A formal testing model for operating room control system using internet of things. *Computers, Materials &*

*Continua*, 66(3) :2997–3011, 2021.

[19] Moez Krichen, Alaeddine Mihoub, Mohammed Y Alzahrani, Wilfried Yves Hamilton Adoni, and Tarik Nahhal. Are formal methods applicable to machine learning and artificial intelligence ? In *2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pages 48–53. IEEE, 2022.

[20] Moez Krichen and Stavros Tripakis. Interesting properties of the real-time conformance relation tioco. In *Theoretical Aspects of Computing-ICTAC 2006 : Third International Colloquium, Tunis, Tunisia, November 20-24, 2006. Proceedings 3*, pages 317–331. Springer Berlin Heidelberg, 2006.

[21] Mariam Lahami, Moez Krichen, Mariam Bouchakwa, and Mohamed Jmaiel. Using knapsack problem model to design a resource aware test architecture for adaptable and distributed systems. In *Testing Software and Systems : 24th IFIP WG 6.1 International Conference, ICTSS 2012, Aalborg, Denmark, November 19-21, 2012. Proceedings 24*, pages 103–118. Springer Berlin Heidelberg, 2012.

[22] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1) :1334–1373, 2016.

[23] Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv :1606.01541*, 2016.

[24] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv :1509.02971*, 2015.

[25] Afef Jmal Maâlej, Manel Hamza, Moez Krichen, and Mohamed Jmaiel. Automated significant load testing for ws-bpel compositions. In *2013 IEEE sixth international conference on software testing, verification and validation workshops*, pages 144–153. IEEE, 2013.

[26] Afef Jmal Maâlej and Moez Krichen. A model based approach to combine load and functional tests for service oriented architectures. In *VECoS*, pages 123–140, 2016.

[27] Afef Jmal Maâlej, Moez Krichen, and Mohamed Jmaiel. Conformance testing of ws-bpel compositions under various load conditions. In *2012 IEEE 36th annual computer software and applications conference*, pages 371–371. IEEE, 2012.

[28] Afef Jmal Maâlej, Moez Krichen, and Mohamed Jmaiel. Model-based conformance testing of ws-bpel compositions. In *2012 IEEE 36th annual computer software and applications conference workshops*, pages 452–457. IEEE, 2012.

[29] Afef Jmal Maâlej, Mariam Lahami, Moez Krichen, and Mohamed Jmaïel. Distributed and resource-aware load testing of ws-bpel compositions. In *ICEIS (2)*, pages 29–38, 2018.

[30] Akshita Mittel and Purna Sowmya Munukutla. Visual transfer between atari games using competitive reinforcement learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 499–

501, 2019.

[31] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[32] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540) :529–533, 2015.

[33] Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv :1506.08941*, 2015.

[34] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic : Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv :1511.06342*, 2015.

[35] Zedong Peng, Yi Zhang, Yiping Feng, Tuchao Zhang, Zhengguang Wu, and Hongye Su. Deep reinforcement learning approach for capacitated supply chain optimization under demand uncertainty. In *2019 Chinese Automation Congress (CAC)*, pages 3512–3517. IEEE, 2019.

[36] KJ Prabuchandran, Hemanth Kumar AN, and Shalabh Bhatnagar. Multi-agent reinforcement learning for traffic signal control. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2529–2534. IEEE, 2014.

[37] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. *NPJ digital medicine*, 1(1) :18, 2018.

[38] Andrea Romdhana, Alessio Merlo, Mariano Ceccato, and Paolo Tonella. Deep reinforcement learning for black-box testing of android apps. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 31(4) :1–29, 2022.

[39] John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. *International Conference on Machine Learning*, 37(1) :1889–1897, 2015.

[40] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587) :484–489, 2016.

[41] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419) :1140–1144, 2018.

[42] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton,

et al. Mastering the game of go without human knowledge. *nature*, 550(7676) :354–359, 2017.

[43] Peter Stone and Manuela Veloso. Multiagent systems : A survey from a machine learning perspective. *Autonomous Robots*, 8 :345–383, 2000.

[44] Richard S Sutton and Andrew G Barto. *Reinforcement Learning : An Introduction*. MIT Press, 2018.

[45] Jiaqi Xiang, Qingdong Li, Xiwang Dong, and Zhang Ren. Continuous control with deep reinforcement learning for mobile robot navigation. In *2019 Chinese Automation Congress (CAC)*, pages 1501–1506. IEEE, 2019.

[46] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell : Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[47] Li Zhang, Flood Sung, Feng Liu, Tao Xiang, Shaogang Gong, Yongxin Yang, and Timothy M Hospedales. Actor-critic sequence training for image captioning. *arXiv preprint arXiv :1706.09601*, 2017.