Proceedings of CONF-SEML 2025 Symposium: Machine Learning Theory and Applications
DOI: 10.54254/2755-2721/2025.TJ23321

Policy Gradient Methods in Deep Reinforcement Learning

Yuhan Gao

School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China
Yuhan.Gao23050406@outlook.com

Abstract: Policy gradient (PG) methods are a fundamental component of deep reinforcement learning (DRL), particularly effective in continuous and high-dimensional control tasks. This paper presents a structured review of PG algorithms, tracing their development from basic Monte Carlo methods like REINFORCE to advanced techniques such as asynchronous advantage actor-critic (A3C), trust region policy optimization (TRPO), proximal policy optimization (PPO), deep deterministic policy gradient (DDPG), and soft actor-critic (SAC). These methods differ in terms of policy structure, optimization stability, and sample efficiency, addressing core challenges in policy learning through gradient-based updates. In addition, this review explores the application of PG methods in real-world domains, including autonomous driving, financial portfolio management, and smart grid energy systems. These applications demonstrate PG methods' capacity to operate under uncertainty and adapt to complex dynamic environments. However, limitations such as high variance, low sample efficiency, and instability in multi-agent and offline settings remain significant obstacles. The review concludes by outlining emerging research directions, including entropy-based exploration, model-based policy optimization, meta-learning, and Transformer-based sequence modeling. This work aims to offer theoretical insights and practical guidance to support the continued advancement and application of policy gradient methods in reinforcement learning.

Keywords: Policy Gradient, Deep Reinforcement Learning, Actor-Critic Algorithms, Sample Efficiency, Multi-Agent Systems.

1. Introduction

Reinforcement Learning (RL) is a learning paradigm that combines principles from artificial intelligence, statistics, optimization, and dynamic systems theory [1]. It models the way humans and animals learn through interactions with their environment [1]. The core idea of RL lies in enabling an agent to perceive the environment, take actions, receive rewards, and adjust its behavior policy to achieve specific goals [1]. The Markov Decision Process (MDP) provides a theoretical modeling framework for this process, encompassing perception, action, and goal achievement [1]. A typical RL system consists of key components such as the agent, environment, policy, reward signal, value function, and an optional environment model [1].

Unlike supervised learning, which relies on labeled data, and unsupervised learning, which seeks to uncover latent structures, reinforcement learning focuses on maximizing cumulative rewards through trial-and-error interactions with the environment [1]. Consequently, RL is widely regarded as the third paradigm in machine learning, complementing supervised and unsupervised learning [1].

DOI: 10.54254/2755-2721/2025.TJ23321

However, traditional RL methods are limited in handling high-dimensional state spaces and continuous action control tasks [2]. Inspired by dopamine-based signaling mechanisms in biological neural systems, researchers have proposed Deep Reinforcement Learning (DRL), which integrates deep neural networks to enhance perception and decision-making capabilities [2]. DRL enables agents to learn directly from high-dimensional sensory inputs and demonstrate superior performance in complex control tasks [2].

To address the limitations of traditional RL methods in continuous control and high-dimensional state spaces, researchers have proposed a range of advanced algorithmic frameworks. Among various DRL approaches, Policy Gradient (PG) methods have emerged as a prominent class of algorithms in DRL due to their capability to directly optimize parameterized policies [3]. PG methods typically perform policy optimization using stochastic gradient ascent [3]. Compared to value-based methods, PG methods offer greater adaptability and expressiveness in continuous action spaces and enable smoother policy updates [3]. However, PG methods also suffer from high variance in gradient estimation and low sample efficiency [4]. Moreover, excessively large policy updates may lead to performance collapse, affecting convergence and stability [4].

This paper provides a comprehensive review of policy gradient methods in deep reinforcement learning. It systematically analyzes their theoretical foundations and representative algorithms along with their applications across diverse domains, including path planning, financial portfolio management, and electricity market pricing. The review also identifies current challenges and outlines future research directions, offering valuable insights for both academic researchers and practical implementations.

2. Policy Gradient methods

2.1. Fundamentals of Policy Gradient methods

Policy Gradient (PG) methods are a fundamental class of reinforcement learning algorithms that optimize policy parameters by computing the gradient of expected cumulative rewards with respect to those parameters. The REINFORCE algorithm, introduced by Williams [3], is a canonical example that estimates policy gradients using sampled returns and updates parameters via the log-probability gradient scaled by return. As a Monte Carlo method, REINFORCE operates over complete episodes, using full trajectory returns to compute unbiased gradients. However, it suffers from high variance, especially in long-horizon tasks, which can impede convergence. To mitigate this, a baseline function—typically the state value—can be subtracted from returns without introducing bias, effectively reducing variance and stabilizing learning [3].

While stochastic PG methods suit discrete action spaces, many real-world problems involve continuous control. To address this, Lillicrap et al. proposed Deep Deterministic Policy Gradient (DDPG), which combines deterministic policy updates with deep function approximators [2]. DDPG extends the actor-critic framework to continuous actions, using a deterministic actor and a critic that estimates the Q-value. The algorithm also employs target networks and experience replay, which reduce variance and enhance training stability in off-policy settings.

Advancements in value estimation have further improved sample efficiency and stability. Schulman et al. introduced Generalized Advantage Estimation (GAE), which refines the advantage function using a weighted combination of temporal-difference errors [4]. GAE introduces a tunable bias-variance trade-off, enabling more stable and efficient learning in high-dimensional environments. Concurrently, Mnih et al. proposed Asynchronous Advantage Actor-Critic (A3C), which updates policy and value networks using multiple actor-learners operating asynchronously, accelerating learning and reducing data correlation [5].

DOI: 10.54254/2755-2721/2025.TJ23321

Another key innovation in PG methods is the integration of entropy into the objective function. Entropy regularization encourages more diverse behaviors, reducing the risk of premature convergence to suboptimal policies. Soft Actor-Critic (SAC), proposed by Haarnoja et al., incorporates this idea into an off-policy actor-critic framework. By maximizing a weighted sum of expected return and policy entropy, SAC improves robustness and sample efficiency, especially in continuous control tasks where exploration is vital [6].

2.2. Key Policy Gradient algorithms

REINFORCE remains the foundational policy gradient method and is often used as a baseline in algorithmic comparisons [3]. It offers simplicity and generality but suffers from high sample variance due to its reliance on Monte Carlo return estimates. The method also struggles in tasks with sparse or delayed rewards, where full trajectory feedback is less informative. To address these limitations, asynchronous actor-critic methods such as A3C introduce multiple parallel learners that share and update global policy and value networks asynchronously [5]. This parallelism increases stability and speeds up convergence by decorating data and improving exploration.

To further enhance training reliability, Schulman et al. proposed Trust Region Policy Optimization (TRPO), which imposes a constraint on policy updates by bounding the Kullback–Leibler (KL) divergence between successive policies [7]. This ensures each policy update remains within a "trust region," preserving training stability and yielding monotonic performance improvement. However, TRPO's constrained optimization introduces computational overhead, limiting its practicality.

To address this, the same authors introduced Proximal Policy Optimization (PPO) [8], which simplifies TRPO by replacing hard KL constraints with a clipped surrogate objective. PPO retains the stability of TRPO while significantly improving practical usability through simplified implementation and reduced computational overhead [8]. Its simplicity and empirical performance have made PPO one of the most widely used algorithms in DRL.

For problems requiring fine-grained control in continuous action spaces, DDPG offers a deterministic variant of the policy gradient approach [2]. It combines a deterministic actor network with a Q-function critic and trains both using off-policy data from a replay buffer. Although DDPG is capable of learning in high-dimensional action spaces, it is sensitive to hyperparameters and exploration noise, which can lead to instability.

SAC extends the actor-critic architecture by adopting a stochastic policy and integrating an entropy-based objective [6]. This maximum entropy framework encourages broader exploration and enables better coverage of the action space. SAC also employs twin Q-networks and a temperature parameter to balance reward and entropy, offering improved performance and sample efficiency in a range of complex, continuous tasks.

A comparative summary of the key characteristics of these policy gradient algorithms is presented in Table 1. The table highlights differences in policy formulation, applicability to action spaces, training strategies, and practical considerations.

Table 1: Comparative summary of key Policy Gradient algorithms in Deep Reinforcement Learning

Algorithm	Policy Type	Action Space	On/ Off Policy	Entropy Regularization	Sample Efficiency	Remarks
REINFORCE	Stochastic	Discrete	On-policy	No	Low	High variance, simple
A3C	Stochastic	Discrete/Continuous	On-policy	No	Medium	Asynchronous updates
TRPO	Stochastic	Continuous	On-policy	No	Medium	KL constraint ensures stability

DOI: 10.54254/2755-2721/2025.TJ23321

Table 1: (continued)

PPO	Stochastic	Continuous	On-policy	No	High	Practical, widely used
DDPG	Deterministic	Continuous	Off-policy	No	High	Sensitive to hyperparameters
SAC	Stochastic	Continuous	Off-policy	Yes	Very High	Maximum entropy objective

3. Application domains of PG methods

3.1. Autonomous driving and path planning

PG methods have demonstrated strong applicability in autonomous systems, particularly in trajectory generation and path planning for autonomous vehicles, UAVs, and mobile robots [9]. Traditional control algorithms like Proportional–Integral–Derivative, Dijkstra, and A* often perform well in static settings but face limitations in dynamic, uncertain, and continuous action environments due to their lack of adaptability and inability to optimize behavior policies directly [9].

PG methods, in contrast, allow agents to learn through environmental interactions and optimize parameterized policies to maximize cumulative rewards. DDPG is a representative algorithm designed for continuous action control. In autonomous driving, DDPG can be employed to control steering angles, throttle, and braking by directly mapping raw sensor inputs to continuous actions [10]. For instance, DDPG has been successfully applied in vision-based path tracking systems, outperforming classical control schemes by enabling smooth trajectory following and reduced lane deviation errors [10].

SAC, leveraging maximum entropy reinforcement learning, introduces entropy-based regularization to encourage exploration and improve sample efficiency [6]. SAC has been shown to stabilize policy learning in highly dynamic driving scenarios, such as obstacle avoidance with moving targets or varying weather conditions [6]. Multi-agent systems also benefit from PG methods in cooperative path planning, where decentralized agents learn to coordinate without centralized control. In such tasks, each agent uses local observations and learns a policy that contributes to global objectives like collision-free routing or energy-efficient movement [11].

The integration of PG methods into autonomous driving not only improves control precision but also enhances decision-making under uncertainty, marking a transition from rule-based to learning-based paradigms in intelligent mobility.

3.2. Financial investment and portfolio optimization

The financial domain presents a highly dynamic and non-stationary environment, where decision-making tasks include portfolio optimization, asset allocation, dynamic hedging, and risk-sensitive investment strategy formulation [12-14]. Traditional methods such as mean-variance optimization or rule-based rebalancing fail to adapt rapidly to market changes. PG methods are increasingly adopted due to their ability to optimize return-driven policies in continuous action spaces—a critical requirement in financial applications where decisions often involve proportionally allocating capital across multiple assets [13].

DDPG has been applied to portfolio management, where the agent observes market indicators and outputs continuous asset allocation weights [14]. Empirical studies show that DDPG-based models outperform baselines in maximizing the Sharpe ratio and reducing maximum drawdown [14]. SAC and PPO have also been used to design robust trading policies under volatility by leveraging entropy-

DOI: 10.54254/2755-2721/2025.TJ23321

regularized exploration and stable policy updates [13-16]. In high-frequency trading, these methods enable learning from tick-level data and adapt to market microstructure changes in real time.

A particularly promising direction is Meta-Policy Gradient (Meta-PG), which enables agents to generalize across market regimes. By encoding prior experience across tasks, Meta-PG can quickly adapt to new market conditions using minimal data [14]. For example, it can retrain a financial policy in response to sudden interest rate shifts or geopolitical shocks without full model retraining. These approaches offer key advantages in financial risk management and policy transferability.

3.3. Energy pricing and smart grid control

In the energy domain, PG methods have found increasing applications in electricity pricing, distributed energy resource management, and smart grid scheduling. Tasks such as energy dispatch, price optimization, and aggregator coordination involve continuous decision-making under uncertainty, making PG algorithms a natural fit [16].

DDPG has been deployed in electric vehicle aggregator systems, where the agent learns charging and discharging policies in response to real-time electricity prices and vehicle battery states [16]. PPO has been applied to maximize grid revenue while maintaining energy balance in the face of fluctuating demand and supply [16]. SAC has shown strong performance in managing battery energy storage systems, ensuring real-time load balancing while optimizing operational efficiency [17]. These models allow energy providers to respond dynamically to peak loads and improve energy utilization efficiency.

Multi-agent scenarios are also prevalent in power systems. For example, in distributed microgrids, each generation or load unit acts as an independent agent. PG-based multi-agent frameworks enable decentralized control and cooperative pricing strategies through shared reward mechanisms [11]. This supports a more resilient and adaptive grid infrastructure.

4. Challenges and future trends in PG methods

Despite their success in domains such as autonomous driving, finance, and energy systems, PG methods face common limitations, including low sample efficiency, high variance, and coordination complexity in multi-agent scenarios. This section summarizes these key challenges and outlines emerging solutions.

4.1. Sample efficiency and variance reduction

A core limitation of PG methods is high variance in gradient estimation, leading to unstable updates and slow convergence [4]. In domains like robotics, healthcare, and energy systems, data collection is costly or constrained by safety, making sample efficiency a critical concern [15, 19]. Variance reduction strategies include using baseline estimators such as value and advantage functions. GAE improves variance control by combining multi-step returns with exponential weighting [4], helping stabilize updates and reduce inefficiency. Algorithms like SAC add entropy regularization to encourage exploration while maintaining robustness [6]. Replay buffers improve sample reuse via off-policy training, and model-based extensions reduce reliance on real interactions by simulating virtual rollouts from learned dynamics [18]. Transformers have been introduced to model long-range temporal dependencies in state-action sequences, enhancing policy generalization. In tasks like multi-step planning, asset reallocation, and energy dispatch, Transformer-based policies outperform conventional architectures by better modeling sequential decision logic [19].

DOI: 10.54254/2755-2721/2025.TJ23321

4.2. Offline Policy Gradient challenges

Offline RL aims to learn policies from static datasets without environmental interaction. However, PG methods suffer from value overestimation and distributional shifts between behavior and target policies [15]. Conservative Q-Learning addresses this by penalizing Q-values for out-of-distribution actions, promoting safer policies [20]. Offline SAC and DDPG variants use entropy regularization, behavior cloning loss, and action clipping to stabilize training and reduce overfitting [6]. Alternative algorithms include Batch-Constrained Q-learning, which limits exploration to dataset-supported actions, and Advantage-Weighted Actor-Critic, which prioritizes high-reward trajectories via advantage-weighted updates [20]. Offline Policy Learning adds trust-region constraints and divergence measures to improve stability. Decision Transformer reframes offline RL as sequence modeling, treating trajectories as token sequences and predicting actions based on return-to-go, achieving strong results without value functions [19]. Hybrid offline-to-online approaches combine offline pretraining with online finetuning. Meta-PG further improves generalization by learning task-adaptive priors that rapidly adapt to distributional shifts [14].

4.3. Multi-agent Policy Gradient issues

In multi-agent systems, the environment is non-stationary, as each agent's policy affects others' observations and rewards [11]. This increases training instability and hinders convergence. The credit assignment problem further complicates identifying individual contributions in cooperative settings [11]. MADDPG addresses these challenges through centralized training and decentralized execution. Centralized critics use global information to guide learning, while decentralized actors operate independently at deployment [11]. MAPPO extends PPO with shared baselines and clipped updates, enhancing scalability and stability [11]. Other advances include value decomposition for reward allocation and message-passing for inter-agent communication and coordination [11]. Hierarchical PG architectures divide control into strategic and tactical layers, where high-level agents set sub-goals and low-level agents optimize execution [11]. Transformer-based multi-agent systems model inter-agent attention, enabling agents to capture spatial and temporal dependencies and improve coordination under partial observability [11].

4.4. Integrated trends in Policy Gradient methods

Recent trends in PG research emphasize combining model-based dynamics learning, meta-learning, and Transformer-based representations. These synergistic frameworks offer enhanced sample efficiency, generalization, and adaptability. A key direction is the convergence of offline learning paradigms with multi-agent coordination through hierarchical and attention-based policy modeling.

4.5. Open problems and future directions

Despite recent progress, key challenges in policy gradient methods remain. First, scaling to multi-agent systems under communication constraints and partial observability is difficult, requiring efficient coordination and stable learning architectures. Second, in dynamic environments like finance and energy, policies often fail to generalize due to non-stationarity, necessitating methods that prevent overfitting and enable fast adaptation. Third, cross-domain policy transfer remains limited, highlighting the need for better abstraction, meta-learning, and transferable representations. Addressing these challenges is essential for deploying PG methods in real-world, safety-critical applications.

DOI: 10.54254/2755-2721/2025.TJ23321

5. Conclusions

This paper presents a comprehensive review of PG methods in DRL, covering their theoretical foundations, representative algorithms, practical applications, and key challenges. Beginning with REINFORCE as a canonical Monte Carlo approach, we examined the evolution of PG methods through variance reduction techniques like GAE, scalable architectures such as A3C and PPO, and sample-efficient algorithms including DDPG and SAC. This paper also explored their deployment in domains like autonomous driving, financial portfolio management, and energy systems—where continuous control and decision-making under uncertainty are essential.

Despite notable progress, PG methods still face core limitations: low sample efficiency, high variance in gradient estimation, and instability in learning—especially in environments with sparse rewards or safety constraints. In multi-agent settings, non-stationarity and credit assignment complicate training. Offline policy learning remains challenging due to distributional shifts and overestimation, though recent methods such as Decision Transformers and conservative Q-learning show promise.

Future research should focus on improving sample efficiency through model-based learning and transformer-based trajectory modeling to enhance applicability in costly or risky domains. Hybrid offline-to-online regimes and meta-policy optimization can increase generalization and adaptability. Finally, scaling PG methods to complex multi-agent systems under partial observability and communication constraints requires advances in coordination, reward decomposition, and hierarchical control. Addressing these challenges is essential for realizing the full potential of PG methods in safety-critical, dynamic real-world systems.

References

[1] *Connell, J. H., & Mahadevan, S. (1997). Robot learning. Kluwer Academic Publishers.*

[2] *Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.*

[3] *Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3–4), 229–256.*

[4] *Schulman, J., Moritz, P., Levine, S., Jordan, M., & Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438.*

[5] *Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. arXiv preprint arXiv:1602.01783.*

[6] *Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv preprint arXiv:1801.01290.*

[7] *Schulman, J., Levine, S., Moritz, P., Jordan, M. I., & Abbeel, P. (2015). Trust region policy optimization. arXiv preprint arXiv:1502.05477.*

[8] *Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.*

[9] *Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A. A. A., Yogamani, S., & Pérez, P. (2020). Deep reinforcement learning for autonomous driving: A survey. arXiv preprint arXiv:2002.00444.*

[10] *Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv preprint arXiv:1801.01290.*

[11] *Fonseca, T., Ferreira, L., Cabral, B., Severino, R., & Praca, I. (2024). EnergAIze: Multi-agent deep deterministic policy gradient for vehicle-to-grid energy management. arXiv preprint arXiv:2404.02361.*

[12] *Naga, H. (2025). Reinforcement learning: Concepts and real-world applications. Global Research Review, 1(1), 142–151. Retrieved from https://scitechpublications.org/index.php/grr/article/view/21*

[13] *Goodness, C. (2025). Artificial intelligence and machine learning in finance: Enhancing efficiency, innovation and decision-making. World Journal of Advanced Engineering Technology and Sciences, 14(3), 134–139.*

[14] *He, J., Hua, C., Zhou, C., & Zheng, Z. (2025). Reinforcement-learning portfolio allocation with dynamic embedding of market information. arXiv preprint arXiv:2501.17992.*

[15] *Levine, S., Kumar, A., Tucker, G., & Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643.*

DOI: 10.54254/2755-2721/2025.TJ23321

[16] *Kamrani, A. S., Dini, A., Dagdougui, H., & Sheshyekani, K. (2025). Multi-agent deep reinforcement learning with online and fair optimal dispatch of EV aggregators. Machine Learning with Applications, 19, 100620.*
[17] *Shojaeighadikolaei, A., Talata, Z., & Hashemi, M. (2024). Centralized vs. decentralized multi-agent reinforcement learning for enhanced control of electric vehicle charging networks. arXiv preprint arXiv:2404.12520.*
[18] *Gu, S., Holly, E., Lillicrap, T., & Levine, S. (2016). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. arXiv preprint arXiv:1610.00633.*
[19] *Kochliaridis, V., Kostinoudis, E., & Vlahavas, I. (2024). Optimizing pretrained transformers for autonomous driving. In Proceedings of the ACM (pp. 1–9).*
[20] *Eldeeb, E., & Alves, H. (2025). Multi-agent meta-offline reinforcement learning for timely UAV path planning and data collection. arXiv preprint arXiv:2501.16098.*