Goal Achievement Guided Exploration: Mitigating Premature Convergence in Learning Robot Control

Shengchao Yan*, **Baohe Zhang*** **& Joschka Boedecker**
Department of Computer Science
University of Freiburg
Freiburg, 79110, Germany
{yan,zhangb,jboedeck}@cs.uni-freiburg.de

Wolfram Burgard
Department of Engineering
University of Technology Nuremberg
Nürnberg, 90443, Germany
{wolfram.burgard}@utn.de

Abstract: Premature convergence to suboptimal policies remains a significant challenge in reinforcement learning (RL), particularly for robots with many degrees of freedom and in tasks with non-convex reward landscapes. Existing work usually utilizes reward shaping to encourage exploring promising spaces. However, this may inadvertently introduce new local optima and impair the optimization for the actual target reward. To address this issue, we propose Goal Achievement Guided Exploration (GAGE), a novel approach that incorporates an agent's goal achievement as a dynamic criterion for balancing exploration and exploitation. GAGE adaptively adjusts the exploitation level based on the agent's current performance relative to an estimated optimal performance, thereby mitigating premature convergence. Extensive evaluations demonstrate that GAGE substantially improves learning outcomes across various challenging whole-body control tasks by adapting convergence based on task success. GAGE can seamlessly integrate into existing RL frameworks, highlighting its potential as a versatile tool for enhancing exploration strategies in RL for robot control.

Keywords: Robots, Reinforcement Learning, Exploration

1 Introduction

Properly dealing with the exploration-exploitation trade-off in reinforcement learning (RL) still is a critical challenge [1, 2]. Constrained by learning time and resources, the agent must balance well between exploring for better policies and exploiting the learned behaviors. There are two prominent challenges in exploration: sparse reward function and local optima. Since we usually provide the agent with dense reward signals, the problem of local optima dominants in robot control tasks. An environment riddled with local optima may provide the agent with redundant or misleading information and distract it from exploring the actual optimization target. For example, in robot locomotion tasks, where robots are rewarded for saving energy in addition to the main speed reward, agents may focus on optimizing energy consumption but only move slowly. Agents trained in environments with local optima are more prone to over-exploitation, leading to premature convergence to a suboptimal solution.

* Denotes Equal Contribution

CoRL 2024 Workshop on Whole-Body Control and Bimanual Manipulation (CoRL 2024 WCBM).

Due to RL's trial-and-error nature, local optima can make the learning process unstable. This instability has been reported as a significant obstacle when reproducing and comparing different RL algorithms [3]. It is important to distinguish this issue from reward hacking [4], where the agent discovers policies that maximize returns in ways the system designer did not anticipate or desire. We focus on premature convergence, where local optima prevent the agents from optimizing the targeted returns. To effectively solve tasks with local optima, preventing premature convergence during exploration is essential. Several factors contribute to this issue, including the inherent non-convexity of tasks, reward shaping, multi-objectives, and function approximation errors introduced by neural networks in deep RL algorithms.

Many methods have been developed to address the exploration-exploitation trade-off [2], but not explicitly for premature convergence. One popular approach, ϵ-greedy, employs a predefined time-decaying parameter ϵ to decrease exploration gradually. However, finding the optimal schedule is far from trivial, as it can vary depending on the task and is further complicated by the unstable nature of reinforcement learning processes. Other methods, such as Proximal Policy Optimization (PPO)[5] and Soft Actor-Critic (SAC)[6], incorporate an entropy-loss component to promote exploration, but this acts only as a soft learning regularizer. Similarly, curiosity-driven intrinsic rewards [7, 8] encourage exploration, yet the exploitation process still follows the behavior of the underlying algorithms, like Deep Q-learning (DQN) [9] and PPO, which are prone to premature convergence.

To address the issue of converging to a suboptimal solution prematurely, we propose a novel approach that incorporates an agent's *goal achievement* into its exploration-exploitation strategy. Goal achievement is defined as the ratio of an agent's current policy return to the optimal policy return, excluding the auxiliary rewards for guiding the learning process. Unlike existing approaches, which often overlook this critical aspect of guiding exploration, our approach ensures that exploration continues when the agent's goal achievement is low, thereby preventing early convergence to suboptimal policies.

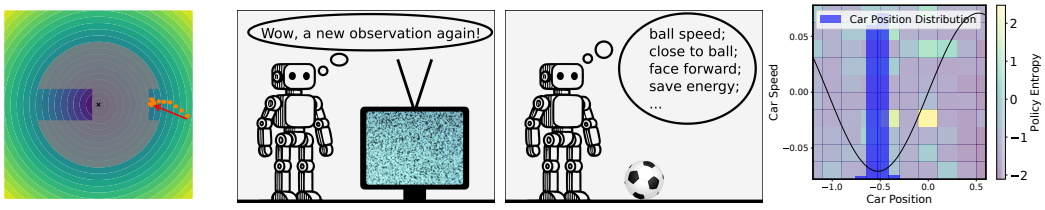
To summarize, in this work, we first investigate the various factors that contribute to premature convergence in RL. We analyze existing exploration-exploitation methods and explain why they fail to prevent premature convergence. To solve the problem, we propose **G**oal **A**chievement **G**uided **E**xploration (GAGE), a method that leverages an agent's goal achievement to define an adaptive exploration schedule during training. We evaluate GAGE across multiple challenging continuous whole-body control tasks. The results demonstrate that GAGE consistently mitigates premature convergence, especially in complex exploration problems with many local optima. Moreover, GAGE's simplicity and compatibility with a wide range of existing RL algorithms distinguish it as a promising solution for enhancing exploration-exploitation strategies.

2 Premature Convergence and Exploration Techniques

Premature convergence is a common issue in optimization and machine learning algorithms like genetic algorithms [10] and reinforcement learning. Despite extensive efforts to enhance exploration efficiency in reinforcement learning (RL) [11, 12, 13], agents may still unintentionally converge to local optima due to various factors. In this section, we identify these factors and examine why existing exploration techniques remain prone to this problem.

2.1 Factors for Premature Convergence

Non-convexity of tasks Non-convexity exists in most real-world tasks and arises from different components, such as the reward function and transition dynamics. It also inherently stems from neural networks, the core part of deep reinforcement learning (DRL). Due to the non-convexity of DRL, sub-optimal solutions can exist even in simple problems. For example, as shown in Fig. 1a, an agent (orange dot) needs to avoid the grey-colored area and is rewarded more when getting closer to the circle's center. However, due to the non-convexity of the reward landscape, agents without sufficient exploration can get stuck in a local optimal solution [14]. In more complex contexts like robotics or traffic management [15, 16], systems often have many degrees of freedom and



(a) non-convexity (b) reward shaping (c) multi-objective (d) approximation error

Figure 1: Factors for premature convergence in deep reinforcement learning.

complex environmental interactions, making dynamics models non-convex and further complicating optimization.

Reward shaping Tasks with sparse rewards and local optima present significant challenges for exploration and credit assignment. To provide agents with dense and informative feedback, previous work has employed reward shaping based on prior knowledge or specific heuristics [17, 18]. While reward shaping can guide agents toward more valuable regions and accelerate convergence to the optimal policy, designing such rewards is tedious and may introduce local optima [19, 20, 21]. Agents might focus on auxiliary rewards while neglecting the actual task objectives [22]. Curiosity-based intrinsic rewards have become popular for enhancing exploration by rewarding agents for discovering new observations or acquiring new knowledge about the environment [23, 24]. This approach encourages agents to visit diverse states within environments. However, as illustrated in Fig. 1b, agents can become trapped by uncontrolled stochasticity in the system dynamics, a phenomenon known as the Noisy TV problem [25].

Multiple objectives Many real-world problems involve multiple, sometimes conflicting, objectives that cannot be adequately evaluated using a single metric. For example, as shown in Fig.1c, a robot learning to dribble a football has to optimize factors such as the ball's velocity, energy consumption, distance to the ball, and facing direction. Simultaneously optimizing all these metrics can lead to premature convergence to suboptimal solutions—for instance, the robot might stay close to the ball, face it, and remain stationary to save energy [26]. The presence of multiple objectives introduces local optima in the reward landscape, hindering the agent from reaching the global or Pareto optima, depending on the definition of the utility function [15, 27, 28]. In this paper, we focus on tasks with linear utility functions that can be addressed using single-objective algorithms rather than exploring Pareto fronts.

Function approximation error Neural networks as function approximators enable reinforcement learning (RL) to tackle extremely high-dimensional problems like Go [29]. However, they are prone to overfitting [30], and RL intensifies this issue due to its non-stationarity and biased datasets. As a result, even in simple tasks like MountainCar [31], modern algorithms such as Soft Actor-Critic (SAC) can suffer from insufficient exploration [32], collecting data only around the initial states (see Fig. 1d, where an SAC agent is trained for 1M steps). Due to premature convergence, the learned policy exhibits low entropy even in unvisited states and is thus unable to explore better solutions.

2.2 Exploration Techniques

Exploration methods for reinforcement learning can be categorized into two groups: undirected and directed [11]. Undirected exploration involves randomly selecting actions based merely on utility estimation. Whereas directed exploration utilizes knowledge of the learning process [8, 25, 33] to guide the exploration. In this section, we discuss popular exploration techniques from the two groups.

Undirected exploration **1)** The ϵ-greedy strategy, commonly used in value-based algorithms [9, 34], employs a time-decaying ϵ to define the probability of selecting either the best action or a

random one during training. However, tuning the schedule requires much effort, because many terms can influence the agent's training progress, and the exploration can hardly be defined by the number of iterations. **2)** Some reinforcement learning algorithms are equipped with an entropy loss term [5, 6] to enhance exploration. However, as a soft regularization for the learning process, it can be insufficient to guide the agent out of local optima. **3)** Noise-based techniques inject noise into the observation, action, or parameter space to enhance policy exploration [35, 36]. As the magnitude of the noise is controlled by either a time-decaying schedule or learned values, this method has limits similar to those of the previous two.

Directed exploration **1)** Curiosity-based methods [37] are widely employed in hard-exploration environments with sparse rewards. They reward the agent for exploring less visited states. Various approaches have been developed to estimate the novelty of a given state transition, such as the state's visitation number [38], the prediction error of a dynamics model [39], or the information gained through transitions [40]. However, they are mostly employed in discrete game environments and their effectiveness in continuous control has not been demonstrated clearly. **2)** Memory-based techniques navigate the agent to promising states as soon as possible through memorizing the visited states [23, 41, 33]. They reduce the number of frequently visited states near the initial ones, collecting more diverse data and thus mitigating premature convergence by reducing repeated data. However, these methods require high memory, as well as complex state compression and searching processes.

3 Goal Achievement Guided Exploration

The learning process should not converge before the agent approaches the maximum possible return. Therefore, it is natural that the convergence level, reflected by the concentration of the action distribution, is correlated to the goal achievement of the current policy. This section defines goal achievement $g(\pi)$ and explains how it can guide learning convergence.

3.1 Goal Achievement

A reward function is typically composed of several terms, each designed for different purposes. Some terms reflect the designer's actual goals, such as winning a game or achieving a target speed, while others, like curiosity-driven intrinsic rewards, are intended to guide the learning process. Goal achievement of the learning progress should be based on these actual goals, as they directly express the designer's objectives. In contrast, auxiliary rewards that are used to encourage exploration do not always align with these goals and can lead to suboptimal solutions as stated in noisy TV problem. Hence, we exclude these auxiliary rewards when measuring goal achievement. For n_g distinct actual reward terms, similar to multi-objective algorithms [15], we define the goal achievement for each of these rewards as:

$$g_i(\pi) = \frac{\mathbb{E}[V_\pi^{g_i}(s_0)]}{\mathbb{E}[V_*^{g_i}(s_0)]}, i \in \{1, \dots, n_g\} \text{ and } 0 \leq g_i(\pi) \leq 1 \tag{1}$$

where s_0 represents the initial state, g_i represents the goal achievement of the i-th objective among n_g performance metrics, and $V_\pi^{g_i}$ and $V_*^{g_i}$ are the i-th components of the vectorized value function for the current policy π and the optimal policy π^*, respectively. We focus on non-negative target rewards. For tasks with negative rewards, applying a sigmoid or an offset to the estimated performance can still guarantee that the goal achievement is between 0 and 1.

In this work, we define the overall goal achievement of the agent as the minimum among goal achievement in each reward term: $g(\pi) = \min(g_i(\pi)), i \in \{1, \dots, n_g\}$. This allows the converged police to optimize jointly all the task-relevant objectives.

In practice, the value function V_π often involves significant estimation errors, and computing V_* directly might also be infeasible. Therefore, we approximate V_π by using the average rewards from recent rollout trajectories. Determining the optimal performance V_* can often be achieved through

heuristics. For example, in many games, the optimal reward $r_{max,t}$ at each step t, up to the max episode length T, is predefined, such as a fixed value awarded for winning. We can then approximate the goal achievement for a certain reward given the current policy π by:

$$g(\pi) \approx \frac{\mathbb{E}_\pi \left[\sum_{t=0}^{T} r_t \right]}{\sum_{t=0}^{T} r_{\max,t}}. \tag{2}$$

Alternatively, when such explicit values are unavailable, the optimal performance can be estimated empirically based on observed performance, as further discussed in Sec. 4.

3.2 Mitigating Premature Convergence via Action Smoothing

To prevent the agent from prematurely converging to local optima and overcommitting to a limited set of actions when goal achievement is low, we apply an action smoothing technique inspired by label-smoothing regularization [42] for image classification, which reduces overconfidence by smoothing the predicted class distribution. This technique ensures that the agent's action distribution does not collapse into a narrow Gaussian peak in continuous spaces. Below, we discuss how to implement action smoothing in continuous control tasks.

In the learning process, exploration is typically facilitated by modeling the policy's action distribution as a Gaussian distribution. This approach is used in both stochastic policies like Soft Actor-Critic (SAC) and Proximal Policy Optimization (PPO) [5, 6], and deterministic policies like Deep Deterministic Policy Gradient (DDPG) [35], where the Gaussian distribution serves as additive noise for exploration. The policy learns the mean $\mu(s)$ of the action distribution, modeled as: $p(a \mid s) \sim \mathcal{N}(\mu(s), \sigma^2)$, where the standard deviation σ can be controlled via a schedule or learned as a parameter. The standard deviation directly represents the concentration of the action distribution. To prevent premature convergence, we define an adaptive lower bound $\sigma_{\mathrm{L}}(\pi)$ on σ, which is negatively correlated with the current policy's goal achievement $g(\pi)$:

$$\sigma_{\mathrm{L}}(\pi) = f(g(\pi)). \tag{3}$$

For simplicity, we employ a linear relationship between σ_{L} and the goal achievement g, leaving the investigation of other possible functions f for future work:

$$\sigma_{\mathrm{L}}(\pi) = -\sigma_0 g(\pi) + \sigma_0, \tag{4}$$

where $\sigma_0 > 0$ is a hyperparameter controlling the minimum allowed σ value when the goal achievement is zero. Agents with a higher σ_0 require more achievement to concentrate their policies. When $\sigma_0 = 0$, this is equivalent to the original algorithms without GAGE.

4 Experiments

This section validates the proposed method by addressing a range of problems characterized by local optima, which often lead to premature convergence in existing reinforcement learning algorithms. First, we apply our approach to solve complex continuous control tasks involving robots with high degrees of freedom. We then conduct ablation studies on the hyperparameters to assess the method's effectiveness in scenarios with unknown optimal goal achievement and evaluate the robustness of the learning process.

To evaluate the effectiveness of our method in preventing premature convergence, we designed five highly challenging continuous control tasks in IsaacLab [43, 26] (see Fig. 2). The specific environment details are provided in Appendix B We implemented our approach using Proximal Policy Optimization (PPO), building upon the IsaacLab framework. Since the original action standard deviations are independently learned parameters, we use goal achievement to set a dynamic lower bound for σ. This is accomplished by applying $\sigma = \sigma_{\mathrm{L}}(\pi)$ whenever it falls below the threshold. The full algorithm is outlined in Appendix A.

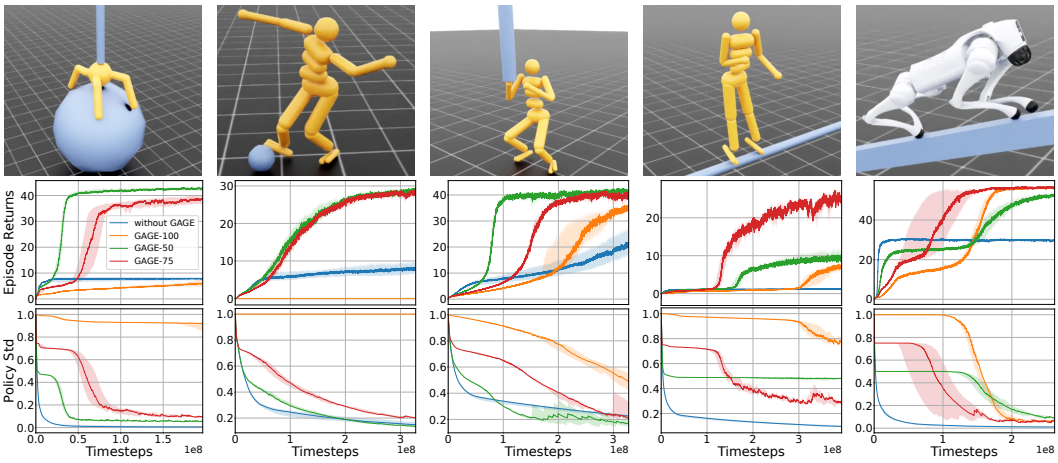
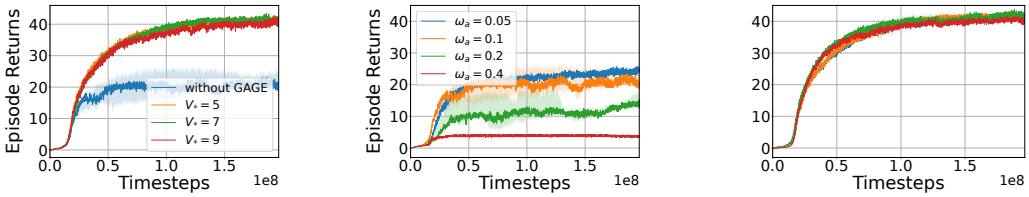


Figure 2: Continuous control experiments. We plot the median over 10 seeds, and the faint area represents the 25% and 75% quantiles, same for Fig. 3. **Top**: tasks from left to right, Ant Acrobatics, Humanoid Dribbling, Humanoid Pole, Humanoid Tightrope, and Dog (Unitree Go2) Balance Beam. **Middle**: Training curves of each method. **Bottom**: Standard deviation of each method $\bar{\sigma}$.



(a) Unknown Optimal Goal

(b) Robustness to Reward Shaping. **Left**: w/o GAGE. **Right**: GAGE.

Figure 3: Ablation Study of GAGE in Humanoid Locomotion Task

Explore Until Solved The training curves for episode returns and the average σ values across all robot joints are shown in Fig. 2. The proposed method successfully solved all the challenging tasks, whereas the baseline algorithm failed. Notably, PPO without our method is equivalent to GAGE with $\sigma_0 = 0$, and varying σ_0 can affect the learning process. However, the algorithm remains robust to this hyperparameter over a relatively wide range. The impact of our approach is evident in the plots of σ. The standard PPO rapidly reduces the policy's standard deviation at the beginning of training, achieving a higher reward by over-exploiting certain reward components, such as energy cost. It continues to reduce the policy's standard deviation even when the target reward plateaus. For instance, the dog robot learns to stand stably on the balance beam and ceases exploration despite having a forward movement target. In contrast, our method keeps exploring and only concentrates the action distribution with increased target rewards.

Unknown Optimal Goal For some tasks, the optimal performance is well-defined, such as achieving a score of 1 to win in board games. However, for other tasks, like a robot locomotion task, the optimal speed may not be straightforward and is often still under discovery. In the well-known humanoid locomotion task, to achieve higher speed without knowing the optimum, researchers often increase the weighting factors to locomotion speed. However, it can result in unnatural behaviors due to imbalanced speed and action rewards. To address this issue, we conducted experiments to demonstrate how GAGE can explore higher speeds without knowing the maximum or altering the reward weights. We define the goal achievement in the Humanoid task as $g_\pi = {}^{v_\pi}/_{v_*}$, where v_π and v_* represent the robot's current and target speeds, respectively. As shown in Fig. 3a, our method successfully increased the robot's running speed from less than 4 to 7 meters per second compared to PPO without GAGE, while maintaining natural behavior through balanced reward weights. When

the target speed is set to 5 m/s, which is below the learned optimal speed ($\sim$ 7 m/s), the GAGE agent is also able to learn the optimal speed. We assume this is because the robot successfully gets rid of the distracting local optima at the beginning of the learning process with lower speeds and inefficient gaits. As a result, it can continue explore higher speed behaviors after above 5m/s even if the sigma lower bound has reduced to zero. Even when the target speed is set to 9 m/s, higher than the known optimal speed, the GAGE agent is able to discover the optimal policy known so far without excessive exploration to meet the goal speed, thus avoiding divergence from the optimal policy.

Improved Robustness to Reward Shaping Reward shaping is crucial yet challenging in reinforcement learning, as even minor adjustments to the weighting of specific rewards can result in unsuccessful learning. Using the humanoid locomotion task, we demonstrate this issue and the effectiveness of our method in mitigating it. As in the previous experiment, we define goal achievement based on locomotion speed. The reward terms include a penalty for large action values, $\omega_a \|\boldsymbol{a}\|^2$. In the experiments, we kept the weights for other rewards constant while varying ω_a. The baseline agents without our method exhibited performance that was highly sensitive to changes in ω_a, with significant impacts on both final speed and locomotion gait. In contrast, our method enabled the agent to maintain high running speeds and achieve consistently high returns across all the ω_a values (see Fig. 3b).

5 Discussion

We introduced Goal Achievement Guided Exploration (GAGE), a method aiming to address premature convergence in reinforcement learning (RL) for robot control. Our approach uses goal achievement as a dynamic factor to guide the agent's exploration, allowing for a better balance between exploration and exploitation. Our experiments demonstrate that GAGE substantially mitigates premature convergence in challenging continuous control tasks by maintaining adequate exploration. Unlike traditional methods such as entropy maximization or curiosity-based exploration, GAGE incorporates an adaptive mechanism that smoothes the action probability distribution based on how well the agent achieves its goal. The strength of GAGE lies in its simplicity and compatibility with existing RL algorithms. It does not require significant architectural changes and can be easily integrated into different environments. The flexibility of GAGE makes it applicable to a wide variety of real-world robotics problems.

Despite these strengths, GAGE offers aspects for improvement. The current version relies on defining an appropriate goal achievement metric, which might not be straightforward in all tasks. In environments where the optimal policy or goal is poorly understood, the approximation of goal achievement might introduce inaccuracies. Additionally, while GAGE has proven effective in the tested environments, its scalability to more complex, high-dimensional tasks has yet to be explored.

Future research should focus on improving the scalability of GAGE and applying it to more complex, dynamic environments. Investigating non-linear relationships between the goal achievement and the standard deviation of Gaussian distributions could further enhance the method's adaptability to diverse RL problems.

References

[1] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[2] P. Ladosz, L. Weng, M. Kim, and H. Oh. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22, 2022. ISSN 1566-2535. doi:https://doi.org/10.1016/j.inffus.2022.03.003. URL https://www.sciencedirect.com/science/article/pii/S1566253522000288.

[3] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger. Deep reinforcement learning that matters. In S. A. McIlraith and K. Q. Weinberger, editors, *Proceedings*

of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 3207–3214. AAAI Press, 2018. doi:10.1609/AAAI.V32I1.11694. URL https://doi.org/10.1609/aaai.v32i1.11694.

[4] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016. URL http://arxiv.org/abs/1606.06565.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[6] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In J. G. Dy and A. Krause, editors, *Proc. of the International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

[7] A. G. Barto. Intrinsic motivation and reinforcement learning. In G. Baldassarre and M. Mirolli, editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 17–47. Springer, 2013. doi:10.1007/978-3-642-32375-1_2. URL https://doi.org/10.1007/978-3-642-32375-1_2.

[8] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2778–2787. PMLR, 2017. URL http://proceedings.mlr.press/v70/pathak17a.html.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. URL http://arxiv.org/abs/1312.5602.

[10] H. M. Pandey, A. Chaudhary, and D. Mehrotra. A comparative review of approaches to prevent premature convergence in GA. *Appl. Soft Comput.*, 24:1047–1077, 2014. doi:10.1016/J.ASOC.2014.08.025. URL https://doi.org/10.1016/j.asoc.2014.08.025.

[11] S. B. Thrun. Efficient exploration in reinforcement learning. Technical report, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, 1992.

[12] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[13] A. Agarwal, Y. Jin, and T. Zhang. Vo*q*l: Towards optimal regret in model-free rl with nonlinear function approximation. In G. Neu and L. Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 987–1063. PMLR, 12–15 Jul 2023. URL https://proceedings.mlr.press/v195/agarwal23a.html.

[14] D. Eckel, B. Zhang, and J. Bödecker. Revisiting safe exploration in safe reinforcement learning, 2024. URL https://arxiv.org/abs/2409.01245.

[15] J. Xu, Y. Tian, P. Ma, D. Rus, S. Sueda, and W. Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *Proc. of the International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 10607–10616. PMLR, 2020. URL http://proceedings.mlr.press/v119/xu20h.html.

[16] S. Yan, L. König, and W. Burgard. Agent-agnostic centralized training for decentralized multi-agent cooperative driving. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

[17] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, Ç. Gülçehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. P. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

[18] S. Yan, J. Zhang, D. Büscher, and W. Burgard. Efficiency and equity are both essential: A generalized traffic signal controller with deep reinforcement learning. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5526–5533, 2020.

[19] A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pages 278–287. Morgan Kaufmann, 1999.

[20] A. Gupta, A. Pacchiano, Y. Zhai, S. M. Kakade, and S. Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6255f22349da5f2126dfc0b007075450-Abstract-Conference.html.

[21] H. Ma, K. Sima, T. V. Vo, D. Fu, and T.-Y. Leong. Reward shaping for reinforcement learning with an assistant reward agent. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=a3XFF0PGLU.

[22] J. Lehman, J. Clune, D. Misevic, C. Adami, L. Altenberg, J. Beaulieu, P. J. Bentley, S. Bernard, G. Beslon, D. M. Bryson, N. Cheney, P. Chrabaszcz, A. Cully, S. Doncieux, F. C. Dyer, K. O. Ellefsen, R. Feldt, S. Fischer, S. Forrest, A. Frénoy, C. Gagné, L. K. L. Goff, L. M. Grabowski, B. Hodjat, F. Hutter, L. Keller, C. Knibbe, P. Krcah, R. E. Lenski, H. Lipson, R. MacCurdy, C. Maestre, R. Miikkulainen, S. Mitri, D. E. Moriarty, J. Mouret, A. Nguyen, C. Ofria, M. Parizeau, D. P. Parsons, R. T. Pennock, W. F. Punch, T. S. Ray, M. Schoenauer, E. Schulte, K. Sims, K. O. Stanley, F. Taddei, D. Tarapore, S. Thibault, R. A. Watson, W. Weimer, and J. Yosinski. The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artif. Life*, 26(2):274–306, 2020. doi:10.1162/ARTL_A_00319. URL https://doi.org/10.1162/artl_a_00319.

[23] N. Savinov, A. Raichuk, D. Vincent, R. Marinier, M. Pollefeys, T. P. Lillicrap, and S. Gelly. Episodic curiosity through reachability. In *Proc. of the International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019. URL https://openreview.net/forum?id=SkeK3s0qKQ.

[24] K. Wang, K. Zhou, B. Kang, J. Feng, and S. Yan. Revisiting intrinsic reward for exploration in procedurally generated environments. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=j3GK3_xZydY.

[25] Y. Burda, H. Edwards, A. J. Storkey, and O. Klimov. Exploration by random network distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

[26] S. Yan, B. Zhang, Y. Zhang, J. Boedecker, and W. Burgard. Learning continuous control with geometric regularity from robot intrinsic symmetry. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 49–55. IEEE, 2024. doi:10.1109/ICRA57147.2024.10610949. URL https://doi.org/10.1109/ICRA57147.2024.10610949.

[27] C. F. Hayes, R. Radulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A. A. Irissappane, P. Mannion, A. Nowé, G. de Oliveira Ramos, M. Restelli, P. Vamplew, and D. M. Roijers. A practical guide to multi-objective reinforcement learning and planning. *Auton. Agents Multi Agent Syst.*, 36 (1):26, 2022. doi:10.1007/S10458-022-09552-Y. URL https://doi.org/10.1007/s10458-022-09552-y.

[28] L. N. Alegre, A. L. C. Bazzan, D. M. Roijers, A. Nowé, and B. C. da Silva. Sample-efficient multi-objective learning via generalized policy improvement prioritization. In N. Agmon, B. An, A. Ricci, and W. Yeoh, editors, *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, pages 2003–2012. ACM, 2023. doi:10.5555/3545946.3598872. URL https://dl.acm.org/doi/10.5555/3545946.3598872.

[29] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. P. Lillicrap, and D. Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nat.*, 588(7839):604–609, 2020. doi:10.1038/S41586-020-03051-4. URL https://doi.org/10.1038/s41586-020-03051-4.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[31] A. W. Moore. Efficient memory-based learning for robot control. Technical report, University of Cambridge, Computer Laboratory, 1990.

[32] O. Eberhard, J. J. Hollenstein, C. Pinneri, and G. Martius. Pink noise is all you need: Colored noise exploration in deep reinforcement learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=hQ9V5QN27eS.

[33] A. Ecoffet, J. Huizinga, J. Lehman, K. O. Stanley, and J. Clune. First return, then explore. *Nat.*, 590(7847):580–586, 2021. doi:10.1038/S41586-020-03157-9. URL https://doi.org/10.1038/s41586-020-03157-9.

[34] H. van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In D. Schuurmans and M. P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2094–2100. AAAI Press, 2016. doi:10.1609/AAAI.V30I1.10295. URL https://doi.org/10.1609/aaai.v30i1.10295.

[35] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2016. URL http://arxiv.org/abs/1509.02971.

[36] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz. Parameter space noise for exploration. In *Proc. of the International Conference on Learning Representations (ICLR)*. OpenReview.net, 2018. URL https://openreview.net/forum?id=ByBAl2eAZ.

[37] J. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 222–227, 1991.

[38] H. Tang, R. Houthooft, D. Foote, A. Stooke, X. Chen, Y. Duan, J. Schulman, F. D. Turck, and P. Abbeel. #exploration: A study of count-based exploration for deep reinforcement learning. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, pages 2753–2762, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3a20f62a0af1aa152670bab3c602feed-Abstract.html.

[39] D. Jarrett, C. Tallec, F. Altché, T. Mesnard, R. Munos, and M. Valko. Curiosity in hindsight: Intrinsic exploration in stochastic environments. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proc. of the International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 14780–14816. PMLR, 2023. URL https://proceedings.mlr.press/v202/jarrett23a.html.

[40] N. Nikolov, J. Kirschner, F. Berkenkamp, and A. Krause. Information-directed exploration for deep reinforcement learning. In *Proc. of the International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Byx83s09Km.

[41] Y. Guo, J. Choi, M. Moczulski, S. Feng, S. Bengio, M. Norouzi, and H. Lee. Memory based trajectory-conditioned policies for learning from sparse rewards. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/2df45244f09369e16ea3f9117ca45157-Abstract.html.

[42] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826. IEEE Computer Society, 2016. doi:10.1109/CVPR.2016.308. URL https://doi.org/10.1109/CVPR.2016.308.

[43] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi:10.1109/LRA.2023.3270034.

A Algorithm Implementation

We provide the pseudo code for PPO+GAGE with Gaussian policy in Alg. 1.

Algorithm 1 Proximal Policy Optimization (PPO) Algorithm with Gaussian Policy + GAGE

1: Initialize policy mean parameters θ_0, value function parameters ϕ_0, standard deviation σ_0, and goal achievement g_0
2: **for** iteration $k = 0, 1, 2, \ldots$ **do**
3: Collect set of trajectories $\{(s_t, a_t, r_t, s_{t+1})\}$ by running policy $\pi_{\theta_k}(a_t|s_t) = \mathcal{N}(\mu_{\theta_k}(s_t), \sigma_k^2)$ in the environment
4: **for** each time step t **do**
5: Compute advantage estimates $\hat{A}_t$ based on value function $V_{\phi_k}(s_t)$
6: **end for**
7: Update the policy by maximizing the PPO-CLIP objective with an added entropy term:

$$\theta_{k+1}, \sigma_{k+1} = \arg\max_{\theta,\sigma} \mathbb{E}_t \Big[\min \left(\frac{\mathcal{N}(\mu_\theta(s_t), \sigma^2)}{\mathcal{N}(\mu_{\theta_k}(s_t), \sigma_k^2)} \hat{A}_t,\ \text{clip} \left(\frac{\mathcal{N}(\mu_\theta(s_t), \sigma^2)}{\mathcal{N}(\mu_{\theta_k}(s_t), \sigma_k^2)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) + \beta H(\pi_\theta(a_t|s_t)) \Big]$$

where $\mu_{\theta_k}(s_t)$ is the mean of the Gaussian action distribution, σ_k is the standard deviation (separately learned), and $H(\pi_\theta(a_t|s_t))$ is the entropy of the policy, encouraging exploration. The term β controls the weight of the entropy regularization.
8: Update the value function by minimizing the following loss:

$$\phi_{k+1} = \arg\min_{\phi} \mathbb{E}_t \left[\left(V_\phi(s_t) - R_t \right)^2 \right]$$

9: Calculate the running mean of g_k.
10: Update the standard deviation parameter σ based on the agent's performance:

$$\sigma_{k+1} = \max(\sigma_{k+1}, -\sigma_0 g_k + \sigma_0)$$

11: **end for**

B Experimental Details

B.1 Tasks Setup

We build up five challenging continuous control tasks in IsaacLab. Three robots with many degrees of freedom learn to do challenging locomotion or dynamic manipulation behaviors. The robots include a humanoid robot with 21 joints, a dog robot (Unitree Go2) with 12 joints, and an ant robot with 8 joints. The humanoid robot is also employed in the locomotion task to investigate maximum speed and robustness to reward weights. In Table 1, we provide the reward composition of different tasks.

Humanoid tightrope (HT) The humanoid robot learns side walking on a tightrope, i.e., a cylindrical bar with a diameter of only 0.1m. This is more challenging than walking forward because balancing with two arms stretching to both sides would be more difficult.

Humanoid dribbling (HD) The humanoid robot learns to dribble a football at a high speed (3.5m/s). Additionally, the robot gets random commands for turning the target direction for up to $\frac{\pi}{4}\text{rad}$.

https://isaac-sim.github.io/IsaacLab/

Table 1: Reward weights of continuous control tasks. The rewards and penalties from left to right are for robot locomotion velocity, environment not terminating, robot orientation, robot distance to the manipulated object, large action commands, energy consumption, joint position too close to limitations, robot velocity perpendicular to the desired direction, object velocity perpendicular to the desired direction, joint torque, joint acceleration, and action changing rate. The actual target reward for goal achievement calculation is marked in green background.

	reward				**penalty**							
	v_x	alive	orient	d_{obj}	$\|a\|^2$	E	θ_{limit}	v_y	$v_{y,\text{obj}}$	T	$\ddot{\theta}$	$\dot{a}$
HT	0.5	1.0	1.0	0	0	0.05	0.25	1.0	0	0	0	0
HD	0.3	0.4	1.0	0.2	0.01	0.01	0.25	0	0.5	0	0	0
HP	2.0	1.0	1.0	0	0.01	0.005	0.125	0	1.0	0	0	0
DB	1.0	1.0	1.0	0	0.005	0	0	1.0	0	1e-6	2.5e-8	0.001
AA	1.0	1.0	1.0	0	0.005	0.05	0.1	0	1.0	0	0	0

Humanoid pole (HP) The humanoid robot learns to walk forward while balancing a pole vertically on its right hand. The target walking speed is 0.5m/s and the pole is 2m long.

Dog balance beam (DB) The dog robot learns to walk on a balance beam. The beam has a square crosssection with 0.1m side length. Moreover, the balance beam is tilted for $\frac{\pi}{9}\text{rad}$ so that the robot has to climb a slope while balancing.

Ant acrobatics (AA) The ant robot with four legs learns to balance a pole vertically on its torso while standing on a ball. The pole has a length of 2m. The ball has a diameter of the same value. Moreover, the robot has to learn to roll the ball forward at a target speed of 1m/s.

B.2 Hyper-Parameters and Implementation

Hyperparameter for GAGE Since we have not changed the base algorithm implementations, we separately provide the additional hyperparameters introduced by GAGE. There is only one hyperparameter g_0 in the experiments. The results with $\sigma_0 = 0.5, 0.75, 1.0$ are given in Sec. 4.

Hyperparameter for algorithm with continuous action We use Proximal Policy Optimization (PPO) as the backbone algorithm for all the experiments. For the continuous control tasks, we adjust the implementation of rsl_rl v2.0.0. We have not changed any hyperparameters for the implemented algorithms. They are kept the same for all agents for a fair comparison (see Table 1).

https://github.com/leggedrobotics/rsl_rl

Table 2: Hyperparameters used for training agents in continuous control tasks.

Hyperparameter	Value
Algorithm	
Value loss coefficient	1.0
Clip parameter (ϵ)	0.2
Use clipped value loss	True
Desired KL divergence	0.01
Entropy coefficient	0.01
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
Max gradient norm	1.0
Learning rate	0.001
Number of learning epochs	5
Number of mini-batches	4
Learning rate schedule	Adaptive
Policy	
Activation function	ELU
Actor hidden dimensions	[128, 128, 128]
Critic hidden dimensions	[128, 128, 128]
Initial noise standard deviation	1.0
Runner	
Number of steps per environment	24
Max iterations	1500
Empirical normalization	False