Off-policy Reinforcement Learning with Model-based Exploration Augmentation

Likun Wang[1] **Xiangteng Zhang**[1] **Yinuo Wang**[1] **Guojian Zhan**[1]
Wenxuan Wang[1] **Haoyu Gao**[1] **Jingliang Duan**[1,2*] **Shengbo Eben Li**[1*]

[1]School of Vehicle and Mobility & College of AI, Tsinghua University
[2]School of Mechanical Engineering, University of Science and Technology Beijing

Abstract

Exploration is fundamental to reinforcement learning (RL), as it determines how effectively an agent discovers and exploits the underlying structure of its environment to achieve optimal performance. Existing exploration methods generally fall into two categories: active exploration and passive exploration. The former introduces stochasticity into the policy but struggles in high-dimensional environments, while the latter adaptively prioritizes transitions in the replay buffer to enhance exploration, yet remains constrained by limited sample diversity. To address the limitation in passive exploration, we propose **Mo**delic **G**enerative **E**xploration (MoGE), which augments exploration through the generation of under-explored critical states and synthesis of dynamics-consistent experiences through transition models. MoGE is composed of two components: (1) a diffusion-based generator that synthesizes critical states under the guidance of a utility function evaluating each state's potential influence on policy exploration, and (2) a one-step imagination world model for constructing critical transitions based on the critical states for agent learning. Our method adopts a modular formulation that aligns with the principles of off-policy learning, allowing seamless integration with existing algorithms to improve exploration without altering their core structures. Empirical results on OpenAI Gym and DeepMind Control Suite reveal that MoGE effectively bridges exploration and policy learning, leading to remarkable gains in both sample efficiency and performance across complex control tasks.

1 Introduction

Reinforcement learning has demonstrated remarkable potential across various tasks, including autonomous driving, large language models, game playing, and embodied artificial intelligence [68, 11, 65, 49, 19, 66, 67]. It optimizes policies through trial and error, with policy performance fundamentally relying on the diversity and coverage of samples collected during interaction with environments. [18, 69, 21]. Similar to imitation learning (IL), RL algorithms face out-of-distribution (OOD) challenges due to limited diversity in training data. However, RL mitigates this issue by leveraging its exploration capabilities to reach unvisited regions of the state space. In this context, enhancing exploration strategies to collect diverse samples and achieve broader state space coverage becomes crucial for improving the effectiveness and generalization of RL algorithms [25, 3].

Conventional exploration strategies can be broadly classified into two categories: active exploration purely based on policy, and passive exploration based on states. Existing approaches achieve active exploration by introducing randomness into the policy [46, 16] or adding exploration bonuses, which help prevent the policy from converging prematurely to a narrow subset of actions [2]. Specifically,

*Corresponding author <duanjl@ustb.edu.cn> <lishbo@tsinghua.edu.cn>.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

methods like SAC [21] and DSAC [13] leverage maximum entropy as exploration bonuses to encourage exploration, while algorithms such as MPO [1] employ multimodal policies to maintain action diversity and stochasticity. However, the active exploration mechanisms are inherently limited by the agent's actual interacted trajectories, which are constrained by the environment's initial states, finite episode lengths, and the agent's current policy [44]. As a result, many critical regions distant from typical rollouts or rarely visited often remain unexplored, reinforcing value estimation errors and narrowing the scope of policy learning [34, 15]. Furthermore, forcing exploration through the policy can divert focus from reward optimization, leading to suboptimal learning [8].

In contrast to active exploration, which relies on policy-driven interactions, passive exploration modifies the sample distribution of the agent to incorporate prioritized information. Passive exploration originates from Prioritized Experience Replay (PER) [53], which enhances learning by selectively reusing valuable experiences but remains fundamentally limited to previously collected samples. To address this restriction, certain approaches like SER [38] and PGR [64] leverage generative models to augment the replay buffer, thereby artificially intensifying the state-action distribution. However, this expansion is still confined to the vicinity of observed data, adhering closely to the original data distribution. Other methods integrate world models with current policies to simulate state transitions, potentially generating higher-quality transitions [22, 23, 24]; nevertheless, these generated samples typically exhibit limited diversity due to their strong dependence on the policy-conditioned transitions originating from existing states. Moreover, generating complete transitions in high-dimensional state spaces further amplifies the bias introduced by synthetic data, as the complexity grows exponentially with dimensionality [63].

To address the issues above in policy exploration, we propose MoGE, a novel exploration paradigm that enhances off-policy RL algorithms by generating critical transitions across the entire state space, guided by exploratory priors. It consists of two components: a generator that produces critical states with high exploratory potential for the current policy and value network, and a dynamics model that simulates one-step transitions, enabling the generation of critical transitions. Specifically, our work makes three main contributions: **(1)** We employ a conditional diffusion generator to sample critical states with high exploratory potential. To guarantee the state-space compliance and feasibility of the generated states, we theoretically prove that the state distribution in the replay buffer asymptotically converges to the stationary occupancy measure of the optimal policy. By continuously fine-tuning the generator on the replay buffer, we ensure that its learned distribution shares a common support with the optimal policy's occupancy measure, generating critical states in a compliant way. **(2)** To guarantee the dynamical consistency of the generated samples, we design a one-step imagination world model to imitate the dynamics of the environment. This world model allows for efficient pre-training through supervised learning, supporting the construction of training experiences and designing the classifier of the conditional diffusion-based critical state generator. **(3)** We propose an off-policy RL training framework that integrates MoGE seamlessly into existing algorithms without requiring any modifications to their original structure. By introducing importance sampling that mixes critical transitions generated by MoGE with replay buffer samples, MoGE enhances exploration, leading to improved performance and sample efficiency. Experiments on standard continuous control benchmarks, including OpenAI Gym [5] and DeepMind Control Suite [61] demonstrate that MoGE, as a plug-in module, consistently improves both the final performance and the sample efficiency of baseline off-policy RL algorithms.

2 Preliminaries

2.1 Reinforcement Learning and Policy Exploration

In RL, an agent interacts with an environment modeled as a Markov Decision Process (MDP) [60], defined as $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ represent the state and action spaces, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ denotes the environment's transition dynamics, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor [35]. To formally describe the target of RL, we introduce the *occupancy measure* $d^\pi(s)$, which is defined as $d^\pi(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbb{P}(s_t = s \mid \pi)$. This occupancy measure represents the visitation frequency of a given state s under the policy π [32]. Likewise, we can define the $d^\pi(s, a) = d^\pi(s)\pi(a|s)$, which represents the visitation frequency of a given state-action pair (s, a). The target of policy is to maximize the expected cumulative reward, which can be expressed in terms of the occupancy measure as $J(\pi) = \mathbb{E}_{(s,a)\sim d^\pi(s,a)}[r(s, a)]$. Meanwhile, a value function is trained to estimates the value of the current policy by minimizing the temporal difference (TD) error,

which is formulated as $J(\pi) = \mathbb{E}_{(s,a)\sim d^\pi(s,a)}\left[\left(Q^\pi(s,a) - (r + \gamma Q^\pi(s',a'))\right)^2\right]$, where $Q^\pi(s,a)$ denote the value function that evaluate the quality of given (s,a). Through this formulation, the actor optimizes the policy towards higher cumulative rewards, while the critic evaluates and stabilizes the learning process by minimizing prediction errors.

Effective policy training depends on discovering policy-improving states. Since such critical states often require active exploration, identifying them under the current policy enables targeted updates that accelerate learning. If these states can be estimated, they can be selectively replayed for more efficient updates. In existing RL frameworks, two common metrics are used to quantify their criticality:

Policy entropy. The policy entropy reflects the randomness of action selection at a given state. High entropy may indicate either insufficient visitation, offering high information gain, or proximity to critical decision points in the MDP where small action changes lead to divergent outcomes [41, 14]. Focusing learning on such regions enhances policy robustness and long-horizon decision-making. For example, in the case of a Gaussian distribution, the utility function is defined as:

$$f(s) = \mathcal{H}(\pi(\cdot|s)) = \frac{1}{2}\log\big((2\pi e)^d \det \Sigma(s)\big), \tag{1}$$

where $\pi(\cdot|s) = N(\cdot; u(s), \Sigma(s))$ while π and e denote the constant.

TD error. High TD error states are critical for exploration, as they highlight regions where the value function poorly approximates returns under the current policy [31, 15]. These states often correspond to areas of high uncertainty or insufficient optimization. Prioritizing them helps reduce value bias, correct suboptimal actions, and improve policy robustness. The TD error for a given state under the current policy can be estimated as:

$$f(s) = \delta_t(s) \approx r(s, \pi_\theta(s)) + \gamma Q_\psi(d_\phi(s, \pi_\theta(s)), \pi(d_\phi(s, \pi_\theta(s)))) - Q_\psi(s, \pi_\theta(s)). \tag{2}$$

2.2 Diffusion Models for Generative Tasks

Diffusion models have emerged as effective generative models due to their ability to capture complex data distributions [29, 54, 55]. Inspired by non-equilibrium thermodynamics, they simulate two complementary processes: a forward diffusion process incrementally adding noise to the original data, and a reverse denoising process reconstructing the data distribution from noise. In the forward diffusion process, an initial data sample $s_0 \sim q(s_0)$ is gradually transformed into Gaussian noise s_T by iteratively applying:

$$q(s_t|s_{t-1}) = \mathcal{N}(s_t; \sqrt{1-\beta_t}s_{t-1}, \beta_t\mathbf{I}), \tag{3}$$

where β_t controls the noise schedule. Letting $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, the true reverse distribution conditioned on s_0 is given by:

$$q(s_{t-1}|s_t, s_0) = \mathcal{N}\left(s_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(s_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right), \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\mathbf{I}\right). \tag{4}$$

Since s_0 is unknown during generation, diffusion models approximate this posterior with a parameterized neural network $\epsilon_\varphi(s_t, t)$ to predict the noise term ϵ_t. To align with the real reverse process, the approximate reverse process is expressed as:

$$p_\varphi(s_{t-1}|s_t) = \mathcal{N}\left(s_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(s_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\varphi(s_t, t)\right), \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\mathbf{I}\right). \tag{5}$$

The training objective is to minimize the noise prediction error:

$$\mathcal{L}_{\text{generator}} = \mathbb{E}_{s_0,\epsilon,t}\left[\|\epsilon - \epsilon_\varphi(\sqrt{\bar{\alpha}_t}s_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|_2^2\right], \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \tag{6}$$

enabling efficient data generation by progressively denoising from standard Gaussian noise.

3 Methods

In this section, we introduce the MoGE paradigm, illustrated in Figure 1. In Section 3.1, we focus on the method and theoretical foundation of generating critical states through the policy model, as

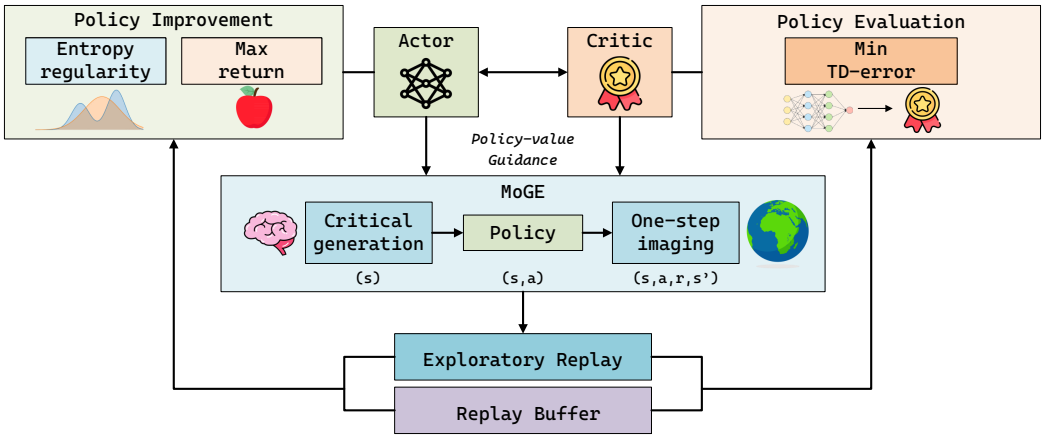


Figure 1: **Overview of MoGE.** MoGE is composed of two sub-modules: a generator and a one-step world model. The generator produces critical states under-explored but potentially valuable for policy exploration under the guidance of policy and value function, while the one-step world model predicts the next state and reward to construct the transitions. The formulated exploratory can be mixed with real samples from the buffer to perform the policy improvement and evaluation.

well as the selection of guidance methods. In Section 3.2, we introduce the structure and application of the one-step imagination world model. In Section 3.3.1, we first analyze the quality of samples generated by MoGE in both novelty and dynamics consistency, and after that, we propose a training framework that integrates MoGE with existing off-policy actor-critic methods in Section 3.3.2.

3.1 Critical State Generation with Steady Occupancy Measurement Alignment

State-based passive exploration leverages prior knowledge to uncover policy-improving states without active search, enabling more directed and sample-efficient policy training. However, when such states are not visited by the policy or stored in the replay buffer, they cannot be utilized for learning. Generative models overcome this limitation by enabling the synthesis of specific states once the model is trained. By leveraging advanced generative techniques such as diffusion and flow matching [37, 17], the model is capable of synthesizing high-fidelity samples that accurately approximate even intractable target distributions.

Motivated by the insight above, MoGE adopts a classifier-guided diffusion model [10], $p_\varphi(s|c)$, which serves as a utility-conditioned generator to synthesize policy-improving states. The unconditional diffusion model learns the manifold of plausible states, while a separately trained classifier provides directional guidance, steering generation toward high-utility regions. This guided synthesis bridges generative modeling and policy optimization, allowing the agent to access critical yet previously unvisited states. In this classifier-guided diffusion framework, the gradient used during inference combines the unconditional diffusion gradient with a classifier-based gradient. Specifically, implementing Bayes' formulation and denoting the classifier c by a continuous *utility function* $f(s)$, the classifier-guided gradient can be expressed as:

$$\nabla \log p_\varphi(s_t|f(s_t)) = \underbrace{\nabla \log p_\varphi(s_t)}_{\textbf{Diffusion gradient}} + \underbrace{\omega \nabla \log p(f(s_t)|s_t)}_{\textbf{Classifier utility gradient}}, \tag{7}$$

where ω is the guidance scale, which controls the strength of this guidance, balancing between following the unconditional diffusion prior and aligning with the target utility function. By selecting an appropriate utility function $f(s)$, this approach allows targeted control over state generation probability, enabling effective exploration of high-value regions.

However, a central challenge in training generative models within RL settings lies in determining an appropriate stationary target distribution. Since the data distribution in RL is inherently policy-dependent and evolves as the policy updates, obtaining a stable and realistic state distribution becomes

infeasible, thereby complicating the training of the generator. As a result, arbitrary or poorly aligned state generation can lead to exploration inefficiencies and instability during policy updates. To address this, the generated states must align closely with an approximately physically grounded and stable occupancy measure, ensuring both state-space compliance and dynamical consistency. In practice, we observe that the behavior policy in the replay buffer exhibits statistically diminishing variation over training, inducing an approximately stationary occupancy measure that can serve as a physically grounded target for the generative model. To formally establish this alignment, we first propose the following theorem as a theoretical guarantee.

Theorem 1 (Steady-State Occupancy Measurement Alignment Theorem)**.** *Let $\beta(a \mid s)$ be a behavior policy, $\pi^*(a \mid s)$ be a specific static policy, and let $\nu_t(s)$ and $d^{\pi^*}(s)$ represent the state occupancy measures under $\beta(a \mid s)$ and $\pi^*(a \mid s)$, respectively. Assuming that the divergence between the policy and $\pi^*(s)$ gradually decreases over the course of training. and that the replay buffer has finite capacity following a First-In-First-Out (FIFO) scheme with states sampled exclusively from the current policy interaction, the following convergence relationship holds:*

$$\lim_{t \to \infty} \mathrm{TV}(\nu_t(s), d^{\pi^*}(s)) = 0. \tag{8}$$

Proof. See Appendix A.2. □

By aligning the unconditional diffusion model's training distribution with the stable occupancy measure in the replay buffer, the critical state generator inherently guarantees generated states remain valid within the true state space since the conditional diffusion shares the same support with the unconditional one [10, 56]. This alignment ensures high-quality, stable critical state generation and enhances the reliability and efficacy of subsequent policy improvements. In this work, we introduce two utility functions below that satisfy these requirements and analyze how each facilitates policy exploration and learning. Since the MoGE is equipped with a transition model, which enables the computation and gradient propagation of certain utility functions that would otherwise be intractable. The two criteria are **policy entropy** and **TD error** (see preliminaries for a detailed introduction). During training, we use different utility signals for different tasks, which facilitates more effective policy learning and accelerates value function convergence.

3.2 Transition Imaging with One-step World Model

When the environment can not return the reward r and next state s' for arbitrary (s, a), the utility function f cannot be directly evaluated without environment interaction. Therefore, we employ learned surrogate models—a reward model $r(s, a)$ and a dynamics model $f(s, a)$—to approximate these quantities and provide differentiable estimates of r and s'. To work in conjunction with the critical state generator, we introduce a one-step imagination world model $\mathcal{F}_\phi$ to estimate the environment dynamics. Since we need to estimate the environment's transition and reward functions for arbitrary (s, a) pairs, we only require the world model to be accurate for one-step dynamics under the current state and policy. Consequently, our world model diverges from conventional designs [22] in both structure and training approach. By focusing on one-step predictions, it trades off long-horizon accuracy to ensure reliable transitions within the region characterized by the optimal policy's occupancy measure.

Our proposed one-step imagination world model $\mathcal{F}_\phi$ consists of five parameterized components to predict the following variables:

$$\begin{aligned}
&\text{Representation:} && \mathbf{z}_t = g_\phi(s_t) \\
&\text{Reconstruction:} && s_t = h_\phi(\mathbf{z}_t) \\
&\text{Latent dynamics:} && \mathbf{z}_{t+1} = d_\phi(\mathbf{z}_t, a_t) \\
&\text{Reward:} && \hat{r}_t = R_\phi(\mathbf{z}_t, a_t) \\
&\text{Termination prediction:} && \hat{c}_t = C_\phi(\mathbf{z}_t, a_t).
\end{aligned} \tag{9}$$

The detailed structure of the world model is depicted in Figure 10 of Appendix C.1. Departing from prior work that relies on probabilistic components [23], we find that implementing all modules of the world model as deterministic networks suffices for effective performance. Except for the latent dynamics model, which uses a two-layer *Transformer* encoder [62] to flexibly handle both sequential and single-step inputs, all other components are implemented as standard MLPs. At each time step t,

Algorithm 1 Off-policy RL training framework with MoGE

Input: Policy π_θ, critical-state generator p_φ, One-step world model $\mathcal{F}_\phi$, critic Q_ψ
Initialize: π_θ, p_φ, $\mathcal{F}_\phi$, Q_ψ

```
// Pretraining
Interact with environment using random actions: (r, s', c) ← env.step(random)
Store random transitions (s, a_rand, r, s') into replay buffer B
Update F_φ, by minimizing model loss L_worldmodel in (10) using random interaction data
Re-warmup the buffer B with the current target π_θ
for each iteration do
    // Data Collection
    Initialize s
    Interact with environment: (r, s', c) ← env.step(π_θ(s))
    Store transition (s, π_θ(s), r, s') into replay buffer B
    // Off-policy RL Learning
    Sample transitions Γ = {(s_t, a_t, r_t, s_{t+1})_{t=0}^T} ~ B
    Update F_φ,p_φ by minimizing model loss L_worldmodel and L_generator in (10),(6) using Γ
    Generate critical states s_e by using classifier-guidance generatorp_φ with utility function f(s)
    Generate full transitions with one-step imaging: (r, s'_e, c) ← F_φ(s_e, π_θ(s_e))
    Combine critical transitions Γ_e = {(s_e, π_θ(s_e), r, s'_e)} with Γ : Γ' = Γ ∪ Γ_e
    Update π_θ, Q_ψ with downstream algorithm using Γ' (for π_θ, do approximate importance
    sampling with λ-mixture )
end for
```

the observation s_t is first encoded into a latent representation $\mathbf{z}_t$ using an encoder network g_ϕ. Given latent state $\mathbf{z}_t$ in feature space and the executed action a_t, the world model outputs: **(1)** a prediction of the next latent state $\mathbf{z}_{t+1}$, **(2)** the corresponding one-step reward $\hat{r}_t$ and **(3)** the termination factor that evaluates whether the transition is terminated. During training, we treat the reward signal and termination factor as augmentations of the next-step state, and design separate loss functions through three output heads. This design simplifies the architecture while preserving the predictive capacity of the world model. The total training loss of the world model is presented as:

$$\begin{aligned}\mathcal{L}_{\text{worldmodel}}(\phi) = \frac{1}{BT}\sum_{n=1}^{B}\sum_{t=1}^{T}\Big[&\underbrace{||\hat{s}_t - s_t||_2}_{\textbf{reconstruction}} + \underbrace{||\hat{r}_t - r_t||_2}_{\textbf{reward}} + \underbrace{c_t \log \hat{c}_t + (1-c_t)\log(1-\hat{c}_t)}_{\textbf{termination}} \\ &+ \underbrace{\beta_1 ||\mathbf{sg}(g_\phi(s_{t+1})) - d_\phi(g_\phi(s_t), a_t)||_2}_{\textbf{dynamics}} \\ &+ \underbrace{\beta_2 ||g_\phi(s_{t+1}) - \mathbf{sg}(d_\phi(g_\phi(s_t), a_t))||_2}_{\textbf{representation}} s\Big],\end{aligned} \tag{10}$$

where $\beta_1 = 0.5, \beta_2 = 0.1$,$(s_t, a_t, r_t, s_{t+1})_{0:T}$ are the transitions sampled from buffer $\mathcal{B}$, and $\mathbf{sg}(\cdot)$ is the stop-gradient operator. T is the total length of the transition that depends on the sample method, and B is the batch size. The model accommodates both sequence and single-step inputs, with $T = 1$ indicates the single-step setting.

3.3 MoGE with Off-policy RL

3.3.1 Sample Quality of MoGE

As a generative exploration framework, MoGE synthesizes novel transitions that extend beyond the replay buffer. In what follows, we examine two key questions that underpin its design and effectiveness:

(1) How do the generated transitions contribute to exploration and policy improvement?

(2) Will the generated transitions influence the policy improvement and evaluation of the algorithm?

(a) Novelty and Policy-Dependent Generation. Let $\mathcal{D}_{\text{replay}}$ denote the replay buffer induced by a historical behavior policy β, corresponding to a state-action distribution $\rho_{\text{replay}}(s, a)$. Conventional passive exploration frameworks [48, 38, 64] intensify transitions from the replay buffer by reordering or imitating, remaining inherently constrained by β. Consequently, the generated transitions $(s, a, s') \sim \rho$ cannot escape the coverage bias of the behavior policy. In contrast, MoGE explicitly constructs a generative distribution

$$\rho_{\text{gen}}(s, a) = p_\varphi(s)\, \pi_\theta(a|s), \tag{11}$$

where $p_\varphi(s)$ denotes a learned state generator guided under a time-varying *utility landscape* $f(s, \pi_\theta(s))$. The critical state distribution is refined as follows:

$$p_\varphi(s) \propto p_\varphi(s)\, \exp\big(\omega\, f(s, \pi_\theta(s))\big), \tag{12}$$

Because f evolves alongside the policy and critic, the target distribution for generation continuously shifts as the policy updates. This property induces **continual novelty**—MoGE adaptively generates new, high-utility regions as the policy changes, rather than remaining tied to the historical behavior distribution. Therefore, MoGE effectively decouples generation from the replay buffer, enabling exploration guided by both the *current policy* and the *estimated value function* instead of fully resampling or imitating past experiences.

(b) Dynamic Consistency and Bellman Validity. While novelty broadens exploration coverage, it must coexist with *dynamical validity* to ensure that generated transitions remain consistent with the environment's transition kernel $p_{\text{env}}(s'|s, a)$. To this end, MoGE employs a learned world model $p_\psi(s'|s, a)$ enforcing the transition-level relation

$$(s, a, s') \sim F_\phi(s'|s, a) \quad \text{such that} \quad s' \approx f_{\text{env}}(s, a), \tag{13}$$

where f_{env} denotes the true environment dynamics. This mechanism ensures that the generated samples adhere to the underlying physical or causal transition relationship.

In contrast, methods that directly synthesize or interpolate transitions (e.g., via diffusion or latent interpolation) without explicit dynamics modeling can easily introduce spurious correlations among (s, a, s') (loss functions that imitate the replay buffer do not ensure that valid transition dependencies are preserved among samples). Such correlations violate the Markov property and lead to inconsistent TD estimates:

$$\mathbb{E}_{(s,a,s')\sim\rho_{\text{syn}}}[r(s, a) + \gamma V(s')] \;\neq\; \mathcal{T}^\pi V(s), \tag{14}$$

where $\mathcal{T}^\pi$ is the Bellman operator. These off-manifold transitions break the Bellman consistency and yield biased policy evaluation. By leveraging the world model F_ϕ, MoGE ensures that generated transitions satisfy the Bellman-consistent transition relation, thus preserving both physical and statistical validity.

(c) Summary. In summary, MoGE achieves *continual novelty* via evolving utility-guided generation that adapts to the current policy, while maintaining *dynamic consistency* through model-based transition regularization. This synergy enables MoGE to explore beyond the behavior policy's support without violating Bellman validity, leading to more reliable and effective policy learning.

3.3.2 Training Off-policy Algorithm with MoGE

To enhance exploration alongside policy improvement and evaluation by introducing the MoGE-generated samples in an off-policy manner, we propose a training framework that integrates MoGE into existing off-policy RL algorithms. Taking Actor-Critic as an example, the training framework is illustrated in Algorithm 1. Notably, due to the existence of the critical generation, a distribution shift arises between the initial state distribution of the buffer and the generator. While this bias is negligible during policy evaluation, it must be addressed during policy improvement through Importance Sampling (IS). Since the distribution of the diffusion model cannot be explicitly represented, the importance sampling ratio for the initial state distribution is intractable. Therefore, we employ a sample mixing method to approximate importance sampling under bounded error. The formulas for policy evaluation and improvement are as follows:

$$\begin{aligned}
\mathcal{L}_{\text{PEV}}(\psi) &= \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}_k}\Big[Q_\psi(s, a) - \big(r + \gamma\, \mathbb{E}_{a'\sim\pi_\theta}[Q_{\bar{\psi}}(s', a')]\big)\Big]^2, \\
\mathcal{L}_{\text{PIM}}(\theta) &= (1-\lambda)\mathbb{E}_{(s,a)\,\sim\,\mathcal{D}_{\text{env}}}\Big[g(s, a)\Big] + \lambda\mathbb{E}_{(s,a)\,\sim\,\mathcal{D}_{\text{gen}}}\Big[g(s, a)\Big],
\end{aligned} \tag{15}$$

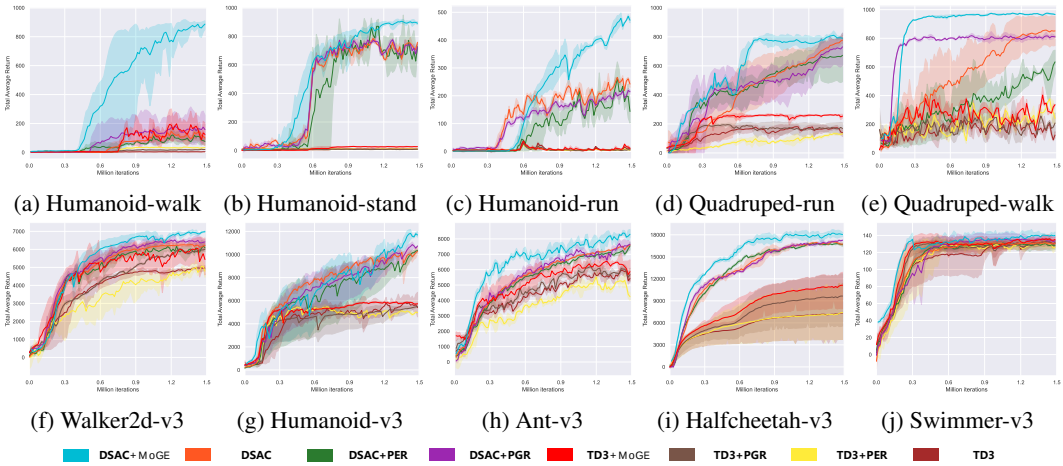


Figure 2: **Training curves on benchmarks.** The solid lines depict the mean performance, while the shaded areas represent the confidence intervals over three seeds. The first row corresponds to the training curves on the DeepMind Control Suite, while the second row represents the results on OpenAI Gym.

where $D_k = (1-k)D_{\text{env}} + kD_{\text{gen}}$, $0 \leq k < 1$, and $g(s,a)$ denotes the return in a single state-action pair (s,a). It is worth noting that the error of this policy improvement method is controllable only when λ is sufficiently small. The details of this approximation method, along with the selection of k and λ, are referred to in the Appendix A.3.

4 Experiments

4.1 Experimental Setup.

Baselines. We choose two widely used representative off-policy RL algorithms as our MoGE baselines: the stochastic policy algorithm DSAC [12, 13] and the deterministic policy algorithm TD3 [16], which achieve active exploration through exploration bonus with policy entropy and random noise injection, respectively. To further evaluate the performance of MoGE, we compare it with passive exploration methods like PGR [64] and prioritized experience replay (PER) [48], where PGR is a state-of-the-art method in data augmentation that directly generates buffer samples using a diffusion model that has been shown to significantly enhance downstream learning.

Benchmarks. We evaluate our method on a diverse benchmark of 10 challenging locomotion tasks drawn from the DeepMind Control Suite (DMC) [61] and the OpenAI Gym [5]. The Gym Benchmark introduces a wide range of control tasks. For example, the Humanoid environment raises the difficulty with high-dimensional state and action spaces (376/17 state/action dims). The DMC tasks feature complex agents: humanoid tasks (67/24 state/action dims) and quadruped tasks (78/12 state/action dims) that demand sophisticated balance and coordination.

Implementation details. To validate the plug-in capability of the MoGE, we preserve the original off-policy algorithms without further fine-tuning. In this paper, the total training step size for all experiments is set at 1.5 million, with the results of all experiments averaged over 3 random seeds. All hyperparameters are aligned with standard implementations, and the configuration details are documented in the Appendix B.

4.2 Experimental Results

All the training curves are shown in Figure 2 and the detailed results are listed in Table 1. Our method, **MoGE** (with DSAC), consistently achieves superior Total Average Return (TAR) across a wide range of locomotion tasks. Despite the challenges introduced by high-dimensional state and action spaces as well as intricate dynamics, it maintains exceptional stability and efficiency, highlighting its robustness and adaptability.

In the challenging DMC Suite tasks, MoGE demonstrates substantial performance enhancements over the original TD3 and DSAC algorithms. MoGE achieves an average Total Average Return (TAR) of **817.7**, significantly outperforming the original DSAC method (568.5) by a notable **+43.8%**. In individual tasks, such as *Humanoid-walk*, MoGE reaches **891.7**, a remarkable **+508.6%** improvement over the original DSAC (146.5). Similarly, TD3 with MoGE significantly surpasses original TD3, delivering a **+73.3%** improvement.

In the OpenAI Gym tasks, MoGE continues to exhibit exceptional performance. MoGE achieves an average TAR of **9135.5**, surpassing the standard DSAC (8301.0) by **+10.0%**. Notably, MoGE sets new benchmark results across all evaluated environments. In *Humanoid-v3*, it attains a score of **12151.1**, a substantial **+16.8%** increase over DSAC (10402.2). MoGE's consistent performance highlights its superior effectiveness, offering clear and significant improvements in both early-stage learning and final asymptotic returns compared to traditional TD3 and DSAC implementations.

Table 1: Total Average Return (TAR) on 5 DMC Suite tasks and 5 OpenAI Gym tasks. Mean ± Std over 3 seeds. **Bold** = best; Higher is better.

Environment	TD3	TD3+PER	TD3+PGR	TD3+MoGE	DSAC	DSAC+PER	DSAC+PGR	DSAC+MoGE
Humanoid-walk	120.1 ± 106.8	34.4 ± 1.2	22.5 ± 2.4	222.4 ± 72.0	146.5 ± 60.9	122.5 ± 48.5	195.0 ± 92.3	**891.7 ± 19.1**
Humanoid-stand	10.6 ± 0.2	11.5 ± 0.9	8.3 ± 0.2	28.4 ± 0.7	776.6 ± 15.6	816.5 ± 94.5	754.4 ± 16.2	**907.5 ± 6.9**
Humanoid-run	15.9 ± 1.8	8.7 ± 2.8	7.0 ± 0.4	25.2 ± 2.5	267.4 ± 3.9	271.1 ± 35.5	223.4 ± 3.3	**488.9 ± 8.7**
Quadruped-run	84.1 ± 5.4	63.3 ± 2.5	83.2 ± 16.6	128.2 ± 0.4	793.9 ± 29.9	662.7 ± 162.6	717.9 ± 107.4	**824.3 ± 19.0**
Quadruped-walk	236.5 ± 19.5	380.2 ± 7.2	290.2 ± 11.7	405.2 ± 54.7	857.9 ± 102.8	649.3 ± 67.7	823.2 ± 17.5	**976.2 ± 3.1**
AVG.DMC	93.4 ± 26.7	99.62 ± 2.9	82.2 ± 6.3	161.9 ± 26.1	568.5±42.7	504.4 ± 81.8	542.8 ± 47.3	**817.7 ± 11.4**
Walker2d-v3	5031.1 ± 84.2	5253.7 ± 206.1	6007.5 ± 4.9	6082.8 ± 606.7	6288.3 ± 83.3	6391.6 ± 246.9	6501.1 ± 87.3	**6978.4 ± 68.7**
Humanoid-v3	5967.1 ± 547.8	5203.9 ± 33.9	5531.8 ± 62.3	5885.8 ± 38.3	10402.2 ± 187.7	10363.6 ± 109.0	11004.0 ± 121.5	**12151.1 ± 35.4**
Ant-v3	6037.5 ± 119.2	6134.7 ± 212.5	5961.3 ± 170.7	6369.1 ± 265.7	7610.0 ± 10.0	7637.0 ± 27.1	7837.2 ± 203.0	**8176.6 ± 44.9**
Halfcheetah-v3	7363.0 ± 3666.4	7438.1 ± 3774.3	9687.1 ± 4099.0	11167.3 ± 2234.8	17072.0 ± 61.3	16913.0 ± 70.2	17324.7 ± 41.1	**18054.9 ± 459.6**
Swimmer-v3	133.5 ± 5.3	131.8 ± 3.3	134.1 ± 2.8	137.1 ± 2.8	132.3 ± 5.1	131.0 ± 6.6	136.0 ± 3.0	**141.3 ± 2.0**
AVG.Gym	4906.4 ± 884.6	4832.4 ± 846.0	5464.4 ± 867.9	5928.4 ± 629.7	8301.0 ± 69.5	8287.2 ± 92.0	8535.6 ± 91.2	**9135.5 ± 122.1**

4.3 Ablation

We perform three ablation studies to evaluate the impact of each core component in our framework:

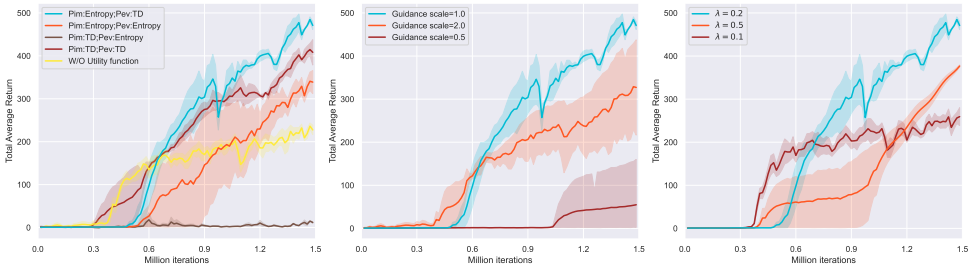
Utility function for exploration. We compare the choice of utility function for different parts of updating, as illustrated in Figure 3a. Compared to policy entropy, TD error is more beneficial during policy evaluation. On the other hand, entropy plays a more significant role in policy improvement since high-entropy regions encourage broader exploration, while high TD-error regions may stem from inaccurate environment estimation, potentially leading to unreliable evaluations.

Guidance scale ω. We chose different guidance intensities to test the balance between the regions with high potential for exploration and feasibility guarantee, as shown in Figure 3b. When the guidance strength is set to 1, it effectively balances the generation of states with high exploratory value and feasibility, ensuring that the diffusion-generated states maintain alignment with the optimal occupancy measure while maximizing exploration potential.

Mix ratio λ. We vary the value of λ by testing from 0.1 to 0.5. Results in Figure 3c show that $\lambda = 0.2$ stable performance across this range. When λ is too small, the policy fails to acquire enough critical states for effective exploration. Conversely, if λ is too large, the discrepancy in the state distribution becomes non-negligible, which aligns with the discussion in Appendix A.3.

5 Related Works

Active exploration. Active exploration methods explicitly modify policies to enhance exploration, which originates from the epsilon-greedy policy [60, 39, 46, 16]. Entropy-based strategies like SAC [21], DSAC [13], DACER [67, 66], and PPO [51] incorporate entropy terms to balance exploration and exploitation, preventing premature convergence and enhancing stability. More advanced exploration approaches explicitly encourage policy diversity. Count-based methods, such as pseudo-counts



(a) Ablation study on the choice of utility function

(b) Ablation study on the guidance scale

(c) Ablation study on the mix ratio

Figure 3: **Ablation study curves.** We select the *Humanoid-run* task in DMC Suite with high complexity to perform all ablation experiments.

[4] and neural density models [43], estimate state visitation frequencies to incentivize exploration in under-sampled regions. Intrinsic motivation strategies, including RND [6] and ICM [45], generate exploration bonuses from novelty signals or prediction errors. Bootstrap DQN [42] further introduces uncertainty-aware exploration by leveraging ensemble networks to identify poorly understood state-action pairs. Recent advances maintain exploration diversity through maximum entropy principles, and exploration-driven policy optimization [1, 45], which directly augments policies for broader state coverage. These methods, while easy to implement, often require complex parameter tuning and may struggle with scalability in high-dimensional spaces compared with MoGE.

Passive exploration. The policy update process not only depends on its optimization objective, but also on the samples collected for estimating the gradient. By shifting the distribution of replayed samples, the policy is able to cover a wider range of states. Prioritized experience replay (PER) [48] dynamically adjusts the replaying frequency of collected samples by TD-error. Some further research [47, 57] uses different metrics for experience replay. Rather than only selecting the samples, data generation is more flexible since it not only models the initial state but also captures its transition. Diffuser [30] utilizes a diffusion probabilistic model that plans by iteratively denoising trajectories. Synthetic experience replay (SER) [38] proposes a diffusion-based approach to flexibly upsample an agent's collected experience. Prioritized generative replay (PGR) [64] generates learning-informative transitions under a given relevance function. Another paradigm of passive exploration is model-based RL [20]. These methods learns a parameterized transition model of state and action and directly optimize the policy through imagined samples, such as Dreamer [22, 23, 24] and other variants [7, 27, 70, 26]. Our proposed MoGE method can generate critical initial states and their transitions, thus guaranteeing data compliance.

6 Conclusion

We proposed MoGE, a novel exploration paradigm that addresses the limitations of passive exploration in off-policy RL. It enhances exploration by generating critical states guided by exploratory priors and then estimates state transitions through a world model, forming valid training samples that guarantee state-space compliance and dynamical feasibility. Experimental results on OpenAI Gym and DeepMind Control Suite benchmarks demonstrate that MoGE significantly improves sample efficiency and overall performance, validating the effectiveness of this exploration paradigm. We believe MoGE establishes a new perspective for exploration augmentation in reinforcement learning, with significant potential for future improvements, such as incorporating adaptive mechanisms for prioritizing critical state transitions or leveraging more expressive generative models beyond conditional diffusion to further boost exploration efficiency and policy robustness.

7 Acknowledgment

This research was generously supported by the Beijing Natural Science Foundation (L257002). The core of this work was completed in the Intelligence Driving Lab (IDLAB) at Tsinghua University. We extend our sincere gratitude to all the lab members for their constructive feedback during this period.

References

[1] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

[2] Zafarali Ahmed, Nicolas Le Roux, Mohammad Norouzi, and Dale Schuurmans. Understanding the impact of entropy on policy optimization. In *International conference on machine learning*, pages 151–160. PMLR, 2019.

[3] Chayan Banerjee, Zhiyong Chen, and Nasimul Noman. Boosting exploration in actor-critic algorithms by incentivizing plausible novel states. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 7009–7014. IEEE, 2023.

[4] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[6] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[7] Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

[8] Eric Chen, Zhang-Wei Hong, Joni Pajarinen, and Pulkit Agrawal. Redeeming intrinsic rewards via constrained optimization. *Advances in neural information processing systems*, 35:4996–5008, 2022.

[9] Xinyue Chen, Che Wang, Zijian Zhou, and Keith Ross. Randomized ensembled double q-learning: Learning fast without a model. *arXiv preprint arXiv:2101.05982*, 2021.

[10] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[11] Oguzhan Dogru, Junyao Xie, Om Prakash, Ranjith Chiplunkar, Jansen Soesanto, Hongtian Chen, Kirubakaran Velswamy, Fadi Ibrahim, and Biao Huang. Reinforcement learning in process industries: Review and perspective. *IEEE/CAA Journal of Automatica Sinica*, 11(2):283–300, 2024.

[12] Jingliang Duan, Yang Guan, Shengbo Eben Li, Yangang Ren, Qi Sun, and Bo Cheng. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE transactions on neural networks and learning systems*, 33(11):6584–6598, 2021.

[13] Jingliang Duan, Wenxuan Wang, Liming Xiao, Jiaxin Gao, Shengbo Eben Li, Chang Liu, Ya-Qin Zhang, Bo Cheng, and Keqiang Li. Distributional soft actor-critic with three refinements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[14] Benjamin Eysenbach and Sergey Levine. If maxent rl is the answer, what is the question? *arXiv preprint arXiv:1910.01913*, 2019.

[15] Sebastian Flennerhag, Jane X Wang, Pablo Sprechmann, Francesco Visin, Alexandre Galashov, Steven Kapturowski, Diana L Borsa, Nicolas Heess, Andre Barreto, and Razvan Pascanu. Temporal difference uncertainties as a signal for exploration. *arXiv preprint arXiv:2010.02255*, 2020.

[16] 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.

[17] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

[18] Ivo Grondman, Lucian Busoniu, Gabriel AD Lopes, and Robert Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, part C (applications and reviews)*, 42(6):1291–1307, 2012.

[19] Agrim Gupta, Silvio Savarese, Surya Ganguli, and Li Fei-Fei. Embodied intelligence via learning and evolution. *Nature communications*, 12(1):5721, 2021.

[20] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[21] 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.

[22] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[23] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[24] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[25] Seungyul Han and Youngchul Sung. Diversity actor-critic: Sample-aware entropy regularization for sample-efficient exploration. In *International Conference on Machine Learning*, pages 4018–4029. PMLR, 2021.

[26] Nicklas Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. *arXiv preprint arXiv:2310.16828*, 2023.

[27] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

[28] Takuya Hiraoka, Takahisa Imagawa, Taisei Hashimoto, Takashi Onishi, and Yoshimasa Tsuruoka. Dropout q-functions for doubly efficient reinforcement learning. *arXiv preprint arXiv:2110.02034*, 2021.

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[30] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[31] David Janz, Jiri Hron, Przemysław Mazur, Katja Hofmann, José Miguel Hernández-Lobato, and Sebastian Tschiatschek. Successor uncertainties: exploration and uncertainty in temporal difference learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[32] Romain Laroche and Remi Tachet Des Combes. On the occupancy measure of non-markovian policies in continuous mdps. In *International Conference on Machine Learning*, pages 18548–18562. PMLR, 2023.

[33] Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. Simba: Simplicity bias for scaling up parameters in deep reinforcement learning. *arXiv preprint arXiv:2410.09754*, 2024.

[34] Donghao Li, Ruiquan Huang, Cong Shen, and Jing Yang. Near-optimal conservative exploration in reinforcement learning under episode-wise constraints. In *International Conference on Machine Learning*, pages 19527–19564. PMLR, 2023.

[35] Shengbo Eben Li. Reinforcement learning for sequential decision and optimal control. 2023.

[36] 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.

[37] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[38] Cong Lu, Philip Ball, Yee Whye Teh, and Jack Parker-Holder. Synthetic experience replay. *Advances in Neural Information Processing Systems*, 36:46323–46344, 2023.

[39] 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.

[40] Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Miłoś, and Marek Cygan. Bigger, regularized, optimistic: scaling for compute and sample efficient continuous control. *Advances in neural information processing systems*, 37:113038–113071, 2024.

[41] Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

[42] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

[43] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR, 2017.

[44] Pathmanathan Pankayaraj and Pradeep Varakantham. Constrained reinforcement learning in hard exploration problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 15055–15063, 2023.

[45] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[46] Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*, 2017.

[47] Mirza Ramicic and Andrea Bonarini. Entropy-based prioritized sampling in deep q-learning. In *2017 2nd international conference on image, vision and computing (ICIVC)*, pages 1068–1072. IEEE, 2017.

[48] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[49] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[50] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[51] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[52] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *International conference on machine learning*, pages 8583–8592. PMLR, 2020.

[53] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

[54] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[55] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[57] Shivakanth Sujit, Somjit Nath, Pedro Braga, and Samira Ebrahimi Kahou. Prioritizing samples in reinforcement learning with reducible loss. *Advances in Neural Information Processing Systems*, 36:23237–23258, 2023.

[58] Bhavya Sukhija, Stelian Coros, Andreas Krause, Pieter Abbeel, and Carmelo Sferrazza. Maxinforl: Boosting exploration in reinforcement learning through information gain maximization. *arXiv preprint arXiv:2412.12098*, 2024.

[59] Bhavya Sukhija, Lenart Treven, Carmelo Sferrazza, Florian Dorfler, Pieter Abbeel, and Andreas Krause. Optimism via intrinsic rewards: Scalable and principled exploration for model-based reinforcement learning. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2025.

[60] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[61] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[62] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[63] Kevin Wang, Hongqian Niu, Yixin Wang, and Didong Li. Deep generative models: Complexity, dimensionality, and approximation. *arXiv preprint arXiv:2504.00820*, 2025.

[64] Renhao Wang, Kevin Frans, Pieter Abbeel, Sergey Levine, and Alexei A Efros. Prioritized generative replay. *arXiv preprint arXiv:2410.18082*, 2024.

[65] Shuhe Wang, Shengyu Zhang, Jie Zhang, Runyi Hu, Xiaoya Li, Tianwei Zhang, Jiwei Li, Fei Wu, Guoyin Wang, and Eduard Hovy. Reinforcement learning enhanced llms: A survey. *arXiv preprint arXiv:2412.10400*, 2024.

[66] Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, et al. Diffusion actor-critic with entropy regulator. *Advances in Neural Information Processing Systems*, 37:54183–54204, 2024.

[67] Yinuo Wang, Likun Wang, Mining Tan, Wenjun Zou, Xujie Song, Wenxuan Wang, Tong Liu, Guojian Zhan, Tianze Zhu, Shiqi Liu, et al. Enhanced dacer algorithm with high diffusion efficiency. *arXiv preprint arXiv:2505.23426*, 2025.

[68] Guojian Zhan, Yuxuan Jiang, Shengbo Eben Li, Yao Lyu, Xiangteng Zhang, and Yuming Yin. A transformation-aggregation framework for state representation of autonomous driving systems. *IEEE Transactions on Intelligent Transportation Systems*, 25(7):7311–7322, 2024.

[69] Guojian Zhan, Xiangteng Zhang, Feihong Zhang, Letian Tao, and Shengbo Eben Li. Bicriteria policy optimization for high-accuracy reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[70] Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. Storm: Efficient stochastic transformer based world models for reinforcement learning. *Advances in Neural Information Processing Systems*, 36:27147–27166, 2023.

A Theoretical Analysis

A.1 Useful lemmas

Lemma 1 (Discounted-Occupancy Lipschitz Lemma)**.** *Let $0 < \gamma < 1$. For any two stochastic policies π and π' sharing the same initial state distribution μ, their discounted state–occupancy measurements $d_\mu^\pi(s)$ and $d_\mu^{\pi'}(s)$ meet the following condition:*

$$\left\|d_\mu^\pi - d_\mu^{\pi'}\right\|_{\mathrm{TV}} \;\le\; \frac{\gamma}{1-\gamma}\,\sup_{s\in\mathcal{S}} \mathrm{TV}\big(\pi(\cdot \mid s), \pi'(\cdot \mid s)\big), \tag{16}$$

where $TV(\cdot,\cdot)$ denotes the total variation of two distributions:

$$TV(a(s), b(s)) = \frac{1}{2}\sum_s |a(s) - b(s)|. \tag{17}$$

Proof. By the definition, the discounted occupancy measure can be reformulated as a fixed-point equation:

$$d_\mu^\pi = (1-\gamma)\mu + \gamma\, d_\mu^\pi P_\pi, \quad d_\mu^{\pi'} = (1-\gamma)\mu + \gamma\, d_\mu^{\pi'} P_{\pi'}, \tag{18}$$

where $P_\pi = \sum_a \pi(a \mid s) P(s' \mid s, a)$ is the one-step transition kernel of π. It denotes the transition probability from state s to s' under the policy π. Defining the subtraction of the two transition kernels as Δ, the following equation holds:

$$\Delta := d_\mu^\pi - d_\mu^{\pi'} = \gamma\, \Delta P_\pi + \gamma\, d_\mu^{\pi'}(P_\pi - P_{\pi'}). \tag{19}$$

By rearranging Δ to the left-hand side and factoring out the common terms, we obtain:

$$\Delta(I - \gamma P_\pi) = \gamma d_\mu^{\pi'}(P_\pi - P_{\pi'}). \tag{20}$$

Because the induced norm of the transition kernel $\|P_\pi\|_1 = 1$ and $\gamma < 1$, $(I - \gamma P_\pi)^{-1}$ exists and $(I - \gamma P_\pi)^{-1} = \sum_{k\ge 0}(\gamma P_\pi)^k$ converges. Multiplying by this inverse, we can derive the Neumann series as follows:

$$\Delta = \gamma \sum_{k=0}^{\infty} (\gamma P_\pi)^k\, d_\mu^{\pi'}(P_\pi - P_{\pi'}). \tag{21}$$

The L_1 norm of the above equation is given by:

$$\|\Delta\|_1 \le \gamma\Big(\sum_{k=0}^{\infty}\gamma^k\Big)\|d_\mu^{\pi'}\|_1 \|P_\pi - P_{\pi'}\|_1. \tag{22}$$

Due to the definition of the transition kernel, we can derive that:

$$\begin{aligned} \|P_\pi - P_{\pi'}\|_{1\to 1} &= \sup_{s\in\mathcal{S}} \left\| \sum_{a\in\mathcal{A}} [\pi(a \mid s) - \pi'(a \mid s)] P(\cdot \mid s, a) \right\|_1 \\ &\le \sum_{a\in\mathcal{A}} |\pi(a \mid s) - \pi'(a \mid s)| \cdot ||1|| \\ &= 2\,\mathrm{TV}(\pi(\cdot \mid s), \pi'(\cdot \mid s)\big) \end{aligned} \tag{23}$$

Since $\|d_\mu^{\pi'}\|_1 = 1$, We can obtain that:

$$\|\Delta\|_1 \le \frac{2\gamma}{1-\gamma}\sup_s \mathrm{TV}(\pi(\cdot \mid s), \pi'(\cdot \mid s)\big). \tag{24}$$

Dividing by 2 converts the L_1 norm to total variation, the proof is completed.

For clarity, we provide the proof in the discrete state-action space. Since we assume the continuous state-action space is a measurable continuous space, the contraction property in the total variation norm and the convergence of the Neumann series of $(I - \gamma P)$ also hold in this setting. Therefore, the conclusion can be directly extended to the continuous state-action space. □

Lemma 2 (FIFO-Buffer Proximity Lemma)**.** *Assume the replay buffer $\mathcal{B}$ holds exactly the most recent K transitions (*FIFO *of fixed size K). We can define the occupancy measurement under the behavior policy in $\mathcal{B}$ as: $\nu_t = \frac{1}{K}\sum_{h=0}^{K-1} d^{\pi_{t-h}}$, and let $\Delta_\tau := \sup_{s\in\mathcal{S}} \mathrm{TV}\big(\pi_{\tau+1}(\cdot \mid s), \pi_\tau(\cdot \mid s)\big)$. Then the following inequality holds:*

$$\big\|\nu_t - d^{\pi_t}\big\|_{\mathrm{TV}} \;\le\; \frac{\gamma}{1-\gamma} \sum_{j=0}^{K-1} \Delta_{t-j}. \tag{25}$$

Proof. By the convexity of total variation, we can obtain:

$$\|\nu_t - d^{\pi_t}\|_{\mathrm{TV}} \le \frac{1}{K}\sum_{h=0}^{K-1} \|d^{\pi_{t-h}} - d^{\pi_t}\|_{\mathrm{TV}}. \tag{26}$$

By using the triangle inequality in, we can obtain that $\|d^{\pi_{t-h}} - d^{\pi_t}\|_{\mathrm{TV}} \le \tau_{\text{mix}} \sum_{j=0}^{h-1} \Delta_{t-j}$ with $\tau_{\text{mix}} = \gamma/(1-\gamma)$. Insert this bound and interchange the sums, we can derive that:

$$\begin{aligned} \|\nu_t - d^{\pi_t}\|_{\mathrm{TV}} &\le \frac{\tau_{\text{mix}}}{K} \sum_{h=0}^{K-1}\sum_{j=0}^{h-1} \Delta_{t-j} \\ &= \tau_{\text{mix}} \sum_{j=0}^{K-1} \frac{K-j}{K}\,\Delta_{t-j} \;\le\; \tau_{\text{mix}} \sum_{j=0}^{K-1} \Delta_{t-j}. \end{aligned} \tag{27}$$

□

Lemma 3 (Behaviour-Mixing Contraction Lemma)**.** *Assuming that each policy π_{t-h} (for $h \in [0, K-1]$) writes the same number of transition samples into the buffer during the last K time steps, and the difference between consecutive policies remains at a negligible scale: $\sup_s \mathrm{TV}(\pi_{t+1}, \pi_t)(s) \ll 1$, the behavior policy represented in the buffer can be approximately expressed as $\beta_t \approx \frac{1}{K}\sum_{h=0}^{K-1} \pi_{t-h}$ Let $\delta_\tau := \sup_{s\in\mathcal{S}} \mathrm{TV}\big(\pi_\tau(\cdot \mid s), \pi^*(\cdot \mid s)\big)$, then the following bound holds:*

$$\sup_{s\in\mathcal{S}} \mathrm{TV}\big(\beta_t(\cdot \mid s), \pi^*(\cdot \mid s)\big) \;\le\; \frac{1}{K}\sum_{h=0}^{K-1} \delta_{t-h}. \tag{28}$$

Proof. Fix $s \in \mathcal{S}$, since the total variation is convex in its first argument, we can derive that:

$$\mathrm{TV}\Big(\frac{1}{K}\sum_{h=0}^{K-1} \pi_{t-h}(\cdot \mid s), \pi^*(\cdot \mid s)\Big) \le \frac{1}{K}\sum_{h=0}^{K-1} \mathrm{TV}\big(\pi_{t-h}(\cdot \mid s), \pi^*(\cdot \mid s)\big). \tag{29}$$

Taking the supremum over s and moving the sup outside the sum gives the claimed inequality. □

Lemma 4 (The λ–mixture estimator bias)**.** *Let $\rho(s,a) = d^{\pi}_{\textbf{\textit{env}}}(s,a)$ be the target state–action distribution and $\rho'(s,a) = d^{\beta}_{gen}(s,a)$ be the behaviour distribution. For any measurable function $g : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ bounded by $\|g\|_\infty$, define the λ–mixture sampling measure as $q_\lambda = (1-\lambda)\rho + \lambda\rho', 0 \le \lambda \le 1$. Denote the exact importance sampling (IS) estimator $\hat{J}_{IS} = \mathbb{E}_{(s,a)\sim q_\lambda}\big[\,w(s,a)\,g(s,a)\big]$ with weight $w = \rho/q_\lambda$, and the λ–*mixture estimator $\hat{J}_{mix} = \mathbb{E}_{(s,a)\sim q_\lambda}\big[g(s,a)\big]$. Then the following inequality holds:*

$$\big|\,\mathbb{E}\big[\hat{J}_{mix}\big] - \mathbb{E}\big[\hat{J}_{IS}\big]\big| \le \lambda\, M\, \sqrt{2\,\mathrm{KL}\big(\rho' \,\|\, \rho\big)} = \mathcal{O}\big(\lambda\,\mathrm{KL}(\pi_\theta\|\beta)^{1/2}\big). \tag{30}$$

Proof. Rewrite the bias as follows:

$$\hat{J}_{\text{IS}} - \hat{J}_{\text{mix}} = \lambda\,\mathbb{E}_{\rho'}\big[\,\big(\tfrac{\rho}{\rho'} - 1\big)\,g\,\big]. \tag{31}$$

By applying Hölder's inequality, we can obtain that:

$$\|\rho - \rho'\|_1 \le \sqrt{2\,\mathrm{KL}(\rho'\|\rho)}. \tag{32}$$

Because $|g| \le M$, the magnitude of the bias is upper-bounded by $\lambda\, M\sqrt{2\,\mathrm{KL}(\rho'\|\rho)}$. Replacing the distribution notation with the corresponding policies gives the stated $\mathcal{O}(\lambda\,\mathrm{KL})$ dependence. □

A.2 Proof of Theorem 1

Proof. From lemma 3, the upper bound of total variation between the behavior policy β_t and the optimal policy π^* can be derived as:

$$\sup_s \mathrm{TV}(\beta_t, \pi^*) \leq \frac{1}{K} \sum_{h=0}^{K-1} \delta_{t-h}. \tag{33}$$

If the target policy converges to the optimal policy, i.e., $\delta_{t-h} \to 0$ holds, then the following relation is satisfied:

$$\lim_{t\to\infty} \sup_s \mathrm{TV}(\beta_t, \pi^*) = 0. \tag{34}$$

The above expression indicates that the behavior policy β_t will gradually converge to the optimal policy π^*. Then, using the triangle inequality, we can obtain the following:

$$\mathrm{TV}(\nu_t, d^{\pi^*}) \leq \underbrace{\mathrm{TV}(\nu_t, d^{\pi_t})}_{A_t} + \underbrace{\mathrm{TV}(d^{\pi_t}, d^{\pi^*})}_{B_t}. \tag{35}$$

From lemma 2, the first part A_t can be derived as :

$$A_t \leq \tau_{\min} \sum_{j=0}^{K-1} \Delta_{t-j}. \tag{36}$$

Since the policy can converge to the optimal policy within a finite number of steps N, the final policy shift error is considered to vanish after these steps. Therefore, the term A_t can be regarded as zero. From lemma 1, the B_t can be reformulated as:

$$B_t = \left\| d^{\pi_t} - d^{\pi^*} \right\|_{\mathrm{TV}} \leq \tau_{\mathrm{mix}} \sup_s \mathrm{TV}\left(\pi_t(\cdot|s), \pi^{\cdot^*}(\cdot|s)\right) = \tau_{\mathrm{mix}} \delta_t. \tag{37}$$

Likewise, due to policy convergence in finite steps, $\delta_t = 0$ holds when $t > N$. Thus, the proof is complete. □

Notably, if the policy does not converge to the optimal policy within a finite number of steps, but the policy shift Δ_t decays sufficiently fast after each update, where the following condition holds:

$$\sum_{t=0}^{\infty} \Delta_t = \sum_{t=0}^{\infty} \sup_{s\in S} \mathrm{TV}(\pi_{t+1}(\cdot \mid s), \pi_t(\cdot \mid s)) < \infty. \tag{38}$$

Under this condition, it can still be guaranteed that $A_t = 0$, and the original proof remains valid.

A.3 Discussion of importance sampling in MoGE

In MoGE, since the critical experiences are generated by the diffusion-based generator and the one-step imagination world model, both the initial-state distribution and the policy of the experiences may differ from those in replay buffer $\mathcal{B}$ under the assumption that the one-step world model can accurately estimate the environmental dynamics, where all transitions share the same dynamics kernel $P(s' \mid s, a)$.

Problem Setting. We consider training samples composed of two distributions: **(1)**$\mathcal{D}_{\text{env}}$ collected with the behavior policy β and the real initial state distribution $d_{\text{env}}(s)$ from the buffer $\mathcal{B}$; **(2)** $\mathcal{D}_{\text{gen}}$ generated synthetically with target policy π_θ by MoGE, where the initial state distribution is denoted as $d_{\text{gen}}(s)$.

For policy evaluation, let Q^π be the action-value function of the target policy π_θ. It obeys the Bellman identity

$$Q^\pi(s,a) \;=\; \mathbb{E}_{s'\sim P,\, a'\sim\pi_\theta}\big[r(s,a) + \gamma\, Q^\pi(s',a')\big], \qquad \forall (s,a). \tag{39}$$

Because Eq. (39) holds in a point-wise way, where the (s,a) can be sampled from anonymous distribution, the squared TD–error of any parameterised critic Q_ψ,

$$\mathcal{L}_{\text{PEV}}(\psi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}_k}\Big[Q_\psi(s,a) - \big(r + \gamma\, \mathbb{E}_{a'\sim\pi_\theta}[Q_{\bar{\psi}}(s',a')]\big)\Big]^2, \tag{40}$$

still attains its global minimum at $Q_\phi = Q^\pi$ for any mixture of the training $\mathcal{D}_k = (1-k)\mathcal{D}_{\text{env}} + k\mathcal{D}_{\text{gen}}, 0 \le k \le 1$. Hence, the critic remains unbiased without importance sampling; off-policy sampling only affects optimization variance. An IS correction becomes necessary only when regressing the full Monte-Carlo return G_0, which is behaviour-dependent. However, for policy improvement, the importance sampling is non-negligible since the existence of a distribution mismatch:

$$\mathcal{L}_{\text{PIM}}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{env}}}\Big[\rho(s,a)\ g(s,a)\Big] + \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{gen}}}\Big[w(s)\ g(s,a)\Big] \tag{41}$$

where $g(s,a)$ denote the return differs in different algorithm, and $w(s) = \frac{d_{\text{env}}(s)}{d_{\text{gen}}(s)}, \rho(s,a) = \frac{\pi_\theta(a|s)}{\beta(a|s)}$. However, for algorithms that compute the objective function directly based on the target, i.e., $g(s,a) = g(s,\pi_\theta(s))$, importance sampling for the policy can be omitted. In this case, policy improvement only requires importance sampling for the initial state distribution. Since computing $w(s)$ and is intractable in high-dimensional continuous spaces. We therefore approximate the IS expectation by a sampling mixture using a mixing rate λ since the bias can be bound by Lemma 4. The target for policy improvement in MoGE can be finally derived as a *sampling mixture*:

$$\mathcal{L}_{\text{PIM}}(\theta) = (1-\lambda)\mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{env}}}\Big[g(s,a)\Big] + \lambda\mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{gen}}}\Big[g(s,a)\Big] \tag{42}$$

with $\lambda \in [0,1)$. For example, the resulting actor loss for SAC-style objectives is like:

$$\begin{aligned}\mathcal{L}_{\text{PIM}}(\theta) = (1-\lambda)\,&\mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{env}}}\big[\alpha\log\pi_\theta(a\mid s) - Q_\psi(s,a)\big] \\ &+ \lambda\,\mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{beh}}}\big[\alpha\log\pi_\theta(a\mid s) - Q_\psi(s,a)\big].\end{aligned} \tag{43}$$

Note that Eq. 43 is the first-order Taylor expansion of the exact IS estimator; its bias scales with $\mathcal{O}\big(\lambda\,\mathrm{KL}(\pi_\theta\|\beta)^{1/2}\big)$. Empirically, a small λ retains the coverage benefit of the generated data while keeping bias and training instability negligible.

Theoretically, the choice of k is unrestricted; however, to reduce training variance, it is advisable to constrain k within a smaller range. In this work, we set $k = 2\lambda$.

B Environmental configuration

B.1 Environment Introduction

DeepMind Control Suite. We chose 5 challenging tasks involving the humanoid and quadruped robots. The final reward for each task is the product of the standing reward and the forward velocity reward, expressed as: ***Reward*** $=$ (***Standing Reward***) $\times$ (***Forward Velocity Reward***).

(a) Humanoid (b) Quadruped

Figure 4: DMC environments.

Humanoid tasks: The Humanoid consists of three primary tasks, each designed to challenge an agent's ability to control a simulated humanoid robot. The three tasks are described as follows:

- **Stand**: The agent's objective is to maintain an upright posture. The reward function encourages stability and a vertical torso position while minimizing deviations from an ideal standing height.
- **Walk**: The agent is rewarded for moving forward at a target velocity of 1 m/s. This task evaluates the agent's capability to coordinate limb movement and maintain balance while walking.

- **Run**: In this task, the agent is required to achieve a high-speed forward motion of 10 m/s. The challenge includes maintaining dynamic stability and efficient stride patterns.

Quadruped tasks: The Quadruped environment represents a four-legged robotic model with two major tasks aimed at testing multi-legged coordination and locomotion:

- **Walk**: The agent must achieve forward movement at a steady pace. This task assesses the stability and synchronization of its four legs during controlled walking.
- **Run**: The agent is required to accelerate to higher velocities, demanding agile gait adjustments and robust stability during high-speed movement.

OpenAI Gym. We chose 5 widely used locomotion tasks in various domains:

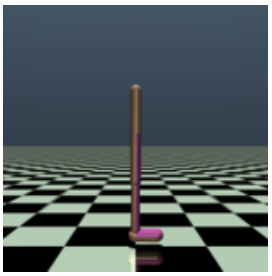


Figure 5: Walker2d-v3

State-action space: $\mathcal{S} \in \mathbb{R}^{17}, \mathcal{A} \in \mathbb{R}^{6}$.

Objective. Maintain forward velocity as fast as possible while avoiding falling over.

Initialization. The walker is initialized in a standing position with slight random noise added to joint positions and velocities.

Termination. The episode ends when the agent falls, the head touches the ground, or after 1000 steps.

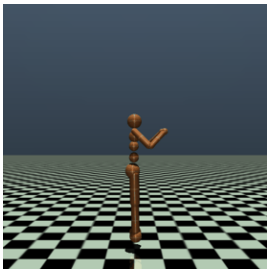


Figure 6: Humanoid-v3

State-action space: $\mathcal{S} \in \mathbb{R}^{376}, \mathcal{A} \in \mathbb{R}^{17}$.

Objective. Maintain balance and walk or run forward at a high velocity while avoiding falls.

Initialization. The humanoid starts in an upright position with slight random perturbations to joint angles and velocities.

Termination. The episode ends when the head height is less than 1.0 meter, the torso tilts excessively, or after 1000 steps.

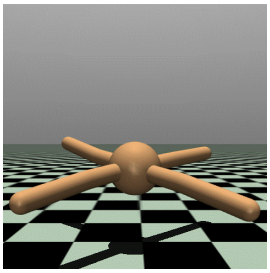


Figure 7: Ant-v3

State-action space: $\mathcal{S} \in \mathbb{R}^{111}, \mathcal{A} \in \mathbb{R}^{8}$.

Objective. Navigate forward as quickly as possible using four legs while maintaining stability.

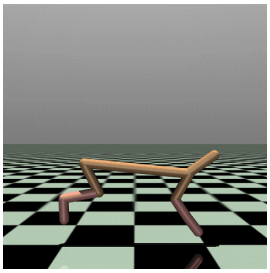
Initialization. The ant is initialized in a stable, upright position with random noise applied to its joints.

Termination. The episode ends if the ant falls, flips over, or reaches the maximum step count of 1000.

B.2 Reproducibility Statement & Detailed Hyperparameters

In MoGE, we adopt the hyperparameter settings without additional fine-tuning and use the same configuration across all previously demonstrated tasks, which are listed in Table 3.

In this paper, all experiments are conducted with a total of 1.5 million environment interaction steps, and the results are averaged over three random seeds. The experiments are performed on an AMD Ryzen Threadripper 3960X 24-Core Processor and an NVIDIA GeForce RTX 4090 GPU. Besides, the walltime results(s) of $1.5M$ steps are included in Table 2.



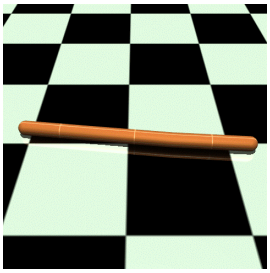
State-action space: $\mathcal{S} \in \mathbb{R}^{17}$, $\mathcal{A} \in \mathbb{R}^{6}$.

Objective. Achieve maximum forward velocity with smooth, coordinated movements.

Initialization. The agent starts with a slight forward tilt and randomized joint noise.

Termination. The episode ends after 1000 steps or if the agent's head touches the ground.

Figure 8: Halfcheetah-v3



State-action space: $\mathcal{S} \in \mathbb{R}^{8}$, $\mathcal{A} \in \mathbb{R}^{2}$.

Objective. Propel forward through water-like dynamics using sinusoidal wave patterns.

Initialization. The swimmer starts in a straight posture with minor random perturbations.

Termination. The episode ends after 1000 steps, with no explicit termination for falling.

Figure 9: Swimmer-v3

The results show that MoGE incurs slightly higher walltime compared to baseline algorithms. Notably, to reduce computational overhead, we perform MoGE training only once every 10 environment interaction steps, which significantly controls the overall compute cost. Moreover, unlike methods requiring retraining the generative model on the buffer until convergence before use, MoGE continuously updates its generator during training. This makes MoGE's diffusion-based training more efficient and lightweight in practice.

C Supplemental clarification

C.1 Design of One-step imagination world model

In MoGE, the structure of the one-step imagination world model is illustrated in Figure 10. Since MoGE is developed under the MDP setting. The reason why modeling dynamics in a latent space instead of directly learning the mapping $f(s, a)ßs'$ is motivated by several practical considerations:

(A)) Learning a compact latent representation helps capture abstract, task-relevant features of the environment, improving generalization and training efficiency even under MDPs. Besides, different dimensions of the state may contribute unevenly to dynamic modeling. Using an encoder to transform the raw state allows the model to extract the most relevant components, which facilitates more accurate transition prediction.

(B) Latent dynamics are often smoother and easier to predict compared to raw state transitions, especially in high-dimensional environments. Fitting dynamics in a latent space typically improves the accuracy of transition modeling.

(C) Decoupling representation learning from dynamics prediction allows better reuse and transfer of components. In fact, in MoGE, the policy network is built on top of the same encoder, enabling a unified state representation that facilitates more effective policy learning.

To further support our claim, a simple experiment is conducted across three environments (each with three random seeds) using the same diffusion and baseline algorithm setup. This experiment compares the TAR between vanilla dynamics (Using the Transformer as a predictor of next states and rewards) and latent dynamics. The results are demonstrated in the Table 4:

The results show that, compared to directly mapping the current state and action to the next state, introducing a latent space leads to better learning of both environment transitions and state representations in the policy network.

Table 2: Performance comparison of MoGE and related baselines on Humanoid-run task.

Env	MoGE	PGR	SER	DSAC
Humanoid-run	151657 $\pm$ 395	207813±1464	210321±788	128153±122

Table 3: Hyperparameter settings.

Hyperparameter	**Value**	**Hyperparameter**	**Value**
Training			
Learning rate	1×10^{-4}	Target network soft-update rate	0.05
Batch size	1024	Buffer size	1_000_000
Optimizer	Adam	Sampling	Uniform
Discount factor (γ)	0.99	Num of vector envs	10 (Only in DMC tasks)
Sample batch size	20		
World Model learning			
Dynamics loss coefficient (β_1)	0.5	Representation loss coefficient (β_2)	0.1
Learning reate	1×10^{-4}	Optimizer	Adam
Diffusion model			
Diffusion steps	100	Noise Schedule	Cosine
Guidance scale	1.0	ϵ-prediction	True
Denoising network	[256,256,256]	Activation	GELU
Optimizer	Adam	Learning rate	1×10^{-5}
Actor			
Minimum policy log std	-20	Policcy network	[256,256,256]
Maximum policy log std	0.5	Activation in hidden dim	GELU
Learning rate	1×10^{-4}	Activation in output dim	Linear
Critic			
Value network	[256,256,256]	Activation in hidden dim	GELU
Learning rate	1×10^{-4}	Activation in output dim	Linear
Architecture (8M)			
Transformer layers	2	Latent space dimension	256
Transformer heads	8	Dropout	0.1
MLP activation	GELU	Normalization	LayerNorm

C.2 Broader baseline experiments with model-free algorithms

To further validate the effectiveness of MoGE compared with active exploration, we perform extra experiments with more mainstream model-free algorithms. All the algorithms are evaluated in standard settings and tested on the 3 OpenAI Gym tasks: Walker2d-v3, Humanoid-v3, and Halfcheetah-v3.

Deep Deterministic Policy Gradient (DDPG) [36]: an off-policy actor-critic method that leverages deterministic policies and experience replay for efficient learning in continuous action spaces.

Trust Region Policy Optimization (TRPO) [50]: an on-policy method that optimizes policies by enforcing a trust region constraint to ensure stable updates.

Proximal Policy Optimization (PPO) [51]: improved upon TRPO by using a clipped surrogate objective for simpler and more efficient training.

Soft Actor-Critic (SAC) [21]: introduces maximum entropy to encourage exploration and improve stability, making it well-suited for complex, high-dimensional tasks.

All the training curves are illustrated in Figure 11 and the detailed results are listed in Table 5.

Table 4: Ablation of MoGE with vanilla vs. latent dynamics on humanoid control tasks.

Env	MoGE w/ latent dynamics	MoGE w/ vanilla dynamics
Humanoid-run	**489** $\pm$ 9	408 $\pm$ 35
Humanoid-walk	**892** $\pm$ 19	684 $\pm$ 71
Humanoid-stand	**907** $\pm$ 7	785 $\pm$ 88

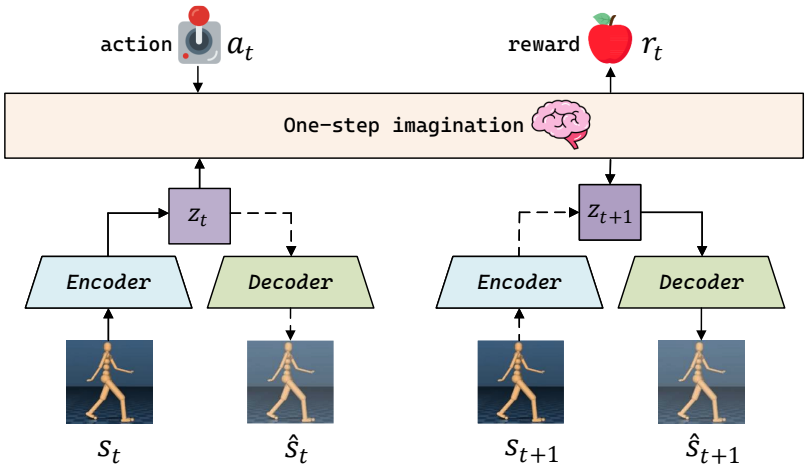


Figure 10: **One-step imagination world model.** During training, the current state s_t is encoded by the representation network h_ϕ into a latent representation $\mathbf{z}_t$. Given this latent state and the action a_t, the one-step world model predicts the next latent state $\mathbf{z}_{t+1}$, the immediate reward $\hat{r}_t$, and termination factor $\hat{c}_t$. The solid-line process represents inference, while the dashed-line process is used for loss function construction.

C.3 Broader baseline experiments with model-based algorithms

Similarly, to validate the performance of MoGE when integrated with reinforcement learning algorithms, we additionally select two mainstream model-based RL algorithms for comparison. These algorithms are sourced from the `https://github.com/nicklashansen/tdmpc2`, and the experiments are performed in DMC Humanoid tasks:

TD-MPC2 [26]: a model-based reinforcement learning algorithm that combines temporal difference learning with model-predictive control to enhance sample efficiency and long-horizon planning.

DreamerV3 [24]: a model-based reinforcement learning framework that leverages a world model for imagination-based training, enabling effective policy learning with high sample efficiency in continuous control tasks.

All the training curves are illustrated in Figure 12 and the detailed results are listed in Table 6.

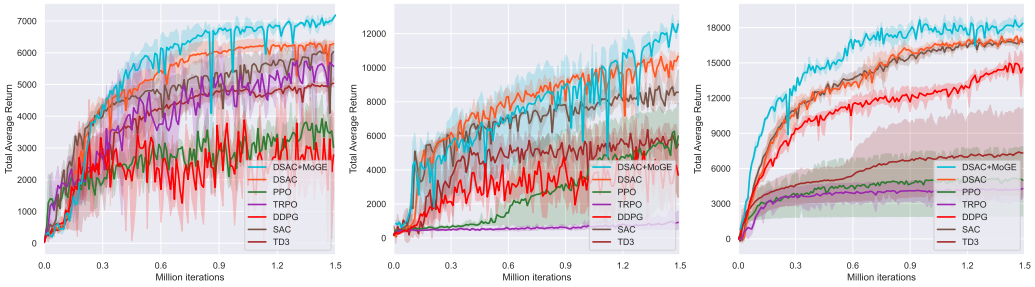
C.4 Broader baseline experiments with modified Actor-Critic algorithms

To provide a more comprehensive evaluation, some half-formulated methods that were modified based on the traditional Actor-Critic method can be included. We conduct the experiments in DMC Humanoid tasks, which are one of the hardest tasks within 3 seeds, and the introductions of the methods are illustrated as follows:

REDQ [9]: a model-free RL algorithm that employs a large ensemble of Q-functions with randomized subset updates to improve sample efficiency and reduce overestimation bias.

DroQ [28]: a lightweight variant of ensemble Q-learning that applies dropout regularization to approximate uncertainty and achieve doubly efficient learning.

BRO [40]: a scalable actor-critic framework that leverages network scaling, regularization, and optimism to balance compute and sample efficiency in continuous control.



(a) Walker2d-v3 (b) Humanoid-v3 (c) Halfcheetah-v3

Figure 11: Supplemental study curves with 6 mainstream model-free algorithms.

Table 5: Total Average Return (TAR) on 3 OpenAI Gym tasks for supplemental experiments. Mean ± Std over 3 seeds. **Bold** = best, Higher is better.

Environment	TD3	SAC	DDPG	TRPO	PPO	DSAC	DSAC+MoGE
Walker2d-v3	5031.1 ± 84.2	5997.0 ± 291.1	3268.1 ± 240.9	5635.6 ± 211.9	3880.9 ± 327.7	6501.1 ± 87.3	**6978.4 ± 68.7**
Humanoid-v3	5967.1 ± 547.8	8831.7 ± 352.2	4548.7 ± 807.6	947.2 ± 503.9	6011.0 ± 2014.4	11004.0 ± 121.5	**12151.1 ± 35.4**
Halfcheetah-v3	7363.0 ± 3666.4	16921.3 ± 380.8	14793.1 ± 462.3	4207.4 ± 756.6	5139.2 ± 2392.8	17324.7 ± 41.1	**18054.9 ± 459.6**
AVG.GYM	6120.4 ± 1432.8	10583.3 ± 341.4	7536.6 ± 503.6	3596.7 ± 490.8	5010.4 ± 1578.3	11472.7 ± 93.0	**12394.8 ± 187.9**

Simba [33]: a large-scale RL framework that exploits simplicity bias in over-parameterized networks to improve stability, generalization, and training efficiency.

As shown in Table 7, except for the simplest case (Humanoid-Stand), MoGE achieves the best performance across all more challenging tasks.

C.5 Broader baseline experiments with enhanced exploration approaches

We have conducted additional experiments to study the effect of different exploration strategies on the DSAC algorithm via ablation comparisons. Due to the development and implementation effort involved, and time constraints, we focus on three representative environments—the Humanoid tasks in DMC—for this evaluation. The three approaches are as follows:

Plan2Explore [52]: a model-based exploration method that uses self-supervised world models to plan informative trajectories without external rewards.

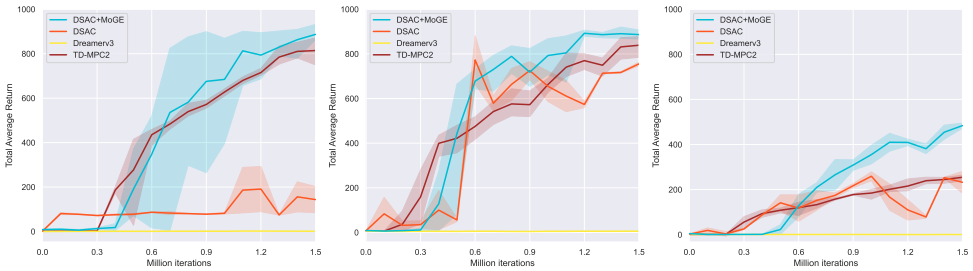
MaxInfoRL [58]: an exploration framework that maximizes mutual information between states, actions, and returns to encourage diverse and informative behaviors.

OMBRL [59]: a model-based exploration algorithm that incorporates optimism through intrinsic reward shaping, enabling scalable and principled exploration.

Demonstrated in Table 8, the results show that MoGE still offers a significant advantage compared to these methods. Intrinsic exploration methods encourage visiting novel or unpredictable states, but their signals (e.g., prediction error, uncertainty) are often task-agnostic and may lead to uninformative or misaligned exploration. In contrast, MoGE leverages task-aware utility functions (e.g., TD-error, entropy) to generate states that are directly aligned with policy improvement objectives**. Besides, unlike intrinsic reward methods that require careful reward balancing and affect the policy's optimization objective, MoGE decouples exploration from reward design, enabling more stable training without interfering with the task-specific learning signal.

D Limitation and Future Work

While MoGE demonstrates strong performance in enhancing exploration and improving sample efficiency in reinforcement learning, several limitations remain. First, the generation of critical states through the diffusion-based generator introduces additional computational overhead compared to



(a) Humanoid-walk (b) Humanoid-stand (c) Humanoid-run

Figure 12: Supplemental study curves with 2 mainstream model-based algorithms.

Table 6: Total Average Return (TAR) on 3 DMC Suite tasks for supplemental experiments. Mean ± Std over 3 seeds. **Bold** = best, Higher is better.

Environment	DSAC	DreamerV3	TD-MPC2	DSAC+MoGE
Humanoid-walk	146.5 ± 60.9	0.9 ± 0.4	814.3 ± 49.8	**891.7 ± 19.1**
Humanoid-stand	776.6 ± 15.6	5.6 ± 0.3	838.9 ± 39.2	**907.5 ± 6.9**
Humanoid-run	267.4 ± 3.9	0.8 ± 0.4	254.6 ± 11.1	**488.9 ± 8.7**
AVG.DMC	396.8 ± 26.8	2.4 ± 0.4	635.9 ± 33.4	**762.7 ± 11.6**

standard replay buffer sampling. Second, MoGE assumes that the state distribution learned by the generator aligns well with the replay buffer's occupancy measure. In practice, minor discrepancies may arise. Overall, while MoGE enhances exploration capabilities, its time cost is influenced by the quality of the learned state distribution and the smoothness of the conditional diffusion process.

In the future, we may explore integrating MoGE with on-policy RL frameworks to enable real-time generation of critical states, further enhancing exploration efficiency during live interactions. Additionally, investigating more expressive utility functions for the diffusion-based generator could improve the selection of high-value states, optimizing policy learning. Moreover, dynamically adjusting the generator's sampling strategy based on task complexity and training progress may further boost robustness and generalization.

E Positive and Negative Social Impact

Our method, MoGE, performs exploration augmentation in reinforcement learning by generating critical samples through a diffusion-based generator and a world model, significantly improving sample efficiency and policy performance in complex control tasks. This capability has promising implications for real-world applications such as embodied AI, autonomous driving, and large-scale decision-making systems, where efficient exploration of vast state spaces is crucial. However, the ability to synthetically generate exploration samples may lead to overconfidence in simulation-trained policies, thereby increasing risks if these policies are deployed prematurely in real-world environments. We advocate for careful validation and consideration of ethical implications to ensure the responsible and safe application of MoGE.

Table 7: TAR for supplemental experiments with **modified AC methods**.

Env	DSAC+MoGE	Simba	BRO	RedQ	DroQ
Humanoid-run	**489** $\pm$ 9	268 $\pm$ 40	417 $\pm$ 15	187 $\pm$ 12	164 $\pm$ 21
Humanoid-walk	**892** $\pm$ 19	801 $\pm$ 13	881 $\pm$ 25	665 $\pm$ 5	682 $\pm$ 14
Humanoid-stand	907 $\pm$ 7	**920** $\pm$ 14	905 $\pm$ 3	902 $\pm$ 4	896 $\pm$ 6

Table 8: TAR for supplemental experiments with **enhanced exploration approaches**.

Env	DSAC+MoGE	Plan2Explore	MaxInfoRL	OMBRL
Humanoid-run	**489** $\pm$ 9	311 $\pm$ 12	197 $\pm$ 4	262 $\pm$ 13
Humanoid-walk	**892** $\pm$ 19	588 $\pm$ 7	481 $\pm$ 7	678 $\pm$ 5
Humanoid-stand	907 $\pm$ 7	801 $\pm$ 14	844 $\pm$ 8	769 $\pm$ 11

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We emphasize our core contributions in the abstract and introduction, highlighting the critical state generator, the one-step imagination world model, and the proposed RL training framework combined with MoGE.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Appendix D we discuss that the limitation of this work is the problem of time cost and the bias in critical state generation.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide complete assumptions in the formulation of the theorem and a complete proof of the theorem in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We give details of the hyperparameters required for the experiments in the Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We will make the full code publicly available when the paper is accepted, but the core code is public with open access in Appendix B.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In the appendix section, we give a very detailed selection of all algorithmic hyperparameters, optimizers.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Whether the data is in a table or a picture, we report the error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In the Evaluation Setups in Appendix B, we give the CPU and GPU servers used for the computation and the time used to train 1.5 million using Humanoid-run as an example.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss both the potential positive societal impacts and negative societal impacts of the work in Appendix E

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: All our baseline algorithms are evaluated under standard settings, and all open-source implementations are properly cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.