Where-to-Learn: Analytical Policy Gradient Directed Exploration for On-Policy Robotic Reinforcement Learning

Leixin Chang[1,2], Xinchen Yao[1], Ben Liu[3], Liangjing Yang[1,2], Hua Chen[1,†]

***Abstract*—On-policy reinforcement learning (RL) algorithms have demonstrated great potential in robotic control, where effective exploration is crucial for efficient and high-quality policy learning. However, how to encourage the agent to explore the better trajectories efficiently remains a challenge. Most existing methods incentivize exploration by maximizing the policy entropy or encouraging novel state visiting regardless of the potential state value. We propose a new form of directed exploration that uses analytical policy gradients from a differentiable dynamics model to inject task-aware, physics-guided guidance, thereby steering the agent towards high-reward regions for accelerated and more effective policy learning. We integrate our exploration approach into a widely used on-policy RL algorithm, Proximal Policy Optimization, to test and demonstrate its effectiveness. We conduct extensive benchmark experiments and demonstrate the effectiveness of the proposed exploration augmentation method. We further test our approach on a 6-DOF point-foot robot for velocity tracking locomotion, and conduct the simulation test and implement a successful sim-to-real deployment as the ultimate validation. Project Page: `wheretolearn.github.io`.**

***Index Terms*—Legged robot, reinforcement learning, model learning for control.**

I. Introduction

ON-POLICY Reinforcement Learning (RL) has demonstrated great potential in various robotic control problems in recent years. [1]–[4]. RL holds the promise of synthesizing a complex robotic controller by leveraging experiential data from robot-environment interactions, guided by objectives defined as task rewards. Policy updates heavily depend on the quality of collected trajectories during exploration, where high-return samples provide positive gradients that reinforce desirable behaviors, while low-return samples suppress undesirable ones. Consequently, the effectiveness of policy improvement relies on whether exploration produces trajectories that are sufficiently informative, highlighting the importance of exploration. Current on-policy model-free RL algorithms, such as Proximal Policy Optimization (PPO) [5], are quite data-expensive to train for the high-dimensional control problem. One of the reasons for the low sample efficiency is that the exploration for data basically relies on the undirected, random perturbations derived from the policy's inherent stochasticity [6], entropy maximization [7] and noise injection [8]. This Brownian-motion style of exploration is agnostic to the environment's underlying physics, making the discovery of high-reward regions a slow and inefficient process. Model-based Reinforcement Learning improves sample efficiency by generating synthetic trajectories from a learned dynamics model [9]–[11], but suffers from compounding errors and distributional mismatch [12].

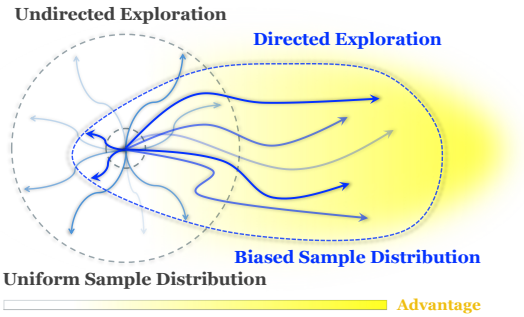

Fig. 1: Illustration of the proposed directed exploration.

In this work, we try to implement more sample-efficient and stable RL training by injecting dynamics priors into the exploration process, making the random exploration more directed. Specifically, rather than directly using the dynamics model to generate the trajectory data, we make use of the analytical policy gradient propagated from a short-term return objective through the differentiable dynamics model, producing an exploratory policy with short-horizon foresight. The exploratory policy is rolled out to generate more informative and directional trajectory data that serve as the augmentation of the data collected by the primary policy in a standard on-policy algorithm like PPO. Finally, the policy is trained on the augmented exploration data with the PPO stable training mechanism.

The main contribution of this work is a novel exploration augmentation method for on-policy RL, which leverages the differentiability of dynamics as a kind of inductive bias or dynamics priors to guide exploration and thus enhance the sample efficiency of the learning process. Further, we theoretically justify our method's mechanism under ideal premises. Specifically, we prove that exploratory policy constitutes a policy improvement and that the exploratory data yield a con-

Manuscript received: October 11, 2025; Revised: February 2, 2026; Accepted: March 3, 2026.

This paper was recommended for publication by Editor Cosimo Della Santina upon evaluation of the Associate Editor and Reviewers comments.

This work was supported in part by the State Key Laboratory (SKL) of Biobased Transportation Fuel Technology, and in part by the Industrial Technology Development Project Grant on Cyber-Physical Control System Design under Grant K20243390.

[1] Zhejiang University-University of Illinois Urbana-Champaign Institute (ZJUI), Haining, China. Emails: {`leixin.23`, `xinchen.22`, `liangjingyang`, `huachen`}`@intl.zju.edu.cn`.

[2] School of Mechanical Engineering, Zhejiang University, Hangzhou, China.

[3] Southern University of Science and Technology, Shenzhen, China. Email: `liub2021@mail.sustech.edu.cn`.

† Corresponding Author.

Digital Object Identifier (DOI): see top of this page.

sistently positive learning signal. Finally, comprehensive empirical experiments, including benchmark tests and sim-to-real experiments on a biped robot, are conducted to demonstrate the effectiveness of our method in exploration augmentation and improving sample efficiency over the PPO algorithm.

II. Related Work

A. Exploration in Reinforcement Learning

In on-policy RL, policy improvement relies on generating sufficiently informative trajectories. Existing methods enhance exploration by rewarding policy stochasticity, e.g., Maximum Entropy [7], [13], or employing task-agnostic intrinsic rewards based on novelty [14]–[16]. Fundamentally differing from the prior work, our approach derives exploration signals directly from the task objective and system dynamics. Leveraging differentiable simulation, we inject dynamics priors to provide physics-informed guidance, rendering exploration efficient and task-oriented. This distinguishes our method by prioritizing exploration *direction* over *breadth*, providing a purposeful *thrust* toward high-reward regions and rendering the learning process significantly more efficient and task-oriented.

B. Policy Learning via Differentiable Dynamics

Differentiable dynamics allow backpropagating analytical gradients from the objective directly to policy parameters [17]–[19], enabling the Analytical Policy Gradient (APG) method. Previous works applied the APG to quadruped locomotion control [20]–[22] or vision-based aerial robot control [23]. Other works integrate APG into actor-critic frameworks [24], [25], using value functions to approximate terminal values for stable training. While APG offers efficient, low-variance gradients [26], [27], it suffers from numerical instability due to contact discontinuities and gradient explosion or vanishing over long horizons [26], [27]. Rather than directly optimizing parameters, we use APG to guide exploration. This yields informative trajectories while circumventing the stability issues of direct analytical optimization.

C. Sample-Efficient Reinforcement Learning

Sample efficiency is a critical bottleneck in robotic RL due to high physical interaction costs. Existing approaches enhance efficiency via off-policy data reuse or model-based priors. Off-policy algorithms like SAC [7] and TD3 [28] utilize replay buffers and double-Q learning to maximize data utility and ensure stability in continuous action spaces. AWR [29] further improves stability by framing policy updates as a weighted supervised learning problem, making it highly effective for learning from demonstrations. Concurrently, model-based RL (MBRL) leverages learned dynamics to generate synthetic trajectories, expand value estimation horizons [9]–[11], [30]–[32], or perform MPC-style planning [33], [34]. However, traditional MBRL often treats dynamics as a black box. In contrast, we leverage analytical gradients from differentiable dynamics to provide physics-informed exploration guidance. This enables directed discovery of high-reward regions, combining on-policy stability with the rapid search capabilities of first-order dynamics priors.

III. Preliminary

A. Reinforcement Learning for Robotic Control

Through the lens of reinforcement learning, the robotic control problem is commonly modeled as a Markov Decision Process (MDP), defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ represents the action space, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition kernel representing the environment dynamics $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ denotes the reward function that assigns a scalar reward to each state transition, quantifying the immediate benefit of taking action $\boldsymbol{a}_t$ in state $\boldsymbol{s}_t$ under transition dynamics described by $\mathcal{T}$. The agent learns a policy $\pi_{\boldsymbol{\theta}}$ to maximizes the discounted cumulative reward $\mathbb{E}_{\tau\sim\pi_{\boldsymbol{\theta}}}\left[\sum_t \gamma^t r_t\right]$, where τ denotes a sampled trajectory under policy $\pi_{\boldsymbol{\theta}}$, r_t represents the immediate reward at time t and $\gamma \in [0, 1)$ is the discounted factor. As mentioned in Section II-A, exploration plays a crucial role in effective policy learning. Typically, on-policy RL exploration mainly comes from policy stochasticity and entropy maximization. Our method augments the exploration by leveraging physics priors contained in the environment dynamics, aiming to achieve more effective and task-oriented exploration.

B. Policy Learning with Analytical Policy Gradient

Conceptually, the environment dynamics can be treated as an abstract function $\boldsymbol{s}_{t+1} = \boldsymbol{f}(\boldsymbol{s}_t, \boldsymbol{a}_t)$ representing the mapping $\mathcal{F} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, where $\boldsymbol{s}_t$ is the system state and $\boldsymbol{a}_t$ is the control input at time step t. Notably, here the dynamics mapping $\mathcal{F}$ is deterministic and can be regarded as a degenerate form of the stochastic transition kernel $\mathcal{T}$ in the MDP formulation of RL. And the environment receives a reward signal $r(\boldsymbol{s}_t, \boldsymbol{a}_t)$ at time step t. The control policy can be represented by a neural network $\pi_{\boldsymbol{\theta}}(\boldsymbol{s})$ that takes $\boldsymbol{s}_t$ as input and outputs action $\boldsymbol{a}_t$. If $\boldsymbol{f}$ is differentiable w.r.t. $\boldsymbol{s}_t$ and $\boldsymbol{a}_t$, the system has differentiable dynamics, enabling gradient propagation for policy learning, as explored in APG [17], [25] and First-order Gradient (FoG) [35].

$$\mathcal{L}_{\boldsymbol{\theta}}^{\text{APG}} = -\sum_{t=0}^{h-1} r(\boldsymbol{s}_t, \boldsymbol{a}_t) = -\sum_{t=0}^{h-1} r(\boldsymbol{s}_t, \pi_{\boldsymbol{\theta}}(\boldsymbol{s}_t)), \tag{1}$$

where h refers to the trajectory horizon length. Following *Backpropagation-Through-Time* (BPTT) technique [19], the gradient of $\mathcal{L}_{\boldsymbol{\theta}}^{\text{APG}}$ in terms of policy parameters $\boldsymbol{\theta}$ can be expressed as (2),

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\boldsymbol{\theta}}^{\text{APG}} = -\frac{1}{h}\sum_{t=0}^{h}\left[\frac{\partial r_t}{\partial \boldsymbol{a}_t}\frac{\mathrm{d}\boldsymbol{a}_t}{\mathrm{d}\boldsymbol{\theta}} + \sum_{k=1}^{t}\frac{\partial r_t}{\partial \boldsymbol{s}_t}\left(\prod_{i=k}^{t}\frac{\partial \boldsymbol{s}_i}{\partial \boldsymbol{s}_{i-1}}\right)\frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{\theta}}\right], \tag{2}$$

where the matrix of partial derivatives $\frac{\partial \boldsymbol{s}_i}{\partial \boldsymbol{s}_{i-1}}$ is the Jacobian of differentiable dynamics $\boldsymbol{f}$. Thus, assuming differentiable dynamics and rewards, the policy gradient is computed analytically via BPTT as shown in (2).

However, the APG method is challenged by the presence of the noisy optimization landscape, gradient exploding and vanishing and empirical bias of gradient computing in the contact-rich environment, which results in unstable training

[24], [27], [36]. In our method, APG is repurposed from a policy optimizer into an exploration-guidance mechanism providing informative trajectories for model-free policy optimization.

IV. Methodology

We aim to augment the undirected exploration of on-policy RL to be a directed and task-oriented exploration by changing the data distribution used for policy network updates. By utilizing directional gradients from environment dynamics to guide exploration toward high-reward regions, our method accelerates policy learning while circumventing the noisy landscapes and gradient instability inherent to APG.

A. Analytical Policy Gradient Augmented Exploration

In each iteration, the exploratory policy is initialized from primary policy and undergoes APG update. We then collect trajectories using both policies in parallel environments and merge them into an augmented dataset. For the update of the policy network and the critic network, we update the critic network with only the data collected by the primary policy and use the aforementioned augmented trajectory dataset to update the policy network under the PPO update rule. The complete pipeline is illustrated in Fig. 2.

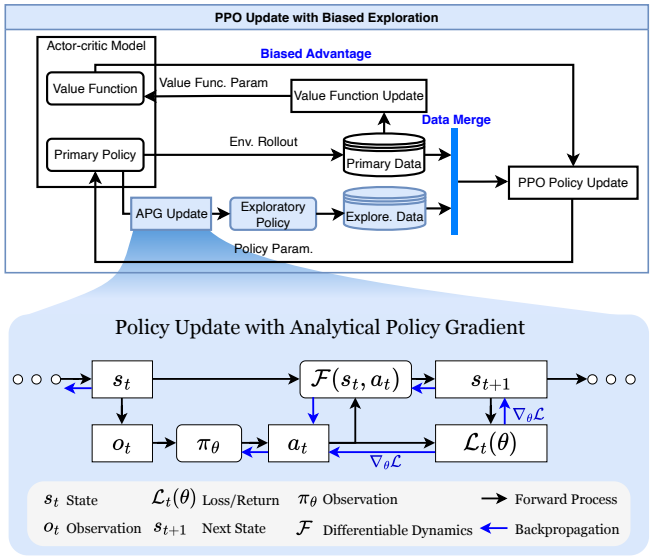

Fig. 2: Method Overview.

1) Policy Update: We use the APG method SHAC [24][1] to update the current primary policy to obtain a temporary exploratory policy. The exploratory policy collects trajectories to augment the original PPO data. To make the experience data more informative in terms of task and dynamics, this exploratory data is merged with the trajectory data collected by the primary PPO policy itself, forming a richer and dynamics-informed dataset for training. Crucially, high-reward regions discovered by the exploratory policy yield large positive advantages against the primary value baseline. This creates a strong learning signal that guides the agent to approach a more promising region in the state-action space. In addition, we use the PPO algorithm to update the policy in the augmented dataset to keep the training stable by leveraging the proximity property of the PPO update. In every policy update iteration, as illustrated in the Fig. 3, the policy parameter initialized from $\boldsymbol{\theta}_k^{\text{primary}}$ is updated with APG to yield a temporary exploratory policy $\pi_{\boldsymbol{\theta}_k^{\text{explore}}}$, where the APG update process is outlined in the bottom frame of Fig. 2. Following the SHAC method [24], the initialized exploratory policy $\pi_{\boldsymbol{\theta}_k^{\text{primary}}}$ is rolled out to collect N sample trajectories with a horizon length h parallel in the simulation. The terminal value of each trajectory is estimated by the critic network, forming an infinite-horizon objective to alleviate the local minima problem of the analytical gradients [24]. The optimization objective for the exploratory policy can be formulated as (3) following [24], where γ is the discount factor. Note that the right part of (3) is the state-action value function, a.k.a. the Q function. Following SHAC [24] and BPTT [19], we further derive the policy gradient of $\mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}}$ shown as (4), with (5) holding when $t_0 \leq t < t_0 + h$. We use the differentiable physics engine `Brax` [17] to implement the gradient computation with JAX autograd function. Specifically, the exploratory policy is modeled as a Gaussian policy as the primary policy. Thus, the reparameterization sampling method is employed for the stochastic policy to enable the computation

[1]We refer to SHAC as APG in the following to emphasize our contribution lies in the general paradigm of exploration augmentation.

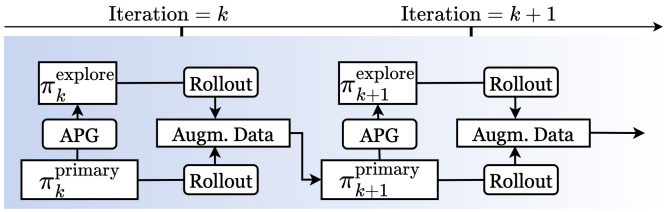

Fig. 3: Policy update iteration of the proposed method. In every iteration, the exploratory policy π_k^{explore} is discarded after exploratory data collection.

Algorithm 1 PPO with APG Directed Exploration

Input: initial policy parameter $\boldsymbol{\theta}_0^{\text{primary}}$, value function parameter $\boldsymbol{\phi}_0^{\text{primary}}$, number of iteration K.
for $k = 0, 1, 2, \cdots, K$ **do**
  Initialize $\pi_{\boldsymbol{\theta}_k^{\text{explore}}}$ from $\pi_{\boldsymbol{\theta}_k^{\text{primary}}}$
  if $k \bmod f = 0$ **then**
    for $i = 0, 1, \cdots, e-1$ **do** ▷ APG Epoch
      Rollout $\pi_{\boldsymbol{\theta}_k^{\text{explore}}}$ for h steps with N parallel agents.
      Query $V_{\boldsymbol{\phi}_k}$ for terminal value to construct loss (3).
      Update $\boldsymbol{\theta}_k^{\text{explore}}$ with (3).
    end for
    Rollout $\pi_{\boldsymbol{\theta}_k^{\text{primary}}}$ with $(1-\alpha)\cdot n$ environments for $\mathcal{D}_k^{\text{primary}}$.
    Rollout $\pi_{\boldsymbol{\theta}_k^{\text{explore}}}$ with $\alpha \cdot n$ environments for $\mathcal{D}_k^{\text{explore}}$.
    Merge $\mathcal{D}_k^{\text{primary}}$ and $\mathcal{D}_k^{\text{explore}}$ to form $\mathcal{D}_k^{\text{aug}}$.
    Compute advantage $\hat{A}_t$ based on $V_{\boldsymbol{\phi}_k}^{\pi^{\text{primary}}}$, $\mathcal{D}_k^{\text{aug}}$.
    Update $\pi_{\boldsymbol{\theta}_k^{\text{primary}}}$ with $\mathcal{D}_k^{\text{aug}}$ by PPO loss (6).
    Update $V_{\boldsymbol{\phi}_k}^{\pi^{\text{primary}}}$ with $\mathcal{D}_k^{\text{primary}}$ and (8).
    Discard $\pi_{\boldsymbol{\theta}_k^{\text{explore}}}$.
  else
    Perform standard PPO to update $\pi_{\boldsymbol{\theta}_k^{\text{primary}}}$.
  end if
end for
Output: policy parameter $\boldsymbol{\theta}_K^{\text{primary}}$, value function parameter $\boldsymbol{\phi}_K^{\text{primary}}$.

of $\frac{\partial \pi_{\boldsymbol{\theta}}(\boldsymbol{s})}{\partial \boldsymbol{\theta}}$ and $\frac{\partial \pi_{\boldsymbol{\theta}}(\boldsymbol{s})}{\partial \boldsymbol{s}_t}$.

$$\mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}} = -\frac{1}{Nh}\sum_{i=1}^{N}\left[\left(\sum_{t=t_0}^{t_0+h-1}\gamma^{t-t_0} r(\boldsymbol{a}_t^i, \boldsymbol{s}_t^i)\right) + \gamma^h V_\phi(\boldsymbol{s}_{t+h}^i)\right] \tag{3}$$

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}} = \sum_{i=1}^{N}\sum_{t=t_0}^{t_0+h-1}\left(\frac{\partial \mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}}}{\partial \boldsymbol{a}_t^i}\right)\left(\frac{\partial \pi_{\boldsymbol{\theta}}(\boldsymbol{s}_t^i)}{\partial \boldsymbol{\theta}}\right) \tag{4}$$

$$\begin{cases} \dfrac{\partial \mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}}}{\partial \boldsymbol{a}_t^i} = -\gamma^{t-t_0}\dfrac{1}{Nh}\dfrac{r(\boldsymbol{a}_t^i, \boldsymbol{s}_t^i)}{\partial \boldsymbol{a}_t^i} + \dfrac{\partial \mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}}}{\partial \boldsymbol{s}_{t+1}^i}\dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{a}_t^i} \\ \dfrac{\partial \mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}}}{\partial \boldsymbol{s}_t^i} = -\gamma^{t-t_0}\dfrac{1}{Nh}\dfrac{r(\boldsymbol{a}_t^i, \boldsymbol{s}_t^i)}{\partial \boldsymbol{s}_t^i} + \\ \qquad \left(\dfrac{\partial \mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}}}{\partial \boldsymbol{s}_{t+1}^i}\right)\left(\dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{s}_t^i} + \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{a}_t^i}\dfrac{\partial \pi_{\boldsymbol{\theta}}}{\boldsymbol{s}_t^i}\right) \\ \dfrac{\partial \mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}}}{\partial \boldsymbol{s}_{t_0+h}^i} = -\gamma^h\dfrac{1}{Nh}\dfrac{\partial V_\phi(s_{t_0+h}^i)}{\partial \boldsymbol{s}_{t_0+h}^i}. \end{cases} \tag{5}$$

In parallel, $\pi_{\boldsymbol{\theta}_k^{\text{explore}}}$ and $\pi_{\boldsymbol{\theta}_k^{\text{primary}}}$ collect trajectory datasets $\mathcal{D}_k^{\text{explore}}$ and $\mathcal{D}_k^{\text{primary}}$ using $\alpha \cdot n$ and $(1-\alpha)\cdot n$ environments, respectively. These are subsequently merged to form the augmented dataset $\mathcal{D}_k^{\text{aug}}$. Then two dataset $\mathcal{D}_k^{\text{explore}}$ and $\mathcal{D}_k^{\text{primary}}$ are merged to form the augmented dataset $\mathcal{D}_k^{\text{aug}}$. Then, the primary policy parameter is updated on this augmented dataset $\mathcal{D}_k^{\text{aug}}$ with the surrogate loss function (6), where $\hat{A}$ denotes the advantage estimate at time step t, computed using the Generalized Advantage Estimation (GAE) to guide the policy update. Afterward, $\pi_{\boldsymbol{\theta}_k^{\text{explore}}}$ is re-initialized from the updated primary policy to prevent APG instability [27] and excessive distributional shifts.

$$\mathcal{L}_{\boldsymbol{\theta}} = -\frac{1}{|\mathcal{D}_k^{\text{aug}}|T}\sum_{\tau\in\mathcal{D}_k^{\text{aug}}}\sum_{t=0}^{T}\Big[\min\Big(\eta_t(\boldsymbol{\theta})\hat{A}_t(\boldsymbol{s}_t,\boldsymbol{a}_t), \\ \text{clip}\left(\eta_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon\right)\hat{A}_t(\boldsymbol{s}_t,\boldsymbol{a}_t)\Big) + \beta\mathcal{H}\big(\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s}_t)\big)\Big] \tag{6}$$

$$\eta_t(\boldsymbol{\theta}) = \begin{cases} \dfrac{\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\boldsymbol{\theta}_k^{\text{primary}}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}, & \text{if } (\boldsymbol{a}_t,\boldsymbol{s}_t)\in\mathcal{D}_k^{\text{primary}} \\ \dfrac{\pi_{\boldsymbol{\theta}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}{\pi_{\boldsymbol{\theta}_k^{\text{explore}}}(\boldsymbol{a}_t|\boldsymbol{s}_t)}, & \text{if } (\boldsymbol{a}_t,\boldsymbol{s}_t)\in\mathcal{D}_k^{\text{explore}}. \end{cases} \tag{7}$$

The surrogate loss (6) is constructed on the composite data $\mathcal{D}_k^{\text{aug}}$ collected by both a primary and an exploratory policy. To ensure unbiased policy gradient estimation, we design a piecewise importance sampling ratio $\eta_t(\boldsymbol{\theta})$ shown as (7). In addition, we also incorporate the policy entropy bonus $\mathcal{H}\big(\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{s}_t)\big)$ in (6) following the original PPO algorithm design [5]. Notably, the reason we retain data from the primary policy rather than relying solely on the exploratory data for policy learning is that APG-guided trajectories, while task-oriented, may lead the policy toward suboptimal behaviors and suffer from training instability induced by low-quality gradients of differentiable dynamics implementation [27]. The primary data here $\mathcal{D}_k^{\text{primary}}$ complements this by preserving broader exploration.

2) Critic Update: For value estimation, the value function $V_{\phi_k}^{\pi^{\text{primary}}}$ is trained by minimizing the mean squared error against the estimated returns $\hat{R}_t$, as defined in (8).

$$\mathcal{L}_\phi = \frac{1}{|\mathcal{D}_k^{\text{primary}}|T}\sum_{\tau\in\mathcal{D}_k^{\text{primary}}}\sum_{t=0}^{T}\left(V_{\phi_k}^{\pi^{\text{primary}}}(s_t) - \hat{R}_t\right)^2 \tag{8}$$

Notably, value function $V_{\phi_k}^{\pi^{\text{primary}}}$ fitting is on the data collected on the primary policy $\pi_{\boldsymbol{\theta}_k^{\text{primary}}}$ rather than $\pi_{\boldsymbol{\theta}_k^{\text{explore}}}$, which means $V_{\phi_k}^{\pi^{\text{primary}}}$ only depends on $\pi_{\boldsymbol{\theta}_k^{\text{primary}}}$. Consequently, $V_{\phi_k}^{\pi^{\text{primary}}}$ could provide an unbiased on-policy value baseline for the advantage estimation. We use the APG update frequency f and the number of update epochs e to control the generation and usage of the exploratory policy. Specifically, an APG update is performed e times to obtain the exploratory policy and the exploratory data collection is executed once every f training iterations. The detailed pipeline is given in Algorithm 1 and Fig. 2.

B. Analysis on Mechanism of Exploration Augmentation

In this section, we analyze how the proposed method guides the policy learning process through the biased advantage introduced by the analytical gradient.

1) Directed Exploration versus Undirected Exploration: Standard on-policy exploration relies on undirected stochasticity or novel state visiting, which promotes broad coverage but is often sample-inefficient in high-dimensional spaces due to its disregard for potential state values [5], [7], [13], [14]. In contrast, we generate a directed exploratory policy, $\pi_{\boldsymbol{\theta}^{\text{explore}}}$, by directly backpropagating the analytical gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}}$ through the differentiable dynamics. Since the objective (3) estimates the discounted return, the resulting gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}}$ explicitly points toward the direction of steepest return maximization in the short horizon. Consequently, $\pi_{\boldsymbol{\theta}_k^{\text{explore}}}$ executes directed exploration, collecting trajectories biased toward high-reward regions that the primary policy may not yet cover. Integrating this informative data $\mathcal{D}^{\text{explore}}$ into the update step provides a high-quality learning signal, thereby accelerating convergence toward optimal policy regions.

2) Biased Advantage as Augmented Learning Signal: In this section, the theoretical mechanism of how the exploratory data improves policy learning is detailed. As mentioned in Section IV-B1, the exploratory policy training loss (3) is an approximation for the negative normalized discounted return, as shown in (9). Here, we take the ideally precise form of exploratory policy loss for theoretical analysis. In the following, we try to justify that the expectation of advantage computed on the exploratory data $\mathcal{D}^{\text{explore}}$ is equal to or greater than that computed on $\mathcal{D}^{\text{primary}}$.

$$\mathcal{L}_{\boldsymbol{\theta}}^{\text{explore}} = -\frac{1}{hN}\sum_{i=1}^{N}\underbrace{\left[\left(\sum_{t=t_0}^{t_0+h-1}\gamma^{t-t_0} r(\boldsymbol{a}_t^i, \boldsymbol{s}_t^i)\right) + \gamma^h V_\phi(\boldsymbol{s}_{t+h}^i)\right]}_{\text{Discounted Return Approximation}} \\ \approx -\frac{1}{h}\cdot\mathbb{E}_{\tau\sim\pi_{\boldsymbol{\theta}}}\left[\sum_t \gamma^t r_t\right] \tag{9}$$

Lemma IV.1 *(Policy Improvement from an APG Update)*
Let the performance objective be $J(\boldsymbol{\theta}) = \mathbb{E}_{\tau\sim\pi_{\boldsymbol{\theta}}}[\sum_t \gamma^t r_t]$. *Suppose the exploratory policy is obtained by one gradient ascent step:*

$$\boldsymbol{\theta}_k^{explore} = \boldsymbol{\theta}_k + \eta \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_k}. \tag{10}$$

Then $\pi_{\theta_k^{explore}}$ *is guaranteed to be an improvement over* π_{θ_k}

Proof. Taylor expansion around θ gives:

$$J(\boldsymbol{\theta}_k^{\text{explore}}) \approx J(\boldsymbol{\theta}_k) + (\boldsymbol{\theta}_k^{\text{explore}} - \boldsymbol{\theta}_k)\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_k), \tag{11}$$

$$J(\boldsymbol{\theta}_k^{\text{explore}}) = J(\boldsymbol{\theta}_k) + \eta \|\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}_k)\|^2 \geq J(\boldsymbol{\theta}) \tag{12}$$

Thus, exploratory policy improves over primary policy. □

Theorem IV.2 *(Superiority of the Exploratory Policy in Expected Advantage)*
Let $\pi_k^{primary}$ *be the primary policy and* $\pi_k^{explore}$ *be the exploratory policy obtained from Lemma IV.1. We assume that the value function* V_k *is a perfect estimator of the true value of the primary policy:* $V_k(s) = V^{\pi_k^{primary}}(s)$. *Then*

$$\mathbb{E}_{(s,a)\sim\pi_k^{explore}}[\hat{A}^{\pi_k^{primary}}(s,a)] \geq \mathbb{E}_{(s,a)\sim\pi_k^{primary}}[\hat{A}^{\pi_k^{primary}}(s,a)]$$

Proof. For data from π_k^{primary},

$$\mathbb{E}_{(s,a)\sim\pi_k^{\text{primary}}}[\hat{A}^{\pi_k^{\text{primary}}}(s,a)] = 0. \tag{13}$$

For data from π_k^{explore}, the policy improvement theorem [37], [38] gives

$$J(\pi_k^{\text{explore}}) - J(\pi_k^{\text{primary}}) = \mathbb{E}_{\tau\sim\pi_k^{\text{explore}}}\left[\sum_t \gamma^t \hat{A}^{\pi_k^{\text{primary}}}(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] \geq 0. \tag{14}$$

Hence,

$$\mathbb{E}_{(s,a)\sim\pi_k^{\text{explore}}}[\hat{A}^{\pi_k^{\text{primary}}}(s,a)] \geq 0 = \mathbb{E}_{(s,a)\sim\pi_k^{\text{primary}}}[\hat{A}^{\pi_k^{\text{primary}}}(s,a)]. \tag{15}$$

This completes the proof. □

Theorem IV.2 formally establishes that data collected by the exploratory policy are superior on average to the primary policy's own value baseline, providing the theoretical grounding for our learning mechanism. The systematic bias forms the source of the augmented learning signal, where the strong gradient from the non-negative advantage directs the policy to increase the action probability in the newly discovered high-return trajectories. While Theorem IV.2 provides the theoretical grounding, the premises may not hold perfectly in practice. The analytical gradient quality and thus the consequent policy improvement are contingent on the fidelity of the differentiable simulator [27]. Furthermore, the learned value function for the terminal value estimation in the optimization objective for obtaining the exploratory policy is an imperfect estimator, especially in the early training stage. Consequently, while the data collected by the exploratory policy provides a powerful guiding signal on the whole, the advantage estimates may not be strictly greater than those from the primary data at every point in the training. To handle potentially suboptimal data from a corrupted $\pi_{\boldsymbol{\theta}^{\text{explore}}}$, our framework uses the value function as a safety filter. Such trajectories yield negative advantage estimates via GAE, allowing PPO objective to effectively down-weight or *reject* them, thus shielding the primary policy.

In general, the exploratory policy acts as a *scout* that identifies promising regions, while the primary policy learns from these curated trajectories. This reduces inefficient wandering, making learning focus the learning on demonstrably better behaviors and accelerating convergence.

V. Experiment Results

We aim to answer these two primary questions with our experiments: (1) Can our exploration augmentation mechanism help the training process access the higher-reward region in the state-action space and further improve the learning sample efficiency through that biased exploration? (2) Is the proposed method deployable in the real world? To this end, we evaluate on a series of comparative experiments on benchmark environments implemented in `MuJoCoPlayground` (MJP) [39] and a set of sim-to-real deployment experiments on a biped point-foot robot, LimX TRON. Training runs on an RTX 4090 GPU with an AMD EPYC 9354 CPU. We employed the JAX-based PPO implementation in `Brax` [17], and we implement our method on top of differentiable dynamics provided by `Brax` and `MuJoCo-XLA` (MJX) [40], [41]. We conducted preliminary hyperparameter searches of our method for each task and report the best settings.

A. Comparative Evaluation with Benchmark Experiment

We validate the effectiveness of our method against PPO with entropy bonus and PPO with RND exploration on eight benchmarks in MJP: four `dm_control_suite` tasks displayed in the top row of Fig. 4 and four locomotion tasks shown in the bottom row. And we provide the curve of SHAC method on the four `dm_control_suite` tasks as a reference for the performance of pure APG. The PPO part of its combination with RND and our method shares the same settings as baseline PPO, while our method's hyperparameters are listed in Table I. And the SHAC follows h of 32, λ of 0.99, N of 128 and γ of 0.95. Reported environment steps include both PPO and APG rollouts.

Fig. 4 illustrates the training curves across eight benchmark tasks, showing the progression of episodic average return against the number of environment steps. Our benchmark results exhibit consistently improved sample efficiency and better training stability against the PPO and PPO with the RND method across most of the tests, though the degree of improvement varies by task. A noteworthy exception is test `Go1Getup`, where a quadruped robot needs to learn the fall

TABLE I: Hyperparameters of our method in the experiments.

Environment	f	h	lr	γ	N	α	e
CartpoleBalance	1	4	3e-5	0.95	256	0.5	2
AcrobotSwingup	1	8	1e-4	0.95	256	0.5	2
FishSwim	1	4	3e-5	0.95	256	0.5	5
PointMass	1	4	3e-5	0.95	256	0.5	5
Go1Getup	5	32	3e-5	0.95	256	0.5	5
BarkourJoystick	1	4	3e-5	0.95	256	0.5	5
Go1FootStand	1	8	3e-5	0.95	256	0.5	5
T1JoystickFlatTerrain	1	4	3e-5	0.95	256	0.5	5
TRONLocomotion	1	4	3e-5	0.95	256	0.5	5

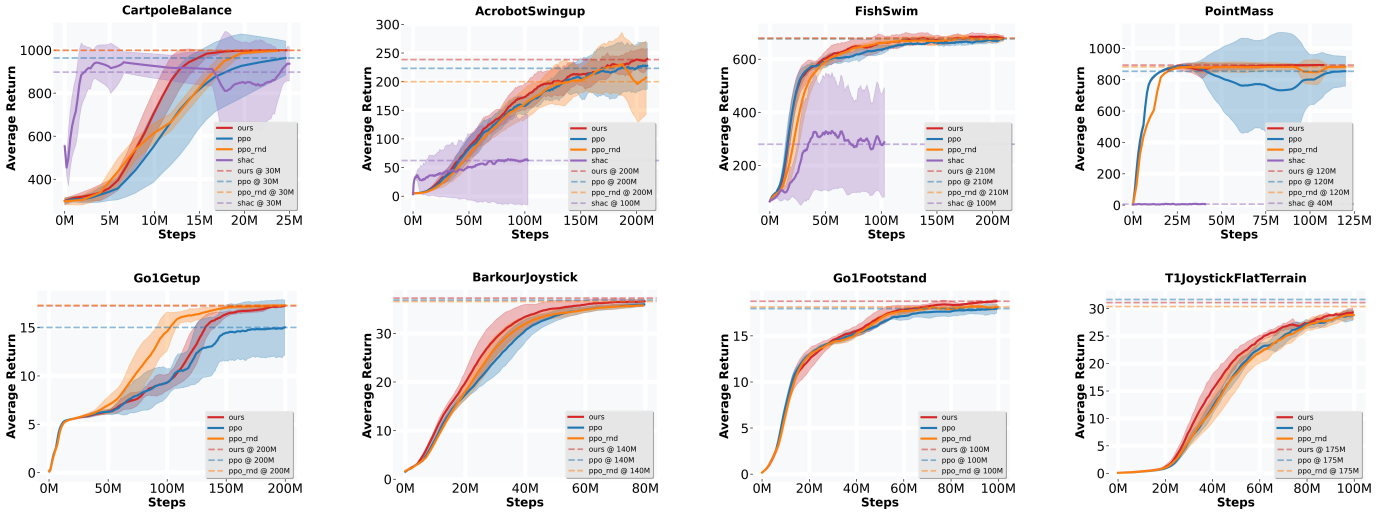

Fig. 4: Training curve of 8 benchmark tasks comparing the proposed method against the PPO baseline, the PPO with RND technique. The curves of SHAC serve as a reference. Solid lines and shaded regions depict the mean and standard deviation among the five trials, respectively. Across most benchmark tasks, our method achieves higher or matched asymptotic performance over the PPO baseline and PPO with RND technique, with improved sample efficiency and better training stability.

recovery from an arbitrary initial state, RND performs better than our method. We attribute this to the highly discontinuous contact dynamics. Under such task and dynamics, the initial analytical gradients can be noisy or misleading in our method and the undirected novelty-seeking of RND is proven to be more effective. This acceleration against the PPO with plain entropy maximization directly demonstrates the effectiveness of the proposed exploration guidance mechanism. And the comparison with PPO integrated with the novelty-based RND exploration technique shows that ours offers a slight superiority. Beyond the performance comparison, we think that our method is conceptually distinct from the novelty-based RND method, and the value of ours lies in this novel exploration mechanism, which utilizes the underlying dynamics priors of the system to augment the exploration.

Fig. 5 shows that the exploratory data yields consistently higher advantage, empirically validating IV.2 and confirming that our mechanism effectively guides the agent toward high-return regions. Furthermore, our method enhances training stability, evidenced by reduced variance across most tasks; notably, in `PointMass`, it maintains stable convergence while PPO suffers performance collapse. Finally, sensitivity analysis in Fig. 6 confirms high robustness to update frequency f, showing consistent performance across different f. Although the horizon length h affects gain magnitude, our method consistently outperforms the PPO baseline across a broad range.

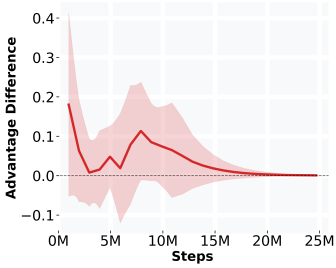

Fig. 5: Advantage difference between the data collected by the exploratory policy and that of the primary policy in `CartpoleBalance` task. Data collected with 5 trials and smoothed with a factor of 0.7.

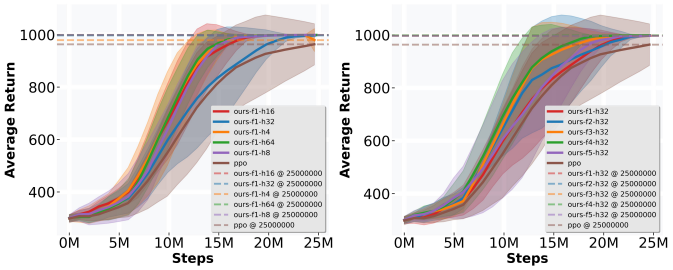

Fig. 6: Sensitivity analysis. Left: APG horizon length h; right: for APG update frequency f. The other hyperparameter follows the benchmark test except for $N = 64$.

B. Simulation Test and Sim-to-real Physical Validation

To validate practical viability, we trained a flat-terrain locomotion policy for the LimX TRON 6-DOF biped robot and conducted simulation and sim-to-real experiments. Specifically, we train velocity-tracking locomotion policies for LimX TRON using both our method and the baseline with reward design shown in Table III. And we conduct the simulation test to evaluate and compare our performance over the baseline. Moreover, we deploy our policy to a real bipedal robot. Fig. 8 shows the training curves: overall episodic return and specific task rewards defined in the first two rows of Table III. The episodic return demonstrates significantly improved sample efficiency, converging almost twice as fast as the baseline to reach similar asymptotic performance. The velocity tracking reward terms curves on the left show the same trend. This further verifies the effectiveness of our exploration augmentation mechanism for guiding the agent to explore high-reward regions. Fig. 9 shows the simulation test scene, and Fig. 10 compares linear and angular velocity tracking performance under varying velocity commands of our method against PPO, where training reward follows Table III. Our method achieves

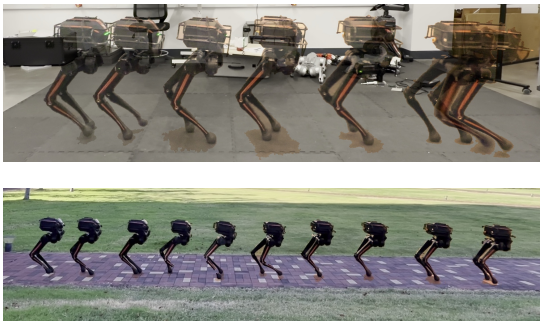

Fig. 7: Biped locomotion sim-to-real transfer experiment with TRON robot. Top: indoor experiment. Bottom: outdoor experiment.

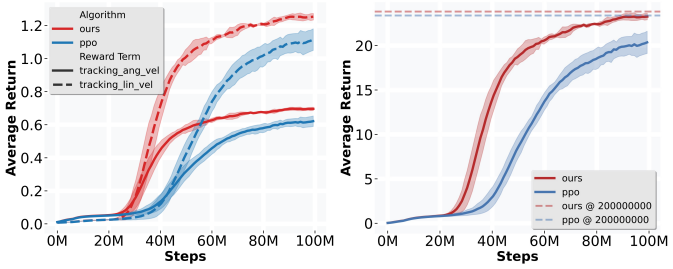

Fig. 8: TRON locomotion policy training curve. Left: angular and linear velocity tracking rewards. Right: overall episodic return.

comparable performance overall, while exhibiting lower tracking error and reduced variance in terms of the velocity tracking task, as displayed in Table II. We attribute improved tracking to APG-directed exploration, which biases the agent toward motion patterns that effectively regulate velocity. Furthermore, the differentiable task rewards, such as the velocity tracking reward in this case, dominate the analytical gradient and steer trajectories toward task-oriented behaviors that are aligned with the control objective, as indicated in Table II and Fig. 8. Fig. 7 shows the successful sim-to-real transfer of our

TABLE II: Velocity tracking performance in the simulation

	Baseline (PPO)	Ours
MSE of Lin. Vel. Tracking $(m/s)^2$	0.035707	0.028816 (↑ 19.30%)
Error Variances of Lin. Vel. $(m/s)^2$	0.034049	0.026701 (↑ 21.58%)
MSE of Ang. Vel. $(rad/s)^2$	0.023856	0.009260 (↑ 61.18%)
Error Variances of Ang. Vel. $(rad/s)^2$	0.017882	0.008821 (↑ 50.67%)

biped locomotion policy, demonstrating effective locomotion performance in both indoor and outdoor environments. This deployment serves as the ultimate validation of the proposed algorithm through the lens of practical viability and provides a sanity check for real-world use.

Fig. 9: Left: 6-DOFs biped robot, LimX TRON. Right: TRON biped locomotion policy test scene with the proposed method.

VI. Discussion

As mentioned in Section I, model-free RL accommodates arbitrary rewards but is data-expensive, while FoG methods

TABLE III: Reward terms for TRON velocity tracking training.

Reward Term	Equation	Weight
Lin. vel. track.	$\exp(-4\lVert v_{xy}^{\text{cmd}} - v_{xy}\rVert_2^2)$	1.5
Ang. vel. track.	$\exp(-4(\omega_z^{\text{cmd}} - \omega_z)^2)$	0.7
Lin. vel. (z)	v_z^2	-0.5
Ang. vel. (xy)	$\lVert\omega_{xy}\rVert_2^2$	-0.05
Orientation	$\lVert g_{xy}\rVert_2^2$	-10.0
Base height	$(h - h_{\text{target}})^2$	-2.0
Joint acceleration	$\lVert\ddot{q}\rVert_2^2$	-2.5×10^{-7}
Torques	$\lVert\tau\rVert_1$	-8.0×10^{-5}
Action rate	$\lVert a_t - a_{t-1}\rVert_2^2$	-0.01
Feet distance	$\text{clip}(d_{\min} - d_{\text{feet}}, 0, 1)$	-50.0
Feet landing vel.	$\sum \text{landing} \cdot v_{z,i}^2$	-0.15
Feet air time	$\sum \text{clip}(t_{\text{air}} - t_{\min}, 0, t_{\max})$	2.0
Feet slip	$\sum \lVert v_{\text{body}}\rVert \cdot \text{contact}_i$	-0.25
Feet phase	$\exp(-\frac{\lVert h_{\text{feet}} - r_z(\phi)\rVert_2^2}{0.01})$	1.0
Joint dev. knee	$\sum \lvert q_{\text{knee}} - q_{\text{knee}}^{\text{default}}\rvert$	-0.05
Joint dev. hip	$\sum \lvert q_{\text{hip}} - q_{\text{hip}}^{\text{default}}\rvert \cdot \lvert\text{cmd}_y\rvert$	-0.15
DOF pos limits	$\sum \text{violations}(q, q_{\text{lim}})$	-2.0
Pose	$\sum (q - q_{\text{default}})^2 \cdot w$	-1.0
Termination	done	-1.0

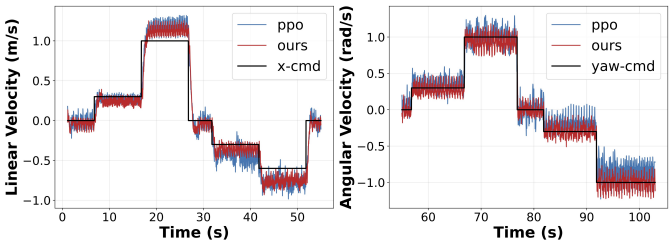

Fig. 10: Biped velocity tracking performance in simulation test.

are sample-efficient but limited to differentiable tasks. Our approach bridges this gap by utilizing analytical gradients solely for exploration guidance. This decoupling potentially enables a hybrid reward structure, where the exploratory policy optimizes dense, differentiable task objectives, such as velocity tracking, while the primary policy optimizes the full reward function, including non-differentiable constraints like gait regularization. Crucially, the model-free update acts as a filter, rejecting exploratory trajectories that violate these constraints via advantage estimation. This design allows our method to exploit dynamics priors for efficiency while preserving compatibility with arbitrary, non-differentiable reward formulations. Furthermore, while demonstrated here with PPO, a widely adopted RL algorithm for robotic control, our framework is fundamentally a modular data augmentation engine. Since the generated exploratory trajectories are decoupled from the primary policy update logic, this paradigm has the potential to be extended to any actor-critic host RL algorithm, such as A2C or TRPO. However, in sparse reward settings where short-horizon gradients vanish, our method leverages the terminal value signal from a well-learned critic. Admittedly, during early training with extremely sparse rewards and an uninformative critic, our method may degenerate to PPO. Reported in [22], [27], analytical gradients in contact-rich tasks often exhibit empirical bias. A dynamics-based curriculum can mitigate this by starting with smoothed dynamics to provide reliable early guidance, before progressively increasing simulation realism. Finally, while offering better wall-clock efficiency in simpler dynamics, our method may currently underperform PPO in complex contact-rich tasks due

to the differentiable simulator's implementation overhead.

VII. Conclusion and Future Work

We proposed an exploration augmentation framework utilizing analytical policy gradients to guide on-policy RL. Theoretical analysis confirms that this strategy yields a positive-biased advantage, driving significantly improved sample efficiency. Extensive benchmarks and sim-to-real deployment on a bipedal robot demonstrate consistent gains in training stability and physical viability, validating the method's effectiveness across diverse tasks. Our approach uniquely combines gradient-guided exploration with flexible, non-differentiable rewards, though it remains sensitive to sparse reward settings and simulator fidelity. Future directions include adaptively tuning the data merging ratio based on gradient reliability and integrating learned dynamics models to extend applicability to complex, black-box environments.

References

[1] J. Hwangbo *et al.*, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[3] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proc. of the 5th Conf. on Rob. Learn.*, vol. 164. PMLR, 08–11 Nov 2022, pp. 91–100.

[4] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "Deepmimic: example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph.*, vol. 37, no. 4, Jul. 2018.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[6] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, no. 3–4, p. 229–256, May 1992.

[7] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. of the 35th Int. Conf. Mach. Learn. (ICML)*, vol. 80. PMLR, 10–15 Jul 2018, pp. 1861–1870.

[8] M. Plappert *et al.*, "Parameter space noise for exploration," 2018. [Online]. Available: https://arxiv.org/abs/1706.01905

[9] T. Kurutach, I. Clavera, Y. Duan, A. Tamar, and P. Abbeel, "Model-ensemble trust-region policy optimization," 2018. [Online]. Available: https://arxiv.org/abs/1802.10592

[10] T. Yu, A. Kumar, R. Rafailov, A. Rajeswaran, S. Levine, and C. Finn, "Combo: Conservative offline model-based policy optimization," in *Adv. Neural Inf. Process. Syst.*, vol. 34. Curran Associates, Inc., 2021, pp. 28 954–28 967.

[11] V. Feinberg, A. Wan, I. Stoica, M. I. Jordan, J. E. Gonzalez, and S. Levine, "Model-based value estimation for efficient model-free reinforcement learning," 2018. [Online]. Available: https://arxiv.org/abs/1803.00101

[12] M. P. Deisenroth and C. E. Rasmussen, "Pilco: a model-based and data-efficient approach to policy search," in *Proc. of the 28th Int. Conf. Mach. Learn. (ICML)*. Madison, WI, USA: Omnipress, 2011, p. 465–472.

[13] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. of the 23rd National Conference on Artificial Intelligence - Volume 3*, ser. AAAI'08. AAAI Press, 2008, p. 1433–1438.

[14] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, "Exploration by random network distillation," 2018. [Online]. Available: https://arxiv.org/abs/1810.12894

[15] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proc. of the 34th Int. Conf. Mach. Learn. (ICML)*, vol. 70. JMLR.org, 2017, p. 2778–2787.

[16] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. D. Turck, and P. Abbeel, "Vime: Variational information maximizing exploration," 2017. [Online]. Available: https://arxiv.org/abs/1605.09674

[17] D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem, "Brax - a differentiable physics engine for large scale rigid body simulation," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, vol. 1, 2021.

[18] M. C. Mozer, *A focused backpropagation algorithm for temporal pattern recognition*. USA: L. Erlbaum Associates Inc., 1995, p. 137–169.

[19] L. Metz, C. D. Freeman, S. S. Schoenholz, and T. Kachman, "Gradients are not all you need," 2022. [Online]. Available: https://arxiv.org/abs/2111.05803

[20] Y. Song, S. b. Kim, and D. Scaramuzza, "Learning quadruped locomotion using differentiable simulation," in *Proc. of the 8th Conf. on Rob. Learn.*, vol. 270. PMLR, 06–09 Nov 2025, pp. 258–271.

[21] J. Y. Luo, Y. Song, V. Klemm, F. Shi, D. Scaramuzza, and M. Hutter, "Residual policy learning for perceptive quadruped control using differentiable simulation," in *2025 IEEE Int. Conf. Robot. Autom. (ICRA)*, 2025, pp. 1–8.

[22] C. Schwarke *et al.*, "Learning deployable locomotion control via differentiable simulation," in *Proc. of the 9th Conf. Rob. Learn.*, vol. 305. PMLR, 27–30 Sep 2025, pp. 3665–3684.

[23] Y. Zhang, Y. Hu, Y. Song, D. Zou, and W. Lin, "Learning vision-based agile flight via differentiable physics," *Nat. Mach. Intell.*, vol. 7, no. 6, pp. 954–966, 2025.

[24] J. Xu *et al.*, "Accelerated policy learning with parallel differentiable simulation," 2022. [Online]. Available: https://arxiv.org/abs/2204.07137

[25] I. Clavera, V. Fu, and P. Abbeel, "Model-augmented actor-critic: Backpropagating through paths," 2020. [Online]. Available: https://arxiv.org/abs/2005.08068

[26] Y. Song, "Learning robot control: from reinforcement learning to differentiable simulation," Ph.D. dissertation, Dissertation, University of Zurich, Sep 2024.

[27] H. J. Suh, M. Simchowitz, K. Zhang, and R. Tedrake, "Do differentiable simulators give better policy gradients?" in *Proc. of the 39th Int. Conf. Mach. Learn. (ICML)*, vol. 162. PMLR, 17–23 Jul 2022, pp. 20 668–20 696.

[28] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. of the 35th Int. Conf. Mach. Learn. (ICML)*, vol. 80. PMLR, 10–15 Jul 2018, pp. 1587–1596.

[29] X. B. Peng, A. Kumar, G. Zhang, and S. Levine, "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning," 2019. [Online]. Available: https://arxiv.org/abs/1910.00177

[30] R. S. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *SIGART Bull.*, vol. 2, no. 4, p. 160–163, Jul. 1991.

[31] L. Kuvayev and R. S. Sutton, "Model-based reinforcement learning with an approximate, learned model," in *Proc. of the ninth Yale workshop on adaptive and learning systems*, 1996, pp. 101–105.

[32] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," in *Adv. Neural Inf. Process. Syst.*, vol. 32. Curran Associates, Inc., 2019.

[33] A. Piché, V. Thomas, C. Ibrahim, Y. Bengio, and C. Pal, "Probabilistic planning with sequential monte carlo methods," in *Int. Conf. on Learn. Represent. (ICLR)*, 2019. [Online]. Available: https://openreview.net/forum?id=ByetGn0cYX

[34] K. Chua, R. Calandra, R. McAllister, and S. Levine, "Deep reinforcement learning in a handful of trials using probabilistic dynamics models," in *Proc. of the 32nd Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates Inc., 2018, p. 4759–4770.

[35] Y.-L. Qiao, J. Liang, V. Koltun, and M. C. Lin, "Efficient differentiable simulation of articulated bodies," in *Proc. of the 38th Int. Conf. Mach. Learn. (ICML)*, vol. 139. PMLR, 18–24 Jul 2021, pp. 8661–8671.

[36] Y. D. Zhong, J. Han, and G. O. Brikis, "Differentiable physics simulations with contacts: Do they have correct gradients w.r.t. position, velocity and control?" 2022. [Online]. Available: https://arxiv.org/abs/2207.05060

[37] S. Kakade and J. Langford, "Approximately optimal approximate reinforcement learning," in *Proc. of the nineteenth Int. Conf. Mach. Learn. (ICML)*, 2002, pp. 267–274.

[38] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. of the 32nd Int. Conf. Mach. Learn. (ICML)*, vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1889–1897.

[39] K. Zakka *et al.*, "Demonstrating MuJoCo Playground," in *Proc. Robot.: Sci. Syst. (RSS)*, LosAngeles, CA, USA, June 2025.

[40] MuJoCo Team, "MuJoCo XLA (MJX) - MuJoCo Documentation," https://mujoco.readthedocs.io/en/latest/mjx.html, 2024, accessed: 2025-08-18.

[41] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2012, pp. 5026–5033.