A Multi-Fidelity Control Variate Approach for Policy Gradient Estimation

Anonymous authors
Paper under double-blind review

Abstract

Many reinforcement learning (RL) algorithms are impractical for deployment in operational systems or for training with computationally expensive high-fidelity simulations, as they require large amounts of data. Meanwhile, low-fidelity simulators—such as reduced-order models, heuristic reward functions, or generative world models—can cheaply provide useful data for RL training, even if they are too coarse for direct sim-to-real transfer. We propose *multi-fidelity policy gradients (MFPGs)*, an RL framework that mixes a small amount of data from the target environment with a control variate formed from a large volume of low-fidelity simulation data to construct an unbiased, variance-reduced estimator for on-policy policy gradients. We instantiate the framework by developing a practical, multi-fidelity variant of the classical REINFORCE algorithm. We show that under standard assumptions, the MFPG estimator guarantees asymptotic convergence of multi-fidelity REINFORCE to locally optimal policies in the target environment, and achieves faster finite-sample convergence rates compared to training with high-fidelity data alone. Empirically, we evaluate the MFPG algorithm across a suite of simulated robotics benchmark tasks in scenarios with limited high-fidelity data but abundant off-dynamics, low-fidelity data. With mild to moderate dynamics gaps, MFPG reliably improves the median performance over a standard baseline trained with only high-fidelity data, matching the performance of leading multi-fidelity baselines despite its simplicity and minimal tuning overhead. Under large dynamics gaps, MFPG demonstrates the strongest robustness among the evaluated multi-fidelity approaches. An additional experiment shows that MFPG can remain effective even when low-fidelity environments exhibit reward misspecification. Thus, MFPG not only offers a novel paradigm for efficient sim-to-real transfer, but also provides a principled approach to managing the trade-off between policy performance and data collection costs.

1 Introduction

Reinforcement learning (RL) algorithms offer significant capabilities in systems that work with unknown, or difficult-to-specify, dynamics and objectives. The flexibility and performance of RL algorithms have led to their adoption in applications as diverse as controlling plasma configurations in nuclear fusion reactors (Degrave et al., 2022), piloting high-speed aerial vehicles (Kaufmann et al., 2023), training reasoning capabilities in large language models (Shao et al., 2024), and searching large design spaces for the automated discovery of new molecules (Bengio et al., 2021; Ghugare et al., 2023). However, in many such applications, datasets must often be gathered from operational systems or from *high-fidelity* simulations. This requirement acts as a significant barrier to the development and deployment of RL policies: excessive interactions with operational systems are often either infeasible or unsafe, and generating simulated datasets for RL can be prohibitively expensive unless the simulations are both cheap to run and carefully designed to minimize the sim-to-real gap.

On the other hand, *low-fidelity* simulation tools capable of cheaply generating large volumes of data are often available. For example, reduced-order models, linearized dynamics, heuristic reward functions, and generative world models all output useful information for RL, even when approximations of the target dynamics and rewards are very coarse.

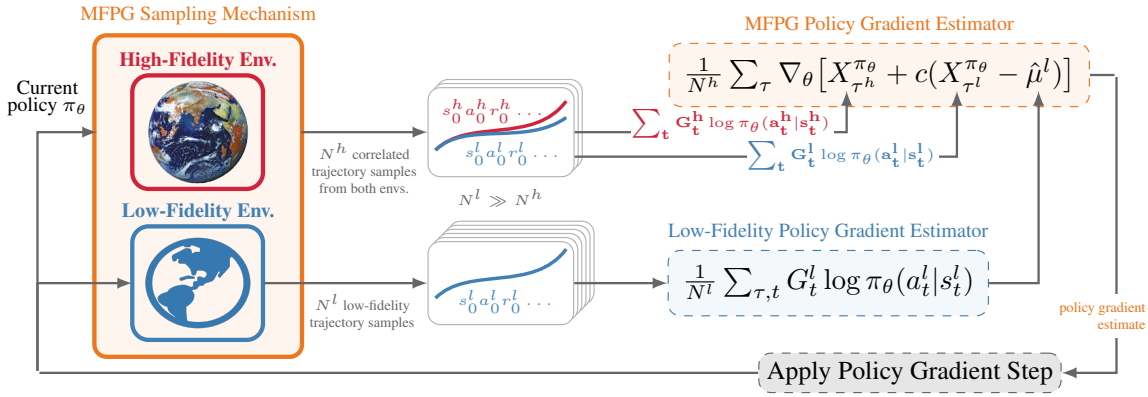

Figure 1: The proposed multi-fidelity policy gradient (MFPG) framework. At each policy update step, MFPG combines a small amount of data from the target (high-fidelity) environment with a large volume of low-fidelity simulation data, thereby forming an unbiased, reduced-variance estimator for the policy gradient.

Towards enabling the training and deployment of RL policies when expensive, high-fidelity samples are scarce, we develop a novel *multi-fidelity* RL algorithm. The proposed algorithm mixes data from the target environment with data generated by lower-fidelity simulations to improve the sample efficiency of high-fidelity data. The proposed framework not only offers a novel paradigm for sim-to-real transfer, but also provides a principled approach to managing the trade-off between policy performance and data collection costs.

More specifically, we present *multi-fidelity policy gradients (MFPGs)*, an RL framework that mixes a small amount of data from the target environment with a control variate formed from a large volume of low-fidelity data to construct an unbiased, variance-reduced estimator for on-policy policy gradients. We further propose a practical algorithm that instantiates the MFPG estimator with the classical on-policy algorithm REINFORCE (Williams, 1992).

Figure 1 illustrates the proposed approach. At each policy update step, the algorithm begins by sampling a relatively small number of trajectories from the target environment, which may correspond to real-world hardware or to a high-fidelity simulation. We then propose a method for sampling trajectories from the low-fidelity environments such that the resulting action likelihoods are highly correlated with those from the high-fidelity trajectories. This method hinges on a nuanced approach to action sampling and is critical to the success of our approach. Next, the algorithm uses the low-fidelity environments to sample a much larger quantity of uncorrelated trajectories, which it uses alongside the previously sampled trajectories to compute an unbiased estimate of the policy gradient to be applied. So long as the random variables corresponding to the policy gradient updates between the high and low-fidelity environments are correlated, the approach is guaranteed to reduce the variance of the policy gradient estimates.

Theoretically, we analyze the proposed MFPG estimator for the REINFORCE algorithm and prove that, under standard assumptions, the MFPG estimator guarantees asymptotic convergence to a locally optimal policy (satisfying first-order optimality conditions) in the *target* (high-fidelity) environment. Moreover, when high- and low-fidelity samples exhibit nonzero correlation, MFPG achieves a faster finite-sample convergence rate than using high-fidelity samples alone.

Empirically, we evaluate the proposed algorithm on a suite of benchmark robotic control tasks where the high- and low-fidelity environments differ in transition dynamics, and additionally on a task with a misspecified reward function in the low-fidelity environment. In high-fidelity sample-scarce regimes, the results reveal three key insights: i) When the dynamics gap is mild, MFPG can substantially reduce the variance of policy gradient estimates compared to standard variance reduction via state-value baseline subtraction. ii) With mild to moderate dynamics gaps, MFPG reliably outperforms the baseline that only uses high-fidelity data in median performance, matching the strongest multi-fidelity methods despite its algorithmic simplicity and minimal tuning overhead. In more challenging scenarios with large dynamics gaps, MFPG is the most

robust among all evaluated multi-fidelity methods, incurring the fewest performance collapses. iii) MFPG can remain effective even in the presence of reward misspecification in the low-fidelity samples.

2 Related Work

This section provides a concise overview of three lines of related work: (i) variance reduction in RL via control variates; (ii) multi-fidelity RL and fine-tuning using high-fidelity data for sim-to-real transfer; and (iii) off-dynamics RL.

Variance Reduction in RL via control variates. The method of control variates (CVs) is commonly used for variance reduction in Monte Carlo estimation (Owen, 2013). In RL, particularly in policy gradient (PG) methods (Williams, 1992), there is a long history of using CVs to reduce variance and accelerating learning, e.g., subtracting constant reward baselines (Sutton, 1984; Williams, 1988; Dayan, 1991; Williams, 1992) or state value functions (Weaver & Tao, 2001; Greensmith et al., 2004; Peters & Schaal, 2006; Zhao et al., 2011) from Monte Carlo returns. Even though state values are known to not be optimal for variance reduction (Greensmith et al., 2004), they are also central to modern algorithms, e.g., actor-critic methods (Schulman et al., 2017). More recently, additional CVs baselines have been studied, such as vector-form CVs (Zhong et al., 2021), multi-step variance reduction along trajectories (Pankov, 2018; Cheng et al., 2020) and their more general form (Huang & Jiang, 2020), and state-action dependent baselines (Gu et al., 2017; Liu et al., 2018; Grathwohl et al., 2018; Wu et al., 2018), which offer better variance reduction under certain conditions (Tucker et al., 2018). The literature on using CVs for policy gradients almost exclusively focuses on single-environment settings. In this work, we consider the problem of fusing data from multi-fidelity environments to construct low-variance estimators, cf. Section 4. Moreover, the proposed multi-fidelity approach can also be readily combined with existing single-fidelity CVs for further improved performance.

To the best of our knowledge, the only existing work leveraging multi-fidelity CVs for RL is (Khairy & Balaprakash, 2024). Our approach differs in two main ways. First, we propose a multi-fidelity PG algorithm for Markov decision processes (MDPs) with either continuous or discrete state and action spaces, whereas Khairy & Balaprakash (2024) estimate state-action values in tabular MDPs without function approximation. Second, we propose a novel policy reparameterization trick that enables the sampling of correlated trajectories across multi-fidelity environments. Crucially, this sampling scheme improves the variance reduction of the proposed MFPG algorithm, and it directly supports continuous settings without restricting the MDPs. By contrast, Khairy & Balaprakash (2024) require action sequences to be matched across multi-fidelity environments, which requires additional assumptions about the MDP structure and transition dynamics.

Multi-fidelity RL and fine-tuning using high-fidelity data. Multi-fidelity methods for optimization, uncertainty propagation, and inference are well-studied in computational science and engineering, where it is often the case that multiple models for a problem of interest are available (Peherstorfer et al., 2018). Such methods provide principled statistical techniques to manage the computational costs of Monte Carlo simulations, without introducing unwanted biases. However, the adoption of such techniques by RL algorithms is significantly more limited. Two bodies of literature which are closely related to this work are multi-fidelity RL (Cutler et al., 2015) and fine-tuning RL policies using limited target-domain data, e.g., in sim-to-real transfer (Smith et al., 2022). Multi-fidelity RL aims to design *training curricula*, i.e., decide when to train in each simulation fidelities, in order to train highly-performant policies for the highest-fidelity environment with minimum simulation costs. Typically, training begins in the lowest-fidelity simulators, with proposed mechanisms to decide when to transition to other fidelities based on estimated uncertainty (Suryan et al., 2017; Cutler et al., 2015; Agrawal & McComb, 2024; Ryou et al., 2024; Bhola et al., 2023) in predicted actions, state-action values, dynamics, rewards, and/or information gain (Marco et al., 2017). Similarly, in sim-to-real and transfer learning settings (Taylor & Stone, 2009; Zhao et al., 2020; Tang et al., 2024; Da et al., 2025; Niu et al., 2024), the objective is to train a highly-performant policy for the real world using simulation data, sometimes supplemented with a small amount of real-world data. Many approaches leverage models or value functions trained in simulation (source domain) to bootstrap fine-tuning in the real world (target domain) (Rusu et al., 2017; Arndt et al., 2020; Taylor et al., 2007; Smith et al., 2022), guide real-world exploration (Yin et al., 2025), or bidirectionally align state distributions between simulation and

real-world agents to improve sample efficiency. Real-world data can also be used to refine the simulator itself (Abbeel et al., 2006; Chebotar et al., 2019; Ramos et al., 2019). The works above introduce new training paradigms and exploration strategies with data from different domains. In contrast, our approach proposes a new policy gradient estimator that incorporates cross-domain data. The proposed approach can be seamlessly integrated into existing paradigms that involve fine-tuning within target domains.

Off-dynamics RL. A particularly important setting is when multi-fidelity environments exhibit mismatched transition dynamics. Classical approaches such as system identification (Ljung, 1998) and domain randomization (Tobin et al., 2017) address this problem from a dynamics modeling perspective, but typically require either access to a high-fidelity dynamics model or a pre-specified distribution over dynamics parameters. Other works (Desai et al., 2020b;a) instead learn action transformations to compensate for transitions produced by low-fidelity dynamics.

Another line of work tackles the off-dynamics RL problem from a policy adaptation perspective, i.e., modifying policy learning with samples from multiple domains. DARC (Eysenbach et al.) estimates importance weights for low-fidelity transitions—via a pair of learned classifiers modeling the likelihood ratio under high- and low-fidelity dynamics—and uses these weights to augment the low-fidelity rewards. Guo et al. (2024) extend this idea with generative adversarial imitation learning. Similar in spirit to DARC (Eysenbach et al.), more recent works estimate the dynamics mismatch by comparing observed low-fidelity transitions with a corresponding high-fidelity prediction—either through explicit target dynamics models to evaluate state-value consistency (Xu et al., 2023) or through latent representation learning (Lyu et al., 2024a). The estimated dynamics gap is then used to filter out unrealistic low-fidelity transitions (Xu et al., 2023) or to augment the low-fidelity rewards by penalizing less likely transitions (Lyu et al., 2024a). A related line of work extends these ideas to *offline* RL, where high-fidelity data come from a fixed dataset (Niu et al., 2022; Liu et al., 2022; Niu et al., 2025). These methods use learned domain classifiers—originally proposed in DARC (Eysenbach et al.)—to regularize policy learning with low-fidelity samples.

In this work, we focus on the *online*, on-policy setting, where the agent has limited access to interact with the high-fidelity (target) environment but can collect abundant low-fidelity samples for each policy update. We benchmark our proposed MFPG algorithm against DARC (Eysenbach et al.) and PAR (Lyu et al., 2024a). A key distinction between MFPG and most existing off-dynamics RL methods lies in their fundamental design philosophy. Prior approaches often rely on regularizing low-fidelity samples for policy learning (Eysenbach et al.; Xu et al., 2023; Lyu et al., 2024a). In contrast, MFPG explicitly *grounds* policy learning in high-fidelity samples, leveraging low-fidelity data solely as a variance-reduction mechanism. This difference has important theoretical implications. Existing methods (Eysenbach et al.; Xu et al., 2023; Lyu et al., 2024a) bound the global *suboptimality* of policies deployed in the high-fidelity (target) environment; the bounds can degrade as the dynamics gap grows—for example, when discrepancies in value functions (Xu et al., 2023) or learned representations (Lyu et al., 2024a) become more pronounced. DARC (Eysenbach et al.) further assumes that policies optimal in the target domain remain near-optimal in the source domain. By contrast, MFPG requires no such assumption: low-fidelity samples are used exclusively for variance reduction—our asymptotic guarantees ensure that the learned policy is locally *optimal* (up to first order) in the high-fidelity environment, regardless of the dynamics mismatch. This design principle enables us to establish faster convergence rates for MFPG with REINFORCE when the correlation between low- and high-fidelity samples is non-zero. Empirically, MFPG demonstrates greater robustness than DARC and PAR. Moreover, unlike prior methods designed specifically for dynamics mismatch, MFPG can also extend to settings with reward misspecification, as illustrated in Section 6.4.

3 Preliminaries

Our objective is to develop RL algorithms capable of leveraging data generated by multiple distinct environments in order to efficiently learn a policy that achieves high performance in a target environment of interest.

Modeling the multi-fidelity environments. We begin by considering a *high-fidelity* environment and a *low-fidelity* environment, each modeled by an MDP. In particular, the high-fidelity environment is an

MDP $\mathcal{M}^h = (S, A, \Delta_{s_I}, \gamma, p^h, R^h)$, which we assume represents either an accurate simulator of the target environment, or the target environment itself. Here, S is the set of environment states, A is its set of actions, Δ_{s_I} is an initial distribution over states, $\gamma \in [0, 1]$ is a discount factor, $p^h(s'|s, a)$ is the probability of transitioning to state s' from state s under action a, and $r \sim R^h(s, a, s')$ defines the probability of observing a particular reward under a given state-action-state triplet. Similarly, the low-fidelity environment is an MDP $\mathcal{M}^l = (S, A, \Delta_{s_I}, \gamma, p^l, R^l)$, whose transition dynamics p^l and reward function R^l differ from those of $\mathcal{M}^h$, and do not necessarily accurately represent the target environment. We assume that the cost of generating sample trajectories in $\mathcal{M}^h$ is significantly higher than that of generating trajectories in $\mathcal{M}^l$, as may be the case if $\mathcal{M}^h$ were to represent real-world hardware while $\mathcal{M}^l$ were to represent cheap simulations.

Our objective is to learn a stochastic policy $\pi_\theta(a|s)$, parameterized by $\theta \in \mathbb{R}^d$, that achieves a high expected total reward in the high-fidelity environment $\mathcal{M}^h$. Fixing a policy π_θ defines a distribution over trajectories in both $\mathcal{M}^h$ and $\mathcal{M}^l$. We denote trajectories in $\mathcal{M}^h$ by $\tau^h = s_0^h, a_0^h, r_0^h, s_1^h, a_1^h, r_1^h, \ldots, s_T^h$, where $s_0^h \sim \Delta_{s_I}$, $a_t^h \sim \pi_\theta(\cdot|s_t^h)$, $s_{t+1}^h \sim p^h(\cdot|s_t^h, a_t^h)$, and $r_t^h \sim R^h(s_t^h, a_t^h, s_{t+1}^h)$. Similarly, we use τ^l to denote trajectories sampled in $\mathcal{M}^l$.

We remark that the proposed method may be extended to integrate data from an arbitrary number of environments, each of which simulates the target environment with a different level of fidelity. One may also define $\mathcal{M}^h$ and $\mathcal{M}^l$ to have different sets of states and actions, as well as different initial distributions and discount factors. Furthermore, the framework can be extended to infinite-horizon cases. However, neither of these extensions significantly enhances the primary conceptual, theoretical, and experimental contributions of this work, and so we leave them to future research.

Policy gradients. PG algorithms aim to maximize the performance measure $J(\theta) := \mathbb{E}_{\tau\sim\mathcal{M}(\pi_\theta)}[R(\tau)]$—the expected total reward along trajectories τ sampled from environment $\mathcal{M}$ under policy π_θ. They do so by using stochastic estimates of the policy gradient $\nabla_\theta J(\theta)$ to perform gradient ascent directly on the policy parameters (Sutton & Barto, 2018). For example, the REINFORCE algorithm (Williams, 1988; 1992) uses Monte Carlo estimates of $\mathbb{E}_{\tau\sim\mathcal{M}(\pi_\theta)}[\nabla_\theta X_\tau^{\pi_\theta}]$ to estimate the policy gradient, where $X_\tau^{\pi_\theta} := \frac{1}{T}\sum_{t=0}^{T-1} G_t \log \pi_\theta(a_t|s_t)$ is a random variable defining the contribution of each trajectory τ to the overall policy gradient. Here, a_t, s_t, and $G_t = \sum_{k=t}^{T-1} \gamma^{k-t} r_k$, denote the selected action, the state, and the reward-to-go at time t in trajectory τ, respectively.

Different policy gradient algorithms use different expressions for $X_\tau^{\pi_\theta}$ when estimating the policy gradient (e.g., G_t may be replaced with an advantage estimate, or $X_\tau^{\pi_\theta}$ may be entirely replaced by a surrogate per-trajectory gradient estimate, as is done in PPO (Schulman et al., 2017)). However, we note that the overall structure of many on-policy algorithms remains the same: use the current policy to sample trajectories in the environment; use the rewards, actions, and states along these trajectories to compute a variable of interest $X_\tau^{\pi_\theta}$; finally, average the gradients of these sampled random variables to estimate $\nabla_\theta J(\theta)$.

4 Multi-Fidelity Policy Gradients

This section introduces our MFPG framework, a mechanism to draw correlated trajectories from multi-fidelity environments, and the corresponding multi-fidelity REINFORCE algorithm.

Multi-fidelity policy gradient estimators via control variates. Because our objective is to optimize the policy performance in the high-fidelity environment $\mathcal{M}^h$, during each step of the policy gradient algorithm we must estimate $\nabla_\theta \mathbb{E}_{\tau^h\sim\mathcal{M}^h(\pi_\theta)}[X_{\tau^h}^{\pi_\theta}]$ from a potentially limited number N^h of sampled high-fidelity trajectories τ^h. In data-scarce settings, existing policy gradient methods can face the challenge of high variance of the gradient estimates (Greensmith et al., 2004). We aim to reduce the estimation variance for the PGs. In this work, we assume that we may also sample a relatively large number $N^l \gg N^h$ of trajectories τ^l from the low-fidelity environment $\mathcal{M}^l$. We use these low-fidelity samples to construct a so-called *control variate* $X_{\tau^l}^{\pi_\theta}$—a correlated auxiliary random variable whose expected value $\mu^l := \mathbb{E}_{\tau^l\sim\mathcal{M}^l(\pi_\theta)}[X_{\tau^l}^{\pi_\theta}]$ is known. We then use the *control variates technique* (Nelson, 1987) to construct an unbiased, reduced-variance estimator for $\nabla_\theta \mathbb{E}_{\tau^h\sim\mathcal{M}^h(\pi_\theta)}[X_{\tau^h}^{\pi_\theta}]$.

Specifically, we construct a new random variable $Z^{\pi_\theta} := X^{\pi_\theta}_{\tau^h} + c(X^{\pi_\theta}_{\tau^l} - \mu^l)$ with coefficient $c \in \mathbb{R}$. We estimate μ^l using the N^l low-fidelity trajectory samples, and we provide a method to jointly sample correlated values $X^{\pi_\theta}_{\tau^h}$ and $X^{\pi_\theta}_{\tau^l}$ (described below). By construction, this new random variable has the property that $\mathbb{E}[Z^{\pi_\theta}] = \mathbb{E}[X^{\pi_\theta}_{\tau^h}]$. Furthermore, depending on the value of c, it may have a significantly reduced variance. In particular, by choosing

$$c^* = -\frac{\mathrm{Cov}(X^{\pi_\theta}_{\tau^h}, X^{\pi_\theta}_{\tau^l})}{\mathrm{Var}(X^{\pi_\theta}_{\tau^l})} = -\rho(X^{\pi_\theta}_{\tau^h}, X^{\pi_\theta}_{\tau^l})\frac{\sqrt{\mathrm{Var}(X^{\pi_\theta}_{\tau^h})}}{\sqrt{\mathrm{Var}(X^{\pi_\theta}_{\tau^l})}} \implies \mathrm{Var}(Z^{\pi_\theta}) = (1-\rho^2(X^{\pi_\theta}_{\tau^h}, X^{\pi_\theta}_{\tau^l}))\mathrm{Var}(X^{\pi_\theta}_{\tau^h}). \quad (1)$$

Note that this choice of c^* is a known closed-form solution minimizing the variance for Z^{π_θ} (Owen, 2013, Ch.8), and $\rho(\cdot,\cdot)$ is the Pearson correlation coefficient between the two random variables.

In practice, we estimate c^* using the sampled trajectories from $\mathcal{M}^h$ and $\mathcal{M}^l$, reusing samples from previous gradient steps by maintaining moving-average statistics. Specifically, we maintain moving averages of the correlation $\hat{\rho}(X^{\pi_\theta}_{\tau^h}, X^{\pi_\theta}_{\tau^l})$ and standard deviations $\sqrt{\widehat{\mathrm{Var}}(X^{\pi_\theta}_{\tau^h})}, \sqrt{\widehat{\mathrm{Var}}(X^{\pi_\theta}_{\tau^l})}$. Upon each policy update, we compute

$$\hat{c}^* = -\Big(\eta_{\mathrm{ma}}\hat{\rho}(X^{\pi_\theta}_{\tau^h}, X^{\pi_\theta}_{\tau^l})_{\mathrm{old}} + (1-\eta_{\mathrm{ma}})\hat{\rho}(X^{\pi_\theta}_{\tau^h}, X^{\pi_\theta}_{\tau^l})_{\mathrm{new}}\Big)\frac{\eta_{\mathrm{ma}}\sqrt{\widehat{\mathrm{Var}}(X^{\pi_\theta}_{\tau^h})_{\mathrm{old}}} + (1-\eta_{\mathrm{ma}})\sqrt{\widehat{\mathrm{Var}}(X^{\pi_\theta}_{\tau^h})_{\mathrm{new}}}}{\eta_{\mathrm{ma}}\sqrt{\widehat{\mathrm{Var}}(X^{\pi_\theta}_{\tau^l})_{\mathrm{old}}} + (1-\eta_{\mathrm{ma}})\sqrt{\widehat{\mathrm{Var}}(X^{\pi_\theta}_{\tau^l})_{\mathrm{new}}}}. \quad (2)$$

Here, the subscripts "old" and "new" denote the statistics carried over from the previous iteration and those estimated from the current batch, respectively. The hyperparameter $\eta_{\mathrm{ma}} \in (0,1)$ controls the exponential moving-average weight.

At every policy gradient step, the proposed MFPG framework thus proceeds as follows: 1) Use policy π_θ to sample N^h correlated trajectories from $\mathcal{M}^h$ and $\mathcal{M}^l$, as well as N^l additional trajectories from $\mathcal{M}^l$. 2) Use the sampled trajectories to compute $\hat{\mu}^l$, c, and the correlated values of the random variables $X^{\pi_\theta}_{\tau^h}$ and $X^{\pi_\theta}_{\tau^l}$. 3) Compute the sampled values of Z^{π_θ}. 4) Use the samples of Z^{π_θ} to compute an unbiased, reduced-variance estimate of the policy gradient $\nabla_\theta \mathbb{E}[Z^{\pi_\theta}] \approx \frac{1}{N^h}\sum_\tau \nabla_\theta\big[X^{\pi_\theta}_{\tau^h} + c(X^{\pi_\theta}_{\tau^l} - \hat{\mu}^l)\big]$.

Sampling correlated trajectories from the multi-fidelity environments. Note that for the random variables $X^{\pi_\theta}_{\tau^h}$ and $X^{\pi_\theta}_{\tau^l}$ to be correlated, they must share an underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$. In other words, every outcome $\omega \in \Omega$ from this probability space should uniquely define a high-fidelity $\tau^h(\omega)$ and low-fidelity $\tau^l(\omega)$ trajectory, as well as the corresponding values of the random variables $X^{\pi_\theta}_{\tau^h}(\omega)$ and $X^{\pi_\theta}_{\tau^l}(\omega)$. Careful treatment of this joint probability space is necessary not only for conceptual completeness, but also to implement a mechanism that samples correlated trajectories from the distinct environments $\mathcal{M}^h$ and $\mathcal{M}^l$ under the same policy π_θ.

Informally, when sampling trajectories from $\mathcal{M}^h$ and $\mathcal{M}^l$, there are six sources of stochasticity: the stochasticity introduced by the policy π_θ, that introduced by the initial state distribution Δ_{s_I}, the stochasticity introduced by the high-fidelity and low-fidelity transition dynamics p^h and p^l, and finally the stochasticity introduced by the two reward functions R^h and R^l. Note that the stochasticity introduced by π_θ is implemented by the agent, and is thus under the control of the algorithm in the sense that the same random outcome ω may be used to generate actions from π_θ in both $\mathcal{M}^h$ and $\mathcal{M}^l$. Similarly, the algorithm may fix the initial states in the low-fidelity environment to match those observed from the high-fidelity samples. On the other hand, the transition dynamics and environment rewards are generated by independent sources of stochasticity in the different environments.

We accordingly define the outcome set of the probability space as $\Omega = \Omega_{\Delta_{s_I}} \times \Omega_\pi \times \Omega_{p^h} \times \Omega_{p^l} \times \Omega_{R^h} \times \Omega_{R^l}$. Here, each outcome $\omega^{\Delta_{s_I}} \in \Omega_{\Delta_{s_I}}$ defines a particular shared initial state. Meanwhile, each policy outcome $\omega^\pi \in \Omega_\pi$ corresponds to a sequence $\omega^\pi = \omega^\pi_1, \omega^\pi_2, \ldots, \omega^\pi_T$ that dictates the random sequence of actions selected by the policy π_θ in both environments. Conceptually, given the outcome ω^π_t at any timestep t of a trajectory, the policy should deterministically output action $a^h_t = \pi_\theta(s^h_t, \omega^\pi)$ in the high-fidelity environment,

and action $a_t^l = \pi_\theta(s_t^l, \omega^\pi)$ in the low-fidelity environment. Note that this does not necessarily imply that $a_t^h = a_t^l$, due to a potential difference in states s_t^h and s_t^l that stems from the different dynamics. Similarly, each outcome $\omega^{p_h} \in \Omega_{p^h}$ is a sequence $\omega^{p_h} = \omega_0^{p_h}, \omega_1^{p_h}, \ldots, \omega_T^{p_h}$ dictating the outcomes $s_{t+1} = p^h(s_t^h, a_t^h, \omega^{p_h})$ of the stochastic transitions in the high-fidelity environment. However, unlike the policy outcome sequence $\omega^\pi \in \Omega^\pi$, the transition outcome sequences $\omega^{p_h} \in \Omega^{p_h}$ and $\omega^{p_l} \in \Omega^{p_l}$, and the reward outcome sequences $\omega^{R_h} \in \Omega^{R_h}$ and $\omega^{R_l} \in \Omega^{R_l}$, are not shared between the high and low fidelity environments.

Formulating these separate outcome sets is conceptually helpful. However, practically, we only explicitly sample values for the policy outcomes $\omega_0^\pi, \omega_1^\pi, \ldots, \omega_T^\pi$ in our implementation. Algorithm 1 outlines the steps of the sampling process. First, the pre-sampled policy outcomes are used to rollout a high-fidelity trajectory τ^h. Next, the initial state in the low-fidelity environment is fixed to match the initial state of τ^h and therefore to effectively enforce a common $\omega^{\Delta_{s_I}}$. Finally, the same sequence of pre-sampled policy outcomes is then used to generate the low-fidelity trajectory τ^l.

Algorithm 1: Correlated trajectory sampling.

Input: Current policy π_θ, multi-fidelity environments $\mathcal{M}^h$ and $\mathcal{M}^l$.
Output: Sampled trajectories $\{\tau_i^h, \tau_i^l\}_{i=1}^{N^h}$.
TrajectoryList $\leftarrow$ EmptyList
for $i \in \{1, 2, \ldots, N^h\}$ **do**
  $\omega_0^\pi \ldots \omega_T^\pi \sim SamplePolicyOutcomes()$
  $s_0^h \sim \Delta_{s_I}; s_0^l \leftarrow s_0^h$
  for $t \in \{0, \ldots, T-1\}$ **do**
    $a_t^h \leftarrow \pi_\theta(s_t^h, \omega_t^{\pi_\theta})$
    $s_{t+1}^h \sim p^h(\cdot|s_t^h, a_t^h)$
    $r_t^h \sim R^h(s_t^h, a_t^h, s_{t+1}^h)$
  for $t \in \{0, \ldots, T-1\}$ **do**
    $a_t^l \leftarrow \pi_\theta(s_t^l, \omega_t^{\pi_\theta})$
    $s_{t+1}^l \sim p^l(\cdot|s_t^l, a_t^l)$
    $r_t^l \sim R^l(s_t^l, a_t^l, s_{t+1}^l)$
  $\tau_i^h \leftarrow s_0^h, a_0^h, r_0^h, \ldots, s_T^h$
  $\tau_i^l \leftarrow s_0^l, a_0^l, r_0^l, \ldots, s_T^l$
  TrajectoryList.append($\{\tau_i^h, \tau_i^l\}$)
return *TrajectoryList*

Correlated action sampling via policy distribution reparameterization. In order to use the sampled outcomes ω_t^π to deterministically select an action under the parameterized policy π_θ, we implement a technique inspired by the so-called *reparameterization trick* used in variational autoencoders (Kingma, 2013). In particular, in continuous action spaces, we draw $\omega_t^\pi \sim \mathcal{N}(0, 1)$, and the policy $\pi_\theta(s_t, \omega_t^\pi)$ is trained to output a state-dependent mean and standard deviation which are used to transform ω_t^π into an action a_t. Meanwhile, in discrete action spaces, one can draw $\omega_t^\pi \sim \text{Uniform}(0, 1)$ and apply the Gumbel-Max trick (Huijben et al., 2022) to sample a_t according to the state-dependent probability distribution defined by the policy $\pi_\theta(s_t, \omega_t^\pi)$.

Defining multi-fidelity variant of the REINFORCE algorithm. To this point, we have defined a mechanism for sampling correlated trajectories τ^h and τ^l from multi-fidelity environments $\mathcal{M}^h$ and $\mathcal{M}^l$, as well as a framework for using said trajectories to construct reduced-variance estimators of the policy gradient from trajectory-dependent variables $X_{\tau^h}^{\pi_\theta}$ and $X_{\tau^l}^{\pi_\theta}$. With these elements of the MFPG framework in place, we may implement a multi-fidelity variant of the REINFORCE algorithm. We note that, in principle, MFPG can also be instantiated with other on-policy policy gradient methods, by replacing the way in which the value of $X_\tau^{\pi_\theta}$ is computed from sampled trajectories. In this work, we focus on multi-fidelity REINFORCE in order to ease theoretical analysis, leaving extensions to other algorithms for future work.

We implement a multi-fidelity variant of the REINFORCE algorithm (Williams, 1988; 1992) by defining $X_\tau^{\pi_\theta} := \frac{1}{T}\sum_{t=0}^{T-1} G_t \log \pi_\theta(a_t|s_t)$, as described above. Additionally, to match the commonly used variance-reduced variant of the REINFORCE algorithm (Peters & Schaal, 2006), we subtract state values estimated by a value network from the Monte Carlo returns, i.e., $X_\tau^{\pi_\theta} := \frac{1}{T}\sum_{t=0}^{T-1}(G_t - V_\phi(s_t)) \log \pi_\theta(a_t|s_t) = \frac{1}{T}\sum_{t=0}^{T-1} A_\phi(s_t, a_t) \log \pi_\theta(a_t|s_t)$. Here, $V_\phi(s_t), A_\phi(s_t, a_t)$ denote value and advantage functions estimated from samples. A common advantage network A_ϕ is used to compute both $X_{\tau^h}^{\pi_\theta}$ and $X_{\tau^l}^{\pi_\theta}$, and is trained using sampled tuples $(s_t^h, a_t^h, r_t^h, s_{t+1}^h)$ from the high-fidelity dataset. Algorithm 2 in Section A summarizes the MFPG REINFORCE algorithm.

5 Theoretical Analysis of Multi-Fidelity Policy Gradients

We now analyze the convergence of multi-fidelity policy gradients for the REINFORCE algorithm. We employ the following assumptions on the high and low-fidelity environment MDPs $\mathcal{M}^h, \mathcal{M}^l$ and the parametrized policy π_θ, which are standard in reinforcement learning literature.

Assumption 1 (Bounded Rewards)**.** The high- and low-fidelity reward functions R^h, R^l are bounded, i.e., there exists $U_R^h, U_R^l > 0$ such that $|R^h(s,a)| \leq U_R^h, |R^l(s,a)| \leq U_R^l$,$\forall\ (s,a) \in \mathcal{S} \times \mathcal{A}$.

Assumption 2 (Differentiable Policy)**.** The policy function π_θ is differentiable with respect to θ everywhere, and the score function $\nabla_\theta \log \pi_\theta(a|s)$ exists everywhere.

Assumption 3 (Bounded Score Functions)**.** The score function $\nabla_\theta \log \pi_\theta$ is bounded, i.e., there exists $B_\Theta > 0$ such that $\|\nabla_\theta \log \pi_\theta(a|s)\| \leq B_\Theta,\ \forall\ \theta \in \mathbb{R}^d, (s,a) \in \mathcal{S} \times \mathcal{A}$.

Assumption 4 (Lipschitz Score Functions)**.** The score function $\nabla_\theta \log \pi_\theta$ is L_Θ-Lipschitz, i.e., $\|\nabla_\theta \log \pi_{\theta_1}(a|s) - \nabla_\theta \log \pi_{\theta_2}(a|s)\| \leq L_\Theta \|\theta_1 - \theta_2\|,\ \forall\ \theta_1, \theta_2 \in \mathbb{R}^d, (s,a) \in \mathcal{S} \times \mathcal{A}$.

Assumption 1 is common in the policy gradient literature (Zhang et al., 2020; Bhatnagar et al., 2007). Assumptions 2 to 4 are also common in prior works investigating the convergence of various policy gradient methods (Papini et al., 2018; 2022; Pirotta et al., 2015; Zhang et al., 2020). Particularly, the commonly used Gaussian and Boltzmann parameterized policies are known to satisfy these assumptions (Zhang et al., 2020). We also make the following assumption on the high-fidelity policy gradient estimate.

Assumption 5 (High-Fidelity Policy Gradient Estimate)**.** The high-fidelity policy gradient estimate $\nabla_\theta X_{\tau^h}^{\pi_\theta}$ is unbiased and has bounded variance, i.e., $\mathbb{E}\left[\nabla_\theta X_{\tau^h}^{\pi_\theta}\right] = \nabla_\theta J(\theta)$, and there exists $\sigma > 0$ such that $\mathbb{E}\left[\|\nabla_\theta X_{\tau^h} - \nabla_\theta J(\theta)\|^2\right] \leq \sigma^2$.

Assumption 5 ensures that the high-fidelity policy gradient estimator is a "good" estimator, in the sense that the high-fidelity policy gradient would have learned a good policy if it had unrestricted access to high-fidelity data. Such an assumption is required to formally show convergence (see, e.g., Papini et al. (2018)).

Finally, in order for the proposed approach to be effective, there must be nonzero correlation between random variables computed from the high- and low-fidelity environments, i.e., the coefficient c^* in Eq. (1) is nonzero. Hence, we make the following assumption.

Assumption 6 (Nonzero Correlation)**.** The Pearson correlation coefficient between the high- and low-fidelity random variables is nonzero: there exists $\rho \in (0, 1]$ such that $\left|\mathrm{Cov}\left(X_{\tau^h}^{\pi_\theta}, X_{\tau^l}^{\pi_\theta}\right) / \sqrt{\mathrm{Var}\left(X_{\tau^h}^{\pi_\theta}\right) \mathrm{Var}\left(X_{\tau^l}^{\pi_\theta}\right)}\right| \geq \rho,\ \forall\ \theta \in \mathbb{R}^d$.

Assumption 6 can be interpreted as a comment on the "goodness" of the low-fidelity estimator: while the low-fidelity estimator may be inaccurate and have return values that are incomparable to the high-fidelity due to different transition dynamics in the two environments, the assumption ensures that the lower fidelity still preserves some meaningful connection with the higher fidelity. This connection is captured by the correlation between the two fidelities.

Under these assumptions, we establish that the variance-reduced MFPG iterates lead to a faster decrease in policy gradient norms in comparison to the high-fidelity-only iterates, when employed for a finite horizon REINFORCE algorithm. Further, we establish the global asymptotic convergence of the MFPG to a first-order stationary point of the high-fidelity objective function. The proof is presented in Section B.

Theorem 1. *Under Assumptions 1 to 6, let* $\{\theta_k^{MFPG}\}_{k\in\mathbb{Z}^+}$ *and* $\{\theta_k^h\}_{k\in\mathbb{Z}^+}$ *be the sequences of policy parameters generated by running the REINFORCE algorithm with MFPG and high-fidelity-only policy gradient estimators, respectively. Then, for a problem with time horizon* T*, after* N *iterations, with stepsizes for both iterates taken to be a sequence* α_k *satisfying* $\sum_k \alpha_k = \infty, \sum_k \alpha_k^2 < \infty$*, the policy gradient* $\nabla J(\theta)$ *evaluated in the high-fidelity (target) environment is bounded as follows:*

$$\min_{k\in[N]} \mathbb{E}\left[\|\nabla J(\theta_k^{MFPG})\|^2\right] \leq \frac{J(\theta^*) - J(\theta_1) + \left(\mathbf{1}-\boldsymbol{\rho}^{\mathbf{2}}\right)\frac{\sigma^2 L_T}{2}\sum_{k=1}^N \alpha_k^2}{\sum_{k=1}^N \left(\alpha_k - \frac{\alpha_k^2 L_T}{2}\right)},$$

where L_T *is the Lipschitz constant of the high-fidelity policy gradient, established in Lemma 1. Moreover, we recover the corresponding rate for the high-fidelity only policy iterates* $\{\theta_k^h\}_{k\in\mathbb{Z}^+}$ *by substituting* $\rho = 0$*. Finally, we have that* $\lim_{k\to\infty} \mathbb{E}\left[\|\nabla J(\theta_k^{MFPG})\|\right] = 0$ *almost surely.*

Proof Sketch. The non-asymptotic bounds leverage the Lipschitzness of the high-fidelity-only policy gradient estimator, which we establish in Lemma 1 in Section B, while the asymptotic convergence proof utilizes the supermartingale convergence theorem (Robbins & Siegmund, 1971) as follows. We first show that the norm of the MFPG estimator is bounded, then construct a specific non-negative auxiliary random variable that we show is a supermartingale and apply the supermartingale theorem to establish the result. The complete proof is presented in Section B.

Theorem 1 establishes that the minimum (high-fidelity) policy gradient norm after k iterations will be smaller for the MFPG REINFORCE algorithm compared to the corresponding high-fidelity-only algorithm. Moreover, the *improvement that the MFPG algorithm brings depends on how correlated the high and low-fidelity environments are.* Note that, as k increases, the right side of the inequality in Theorem 1 diminishes to 0, due to the stepsize conditions. Further, the theorem implies that, asymptotically, the MFPG estimator finds a policy that satisfies the first-order optimality conditions for the high-fidelity objective.

6 Experiments

This section empirically evaluates the proposed MFPG algorithm in a suite of simulated benchmark RL tasks. In high-fidelity data-scarce regimes, the experimental results demonstrate the following **key insights**:

- (**I1**) When the multi-fidelity dynamics gap is mild, MFPG can substantially reduce the variance of PG estimates compared to standard variance reduction via state-value baseline subtraction.
- (**I2**) With mild to moderate dynamics gaps, MFPG reliably outperforms the baseline that only uses high-fidelity data in median performance, matching the leading multi-fidelity methods despite its simplicity and minimal tuning overhead. In more challenging scenarios with large gaps, MFPG is the most robust among all evaluated multi-fidelity methods, incurring the fewest performance collapses.
- (**I3**) MFPG can remain effective even when low-fidelity samples exhibit reward misspecification.

6.1 Experiment Setup

Task settings. We evaluate our approach on the off-dynamics RL benchmark introduced by Lyu et al. (2024b), focusing on three standard MuJoCo tasks (Todorov et al., 2012): (i) Hopper-v3, (ii) Walker2d-v3, and (iii) HalfCheetah-v3.[1] In each task, the low-fidelity environment corresponds to the original dynamics, while the high-fidelity variants include changes in gravity, friction, and kinematics. Gravity and friction are varied across five levels $\{0.5\times, 0.8\times, 1.2\times, 2.0\times, 5.0\times\}$; the intermediate $0.8\times$ and $1.2\times$ levels are not part of the original benchmark but are introduced here to examine settings with mild dynamics shifts. Kinematic variations are categorized as easy, medium, or hard, each constraining a specific joint to a reduced range

[1]The benchmark (Lyu et al., 2024b) also contains the Ant-v3 environment. Unlike the other three tasks, the observation space and reward function of Ant-v3 also depend on contact forces, which cannot be reset to specific values by the user. Because of the difficulty in matching initial states across fidelities, we focus our evaluations on the other three tasks.

of motion to mimic actuator damage. For kinematic shifts, we consider cases where the agent's thigh joint is impaired (leg joint for Hopper-v3, since the original benchmark does not include a thigh joint shift for Hopper). In total, the task set spans 39 distinct settings.

Baselines. We compare MFPG against the following baselines: **High-Fidelity Only**, which trains solely on the limited high-fidelity samples; **Low-Fidelity Only (100×)**, which trains on abundant low-fidelity data (100× more samples) and is evaluated in the high-fidelity environment; **DARC** (Eysenbach et al.), which learns domain classifiers to estimate dynamics gaps for the sampled low-fidelity transitions and uses the estimated gaps to augment low-fidelity rewards; **PAR** (Lyu et al., 2024a), which augments low-fidelity rewards using dynamics mismatch estimated in a learned latent space; and **More High-Fidelity Data (15×)**, an oracle baseline that trains with access to 15× more high-fidelity samples. Please refer to Section D for a more detailed description of the baselines.

We instantiate the MFPG framework with the on-policy policy gradient algorithm REINFORCE (Williams, 1992) with state value subtraction. DARC (Eysenbach et al.) and PAR (Lyu et al., 2024a) were originally demonstrated with the off-policy soft actor-critic algorithm (Haarnoja et al., 2018). To isolate the effect of their multi-fidelity mechanisms from the backbone algorithm and enable fair comparison, we adapt both PAR and DARC to the REINFORCE backbone. We include our code in the supplementary materials and will release the project as open source upon acceptance.

Experiment regime. We evaluate MFPG in a regime where high-fidelity samples are scarce, reflecting many real-world applications such as robot learning and molecule design. To simulate this setting, we restrict all methods—except the oracle baseline More High-Fidelity Data (15×)—to a buffer of 100 high-fidelity samples per policy update. Multi-fidelity methods may supplement this limited buffer with up to 100× additional low-fidelity samples per update. Training for all methods is conducted for up to 1M high-fidelity environment steps, except for the More High-Fidelity Data (15×) baseline, which is trained for 15M high-fidelity steps.

Hyperparameter configuration. Given the restriction on high-fidelity samples per policy update, we first tune hyperparameters for the High-Fidelity Only REINFORCE baseline (with state-value baselines subtracted). This tuned configuration serves as the common backbone for all multi-fidelity methods (MFPG, DARC, and PAR), in line with standard practice in the off-dynamics RL literature (Xu et al., 2023; Lyu et al., 2024a). Their method-specific hyperparameters, including the number of low-fidelity samples for DARC and PAR, are tuned separately. The Low-Fidelity Only (100×) baseline is implemented as a direct ablation of the multi-fidelity methods. Finally, the oracle baseline More High-Fidelity Data (15×) is additionally tuned to ensure fair benefit from the extra high-fidelity samples. For further details on hyperparameter setup, see Section D. Overall, our tuning protocol grants multi-fidelity baselines more tuning than MFPG, underscoring the simplicity and minimal tuning overhead of MFPG.

Notably, in our tuning process, PAR is particularly sensitive to hyperparameters; cf. Fig. 14 in Section D. Following (Lyu et al., 2024a), we adopt separate configurations *per task* for PAR, whereas all other methods use a single configuration across settings. MFPG has the fewest hyperparameters among all multi-fidelity methods. We tuned only the exponential moving-average weight η_{ma} in Eq. (2) and found the performance to be insensitive to this choice.

Result reporting. We evaluate all methods using five random seeds. Since five-seed results remain noisy and standard deviations are less reliable with a smaller number of runs, we report median, maximum, and minimum values. In bar plots, bar heights denote medians and error bars indicate maxima and minima; in line plots, solid lines represent medians and shaded regions correspond to maxima/minima.

Performance is evaluated in the high-fidelity environment every 2000 high-fidelity steps, using 10 evaluation episodes with a deterministic policy each time. We adopt two primary metrics: (i) the *final evaluation return*, defined as the median/max/min return averaged over the last 20 evaluation steps, and (ii) area under curve (AUC), the area under the median/max/min evaluation-return curve across training, estimated by integrating evaluation return along the high-fidelity step axis using the composite trapezoidal rule. AUC captures a method's *accumulated* return performance (Stadie et al., 2015; Osband et al., 2016; Hessel et al.,

2018). Finally, to assess robustness, we also report the number of *performance collapses*, defined as cases where the median of a method falls below 50% of the median value of the corresponding High-Fidelity Only baseline.

6.2 Empirical Variance Reduction

This experiment supports our key insight **I1**: When high-fidelity samples are scarce and the dynamics gap between environments is mild, MFPG can substantially reduce the variance of policy gradient estimates compared to standard variance reduction via baseline subtraction.

We use a Hopper task with 1.2× friction as an example. Figure 2 (top) shows the variance of PG estimates from MFPG versus a few variants of the High-Fidelity Only baseline (more specifically, we plot the variances of the scalar quantities Z^{π_θ} and $X^{\pi_\theta}_{\tau^h}$ before differentiation with respect to policy parameters). The plotted High-Fidelity Only variants include a baseline that has access to the same limited number of high-fidelity samples per policy update as MFPG (i.e., the same batch size), baselines with more high-fidelity samples (a batch size of 1000 and 3000), and variants of these High-Fidelity Only baselines that also use state-value function subtraction in an effort to variance reduction.

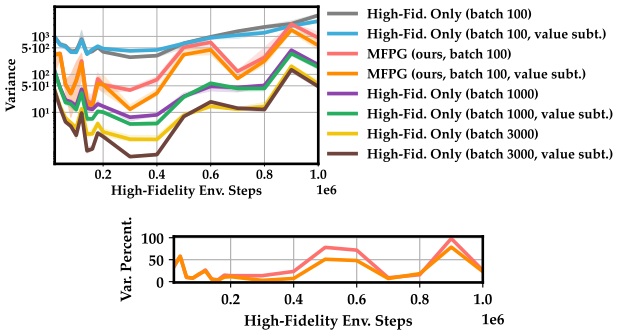

Figure 2: Variance of policy gradient estimates for MFPG versus variants of the High-Fidelity Only baseline on Hopper-v3 (top), and the percentage of MFPG variance relative to single-fidelity counterparts of the same batch size (bottom). In high-fidelity data-scarce regimes with mild dynamics gaps, MFPG generally exhibits far lower PG variance than the single-fidelity counterparts.

We train a policy using the single-fidelity REINFORCE algorithm with state-value subtracted for 1 million steps and save the trained policy at 18 different checkpoints. For each of these saved policies, we collect 200 batches of both high- and low-fidelity data (with low-fidelity data collected at 100× the amount of high-fidelity data), where the size of each high-fidelity batch, i.e., the amount of high-fidelity data, varies between approaches as described above. We then record the empirical variance of the policy gradient estimates from these 200 batches, for each checkpointed policy. We repeat this experiment for 5 random seeds and report aggregate statistics in Figure 2 (top) where each line is based on 21600 batches of policy gradient estimates. Solid lines denote medians and shaded regions (which largely overlap the median lines) indicate the range between maxima and minima across the 5 seeds. Figure 2 (bottom) reports the ratio of median variance of policy gradient estimators for two variants of MFPG (with and without value-function subtraction), measured relative to their single-fidelity counterparts under the same batch size.

As shown in Figure 2, in data-scarce regimes, MFPG can substantially reduce the variance of policy gradient estimators compared to single-fidelity baselines at most training checkpoints. This variance reduction is generally far greater than that achieved by standard state-value subtraction, narrowing the gap to baselines that rely on significantly more high-fidelity samples (e.g., 10×).

6.3 Large-Scale Comparison Across Dynamics Variation Types and Levels

This section reports the large-scale evaluation of MFPG against baseline methods under the task settings described in Section 6.1 and supports our key insight **I2**. For clarity, we present in Figs. 3 and 4 a representative subset of results—specifically, one dynamics variation type with multiple variation levels for each task. The complete results across all 39 task settings are provided in Section C.

Fig. 3 presents the final evaluation return of all methods on 15 representative task settings (out of 39). Each row corresponds to one task, with the high-fidelity environment varying across multiple dynamics levels. For the simpler Hopper and Walker2d tasks, training is relatively stable, and the final evaluation return and AUC

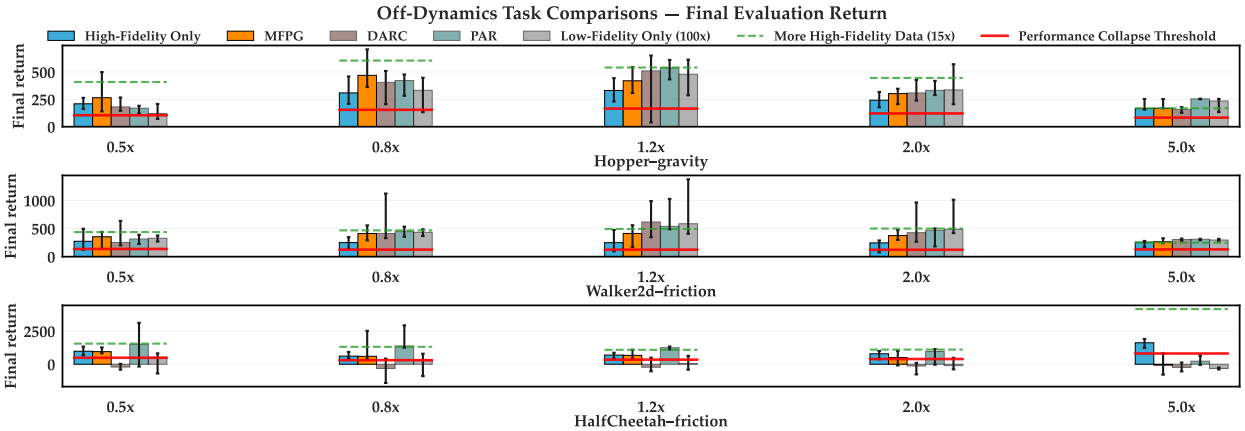

Figure 3: Final evaluation returns of MFPG versus baselines in high-fidelity data-scarce regimes. On Hopper-v3 and Walker2d-v3, under mild–moderate multi-fidelity dynamics gaps, MFPG reliably improves median performance over the High-Fidelity Only baseline and matches that of leading multi-fidelity methods, despite its simplicity and minimal tuning overhead. Under large dynamics gaps, all multi-fidelity approaches converge toward the High-Fidelity Only baseline performance. On the more challenging HalfCheetah-v3, training stability decreases; here we additionally report the AUC metric (Fig. 4) to capture accumulated performance, as it may diverge from final return trends. In this task, MFPG ranks among the best-performing methods overall and demonstrates the strongest robustness (cf., Fig. 4).

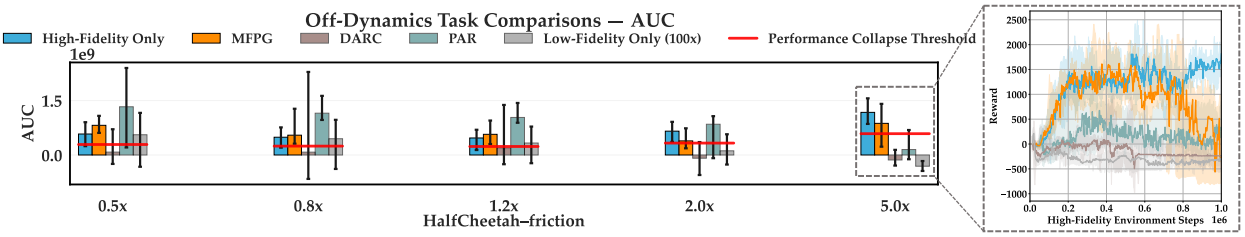

Figure 4: AUC of MFPG versus baselines in high-fidelity data-scarce regimes on HalfCheetah-v3 with friction variations. Under an extreme dynamics gap (5× friction), we additionally report evaluation return curves for all methods—MFPG is the only multi-fidelity approach that remains comparable to High-Fidelity Only in accumulated performance (AUC), while all multi-fidelity methods suffer severe degradation in terms of final return (cf., Fig. 3).

metrics are consistent; thus, we report only the final evaluation return. In contrast, the HalfCheetah task is more challenging, with training exhibiting greater fluctuations that occasionally reduce reward performance (baselines fluctuate far more substantially than MFPG). To capture accumulated learning performance more comprehensively in these settings, we report both the final return (Fig. 3, row 3) and the AUC metric (Fig. 4).

Hopper and Walker2d tasks. At a high level, when the multi-fidelity dynamics gap is mild—e.g., rows 1–2 with 0.8× and 1.2× gravity or friction in Fig. 3—the median of MFPG improves reliably over High-Fidelity Only, narrowing the gap to the median of the oracle baseline (with privileged access to 15× more high-fidelity data). As the dynamics gap grows but remains moderate (e.g., 0.5× and 2.0× gravity or friction in rows 1–2), where cross-fidelity correlation weakens, the median of MFPG still generally improves upon High-Fidelity Only, albeit with smaller margins. In most of these settings, the medians of other baselines also improve upon High-Fidelity Only and are competitive with one another. Notably, MFPG generally matches the performance of leading multi-fidelity methods despite its algorithmic simplicity and minimal tuning overhead. In settings with large dynamics gaps, i.e., 5× shifts, the performance of all evaluated approaches becomes generally comparable to High-Fidelity Only.

Another interesting observation is that the performance of Low-Fidelity Only (100×) does not consistently correlate with the dynamics shift level. For instance, in Hopper with 2× gravity, Low-Fidelity Only (100×) outperforms High-Fidelity Only in median, whereas at 0.5× gravity it yields a lower median—highlighting the inherent difficulty of predicting when zero-shot transfer will be effective in practice.

HalfCheetah task. The HalfCheetah task is generally more challenging for all evaluated methods. DARC and Low-Fidelity Only (100×) are particularly fragile, with performance collapsing on both final evaluation return and AUC in most settings. When the dynamics gap is mild-moderate, PAR is the strongest baseline overall, often achieving the highest median performance. However, PAR requires *computationally expensive, per-task* hyperparameter tuning, as PAR is sensitive to the reward-augmentation weight β; cf. Fig. 14 in Section D. Even with such per-task configurations, the performance of PAR often exhibits large cross-seed variance with low minima (cf., 0.5× and 2.0× friction in Figs. 3 and 4, and several additional cases in Section C). Under large dynamics gaps, i.e., 5x friction shift, all methods experience severe performance collapse in terms of final return. Notably, MFPG remains comparable to High-Fidelity Only on the AUC metric. Fig. 4 shows the return curves under the 5× friction setting: PAR, DARC, and Low-Fidelity Only (100×) make little progress throughout training, whereas MFPG initially tracks High-Fidelity Only before showing some instability, resulting in lower final return but strong accumulated performance (i.e., AUC). The observed instability may stem from noise in the estimated control variates coefficient $\hat{c}^*$. Designing more reliable estimation schemes constitutes an avenue for future work.

A similar trend in PAR performance is observable under other types of dynamics variation of HalfCheetah, cf., gravity and kinematic shifts in Figs. 12 and 13 of Section C. In HalfCheetah with kinematic variation, PAR's median performance drops sharply below that of MFPG at the medium level and collapses entirely at the hard level on both metrics, whereas MFPG closely tracks the median of the High-Fidelity Only baseline throughout.

Robustness. Across all 39 task settings, the numbers of performance collapses (in final return / AUC) are as follows: MFPG (2 / 0), PAR (4 / 2), Low-Fidelity Only (100×) (11 / 9), and DARC (12 / 10). Among all methods, MFPG exhibits the strongest robustness.

MFPG performance vs. correlation. We examine one task in detail to illustrate the qualitative behavior of MFPG under varying dynamics gaps. Figure 5 presents evaluation-return curves of MFPG and High-Fidelity Only in the Hopper task with three levels of gravity variation (0.8×, 2.0×, and 5.0×), along with the estimated Pearson correlation coefficients between high- and low-fidelity policy gradient losses throughout training. In this task, the correlation between fidelities generally decreases as the dynamics shift becomes larger. Correspondingly, the median performance gains of MFPG over High-Fidelity Only are

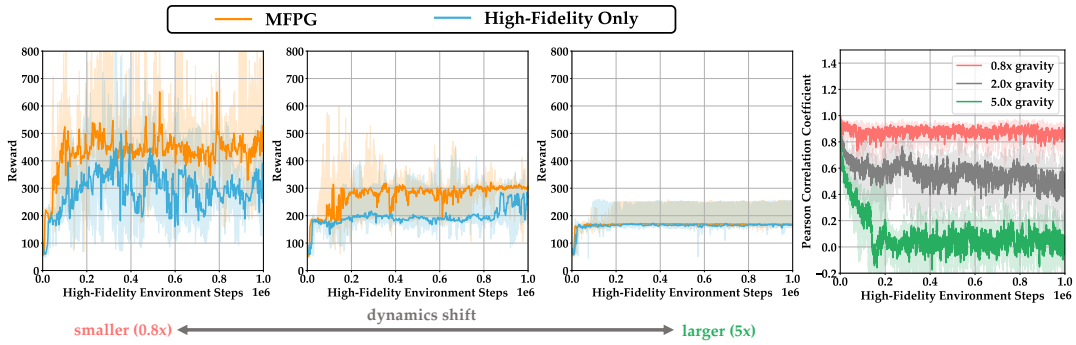

Figure 5: Evaluation return curves of MFPG versus the High-Fidelity Only baseline on Hopper-v3, corresponding to Fig. 3, under mild, moderate, and large gravity variations (left to right: 0.8×, 2.0×, 5.0×), together with estimated Pearson correlation coefficients between high- and low-fidelity policy gradient losses. MFPG performance increases as the cross-fidelity correlation strengthens.

greater when the correlation is higher (smaller gap). In the extreme case of 5× gravity, the correlation drops sharply, and MFPG performance nearly coincides with that of High-Fidelity Only.

6.4 Low-Fidelity Environment with a Misspecified Reward

This experiment is designed to support our key insight **I3**, that when high-fidelity samples are scarce, MFPG can remain effective even in the presence of reward misspecification in the low-fidelity samples.

As discussed in Section 4, so long as there is a statistical relationship between the random variables of interest $X_{\tau^h}^{\pi_\theta}$ and $X_{\tau^l}^{\pi_\theta}$ (i.e., $\rho^2(X_{\tau^h}^{\pi_\theta}, X_{\tau^l}^{\pi_\theta})$ is non-negligible), the MFPG framework will reduce the variance of the policy gradient estimates (with respect to the high-fidelity environment).

To demonstrate this point, we examine a situation in which the reward function in the low-fidelity environment is the negative of that from the high-fidelity environment. Figure 6 reports the performance of MFPG in comparison to the High-Fidelity Only and Low-Fidelity Only baselines. All approaches in this example use the plain REINFORCE algorithm without state-value subtraction. The core idea extends to the case with value-function subtraction, except that separate value functions must be learned for the multi-fidelity environments because of the misspecified reward.

As expected, the Low-Fidelity Only baseline is entirely ineffective under the drastically misspecified reward: instead of learning to move forward, the agent learns to end the episode as quickly as possible. Meanwhile, the High-Fidelity Only baseline makes limited progress due to the scarcity of high-fidelity samples. In contrast, MFPG effectively combines the two data sources and achieves substantially better performance than both baselines.

Intuitively, although the values of $X_{\tau^l}^{\pi_\theta}$ are entirely different from those of $X_{\tau^h}^{\pi_\theta}$, they are negatively correlated in this example. The MFPG algorithm takes advantage of this relationship to compute a correction for the low-fidelity estimator $\hat{\mu}^l$, using only a small number of high-fidelity samples. Although this is an extreme example in the sense that the low- and high-fidelity tasks are polar opposites of each other, it highlights a useful feature of the MFPG framework: the low-fidelity rewards and dynamics might be very different from the target environment of interest (making direct sim-to-real transfer infeasible), and yet still provide useful information for multi-fidelity policy training.

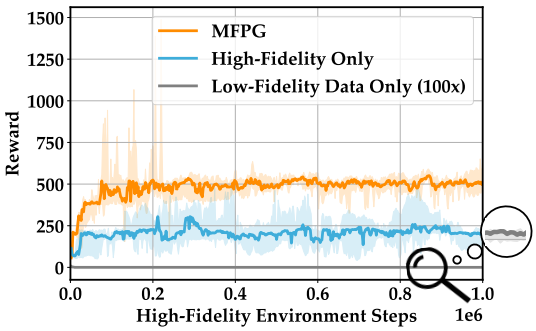

Figure 6: MFPG, High-Fidelity Only, and Low-Fidelity Only on Hopper-v3, where the low-fidelity environment is misspecified with a negated reward. Although low-fidelity data alone is ineffective for training, MFPG successfully leverages both fidelities to achieve substantial improvements over High-Fidelity Only in high-fidelity data-scarce regimes.

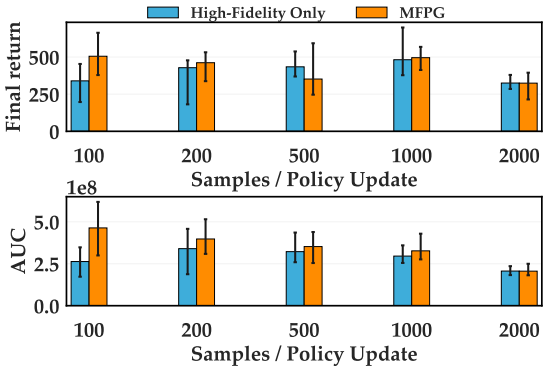

Figure 7: Final return (top) and AUC (bottom) of MFPG versus High-Fidelity Only on Hopper-v3 with 1.2× friction, across varying high-fidelity batch sizes (i.e., samples per update). MFPG yields substantial median performance gains over High-Fidelity Only in high-fidelity data-scarce regimes (batch size 100). When high-fidelity data is sufficient (e.g., batch size 2000), augmenting with abundant low-fidelity data provides no further benefit.

6.5 Additional Experiments

This section presents additional experiments that provide a more detailed examination of different aspects of the MFPG framework.

MFPG performance with different amounts of high-fidelity samples. Our main evaluation in Section 6.3 focuses on the high-fidelity data-scarce regime, which reflects many real-world scenarios where collecting large quantities of high-fidelity samples for each policy update is prohibitively expensive (e.g., in robotics or molecule design). In this section, we study how MFPG compares to the High-Fidelity Only baseline as the number of high-fidelity samples per policy update varies.

Figure 7 reports final return (top) and AUC (bottom) in the Hopper task with 1.2× friction, trained for 1 million steps using MFPG with 100× low-fidelity data and High-Fidelity Only across different batch sizes (i.e., high-fidelity samples per update). Note that with a fixed total number of high-fidelity environment steps, the number of gradient updates varies across batch sizes; consequently, the reported metrics do not necessarily increase monotonically with larger batch sizes. In the data-scarce regime (batch size 100), the medians of MFPG are substantially greater than those of High-Fidelity Only. The margin of improvement decreases as more high-fidelity samples become available (e.g., batch size 200), and when high-fidelity samples are abundant (e.g., batch size 2000), MFPG closely matches High-Fidelity Only. These results highlight that MFPG is most effective in high-fidelity data-scarce settings.

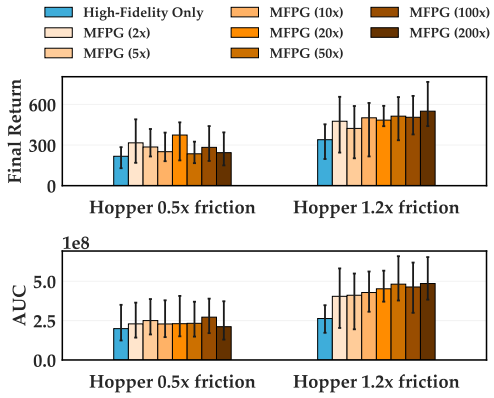

Figure 8: Final return (top) and AUC (bottom) of MFPG with increasing amounts of low-fidelity data (light to dark orange) versus High-Fidelity Only (blue) on Hopper-v3 with 0.5× and 1.2× friction. Overall, incorporating additional low-fidelity samples yields greater benefits in mild-gap settings.

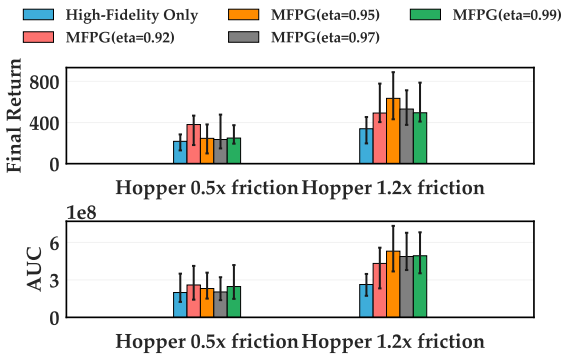

Figure 9: Final return (top) and AUC (bottom) of MFPG with varying exponential moving-average weights (η_{ma} in Eq. (2)) versus High-Fidelity Only on Hopper-v3 with 0.5× and 1.2× friction. MFPG remains robust to the choice of η_{ma} with medians consistently exceeding those of High-Fidelity Only.

MFPG performance with different amounts of low-fidelity samples. We assume throughout this work that low-fidelity samples are cheap to collect, and all methods are given access to an abundant budget of low-fidelity data. Consequently, optimizing the efficiency of low-fidelity sample usage (Cutler et al., 2015) is not the focus of this work. Instead, this section qualitatively examines how the performance of MFPG varies with different amounts of low-fidelity data.

In principle, the performance of MFPG is robust with respect to the amount of low-fidelity samples, so long as the low-fidelity sample amount is sufficient to estimate the low-fidelity sample mean $\hat{\mu}^l$. Figure 8 reports the final return (top) and AUC (bottom) of MFPG with increasing amounts of low-fidelity samples (indicated from light to dark orange) for estimating low-fidelity sample mean ($\hat{\mu}^l$ in Fig. 1), compared with High-Fidelity Only in Hopper with 0.5× and 1.2× friction. At a high level, when the dynamics gap is mild (e.g., 1.2× friction), increasing the number of low-fidelity samples tends to yield higher median performance,

especially in terms of AUC. However, under larger dynamics gaps, no clear monotonic trend is observed. Overall, additional low-fidelity samples are most beneficial in mild-gap settings.

Sensitivity to exponential moving-average weight. We further examine the sensitivity of MFPG to the choice of the exponential moving-average weight η_{ma} in Eq. (2). Figure 9 reports the final return and AUC of MFPG with different η_{ma} values in the Hopper task under 0.5× and 1.2× friction, alongside the High-Fidelity Only baseline. Overall, MFPG is relatively robust to the choice of η_{ma}, i.e., the medians do not fall below those of High-Fidelity Only baseline, and its median performance improves more substantially in settings with smaller dynamics gaps—consistent with the key insights from our baseline comparison in Section 6.3.

Reparameterization trick. We examine the effect of the reparameterization trick described in Section 4 on our MFPG method. Figure 10 shows that when the dynamics gap is mild (cf., the 1.2× friction setting in Hopper, right panel), reparameterization substantially accelerates learning; in contrast, when the gap is larger (cf., the 0.5× friction setting, left panel), it offers much weaker benefit. This pattern is intuitive: when low-fidelity dynamics are close to those of the high-fidelity environment, stochasticity in action selection dominates; reparameterization controls this source of randomness and produces highly correlated rollouts. In the extreme case where the low-fidelity dynamics are perfect (and deterministic), reparameterization makes the low- and high-fidelity rollouts identical, so that $X_{\tau^h}^{\pi_\theta}$ and $X_{\tau^l}^{\pi_\theta}$ cancel out. The policy gradient estimate then reduces to μ^l, which is estimated with abundant (perfect) low-fidelity samples. By contrast, when the dynamics gap is large, discrepancies in transition dynamics dominate, and even with reparameterization, divergence between high- and low-fidelity rollouts persists, resulting in weaker correlation.

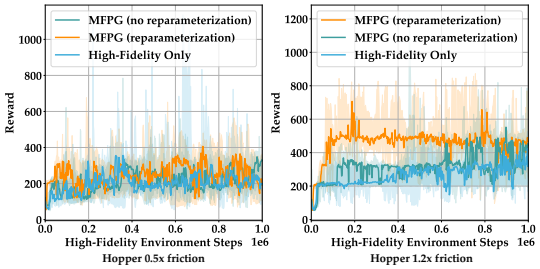

Figure 10: MFPG with and without the reparameterization trick versus High-Fidelity Only on Hopper-v3 with 0.5× (left) and 1.2× (right) friction. The reparameterization trick proves especially critical in mild-gap settings (e.g., 1.2× friction).

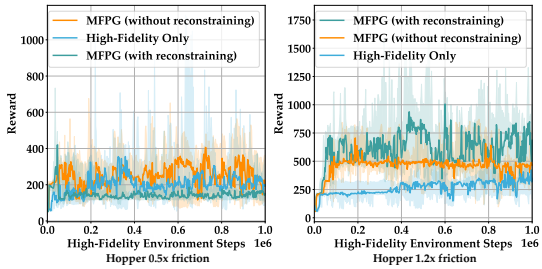

Figure 11: MFPG with and without periodic reconstraining of low-fidelity trajectories versus High-Fidelity Only on Hopper-v3 with 0.5× (left) and 1.2× (right) friction. Reconstraining can enhance MFPG performance in some settings (cf. right) but degrade it in others (cf. left). Accordingly, we do not employ this trick in our evaluations of MFPG.

Variance-bias trade-off. In practice, we observed that periodically reconstraining (the correlated) low-fidelity rollouts back to their high-fidelity counterparts can strengthen correlation and improve performance (e.g., the 1.2× friction setting in Fig. 11, right panel). However, this procedure may also introduce bias into the policy gradient estimator, leading to degraded performance in certain cases (e.g., the 0.5× friction setting in Fig. 11, left panel). These findings highlight a practical variance–bias trade-off. We do not adopt this reconstraining trick in our experiments. Developing systematic, unbiased mechanisms for drawing even more highly correlated samples across environments remains an important direction for future work.

7 Conclusions

We present a multi-fidelity policy gradient (MFPG) framework, which mixes a small amount of potentially expensive high-fidelity data with a larger volume of cheap lower-fidelity data to construct unbiased,

variance-reduced estimators for on-policy policy gradients. We instantiate this framework through a practical, multi-fidelity variant of the standard REINFORCE algorithm. Theoretically, we show that under standard assumptions, the MFPG estimator guarantees asymptotic convergence of multi-fidelity REINFORCE to locally optimal policies (up to first order optimality conditions) in the target high-fidelity environment of interest. Moreover, whenever the statistical correlation between high- and low-fidelity samples is nontrivial, MFPG REINFORCE achieves faster finite-sample convergence than its single-fidelity counterpart. Empirical evaluation of MFPG in high-fidelity data-scarce regimes across a suite of simulated robotics benchmark tasks with varied multi-fidelity transition dynamics highlights both its efficacy and robustness. With mild to moderate dynamics gaps, MFPG reliably improves median performance over a standard high-fidelity-only baseline and matches the performance of leading multi-fidelity methods, despite its simplicity and minimum tuning overhead. Under large gaps, MFPG demonstrates the greatest robustness among the evaluated multi-fidelity approaches. Furthermore, an additional experiment demonstrates that MFPG can remain effective even in the presence of reward misspecification in the low-fidelity environments.

In summary, the proposed framework offers a novel paradigm for leveraging multi-fidelity data in reinforcement learning, providing promising directions for efficient sim-to-real transfer and principled approaches to managing the trade-off between policy performance and data collection costs. Future research spans several avenues. First, extending the MFPG framework to modern RL algorithms—including actor–critic and off-policy/offline methods—raises new questions about bias–variance trade-offs that merit deeper investigation. Second, developing systematic techniques to enhance correlation between high- and low-fidelity samples without introducing bias is critical for further performance gains. Finally, applying MFPG to real-world robotic systems represents an important step toward realizing its potential in practical sim-to-real transfer.

Broader Impact Statement

This work introduces a methodological contribution—multi-fidelity policy gradients—for improving the (high-fidelity) sample efficiency of reinforcement learning in settings with access to both high- and low-fidelity environments. We do not foresee immediate societal risks arising directly from this research. Potential downstream applications may include robotics and autonomous systems, where increased efficiency can reduce computational and data collection costs.

References

Pieter Abbeel, Morgan Quigley, and Andrew Y Ng. Using inaccurate models in reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 1–8, 2006.

Akash Agrawal and Christopher McComb. Adaptive learning of design strategies over non-hierarchical multi-fidelity models via policy alignment. *arXiv preprint arXiv:2411.10841*, 2024.

Karol Arndt, Murtaza Hazara, Ali Ghadirzadeh, and Ville Kyrki. Meta reinforcement learning for sim-to-real domain adaptation. In *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 2725–2731. IEEE, 2020.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021.

Shalabh Bhatnagar, Mohammad Ghavamzadeh, Mark Lee, and Richard S Sutton. Incremental natural actor-critic algorithms. *Advances in neural information processing systems*, 20, 2007.

Sahil Bhola, Suraj Pawar, Prasanna Balaprakash, and Romit Maulik. Multi-fidelity reinforcement learning framework for shape optimization. *Journal of Computational Physics*, 482:112018, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2023.112018. URL `https://www.sciencedirect.com/science/article/pii/S0021999123001134`.

Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019*

International Conference on Robotics and Automation (ICRA), pp. 8973–8979, 2019. doi: 10.1109/ICRA.2019.8793789.

Ching-An Cheng, Xinyan Yan, and Byron Boots. Trajectory-wise control variates for variance reduction in policy gradient methods. In *Conference on Robot Learning*, pp. 1379–1394. PMLR, 2020.

Mark Cutler, Thomas J. Walsh, and Jonathan P. How. Real-world reinforcement learning via multifidelity simulators. *IEEE Transactions on Robotics*, 31(3):655–671, 2015. doi: 10.1109/TRO.2015.2419431.

Longchao Da, Justin Turnau, Thirulogasankar Pranav Kutralingam, Alvaro Velasquez, Paulo Shakarian, and Hua Wei. A survey of sim-to-real methods in rl: Progress, prospects and challenges with foundation models. *arXiv preprint arXiv:2502.13187*, 2025.

Peter Dayan. Reinforcement comparison. In *Connectionist Models*, pp. 45–51. Elsevier, 1991.

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Siddharth Desai, Ishan Durugkar, Haresh Karnan, Garrett Warnell, Josiah Hanna, and Peter Stone. An imitation from observation approach to transfer learning with dynamics mismatch. *Advances in Neural Information Processing Systems*, 33:3917–3929, 2020a.

Siddharth Desai, Haresh Karnan, Josiah P Hanna, Garrett Warnell, and Peter Stone. Stochastic grounded action transformation for robot learning in simulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6106–6111. IEEE, 2020b.

Benjamin Eysenbach, Shreyas Chaudhari, Swapnil Asawa, Sergey Levine, and Ruslan Salakhutdinov. Off-dynamics reinforcement learning: Training for transfer with domain classifiers. In *International Conference on Learning Representations*.

Raj Ghugare, Santiago Miret, Adriana Hugessen, Mariano Phielipp, and Glen Berseth. Searching for high-value molecules using reinforcement learning and transformers. *arXiv preprint arXiv:2310.02902*, 2023.

Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*, 2018.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. In *International Conference on Learning Representations*, 2017.

Yihong Guo, Yixuan Wang, Yuanyuan Shi, Pan Xu, and Anqi Liu. Off-dynamics reinforcement learning via domain adaptation and reward augmented imitation. *Advances in Neural Information Processing Systems*, 37:136326–136360, 2024.

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*, pp. 1861–1870. Pmlr, 2018.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jiawei Huang and Nan Jiang. From importance sampling to doubly robust policy gradient. In *International Conference on Machine Learning*, pp. 4434–4443. PMLR, 2020.

Iris AM Huijben, Wouter Kool, Max B Paulus, and Ruud JG Van Sloun. A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning. *IEEE transactions on pattern analysis and machine intelligence*, 45(2):1353–1371, 2022.

Elia Kaufmann, Leonard Bauersfeld, Antonio Loquercio, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. Champion-level drone racing using deep reinforcement learning. *Nature*, 620(7976):982–987, 2023.

Sami Khairy and Prasanna Balaprakash. Multi-fidelity reinforcement learning with control variates. *Neurocomputing*, 597:127963, 2024.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Hao Liu, Yihao Feng, Yi Mao, Dengyong Zhou, Jian Peng, and Qiang Liu. Action-dependent control variates for policy optimization via stein identity. In *International Conference on Learning Representations*, 2018.

Jinxin Liu, Hongyin Zhang, and Donglin Wang. Dara: Dynamics-aware reward augmentation in offline reinforcement learning. *arXiv preprint arXiv:2203.06662*, 2022.

Lennart Ljung. System identification. In *Signal analysis and prediction*, pp. 163–173. Springer, 1998.

Jiafei Lyu, Chenjia Bai, Jing-Wen Yang, Zongqing Lu, and Xiu Li. Cross-domain policy adaptation by capturing representation mismatch. In *International Conference on Machine Learning*, pp. 33638–33663. PMLR, 2024a.

Jiafei Lyu, Kang Xu, Jiacheng Xu, Mengbei Yan, Jingwen Yang, Zongzhang Zhang, Chenjia Bai, Zongqing Lu, and Xiu Li. Odrl: A benchmark for off-dynamics reinforcement learning. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. URL `https://openreview.net/forum?id=ap4x1kArGy`.

Alonso Marco, Felix Berkenkamp, Philipp Hennig, Angela P Schoellig, Andreas Krause, Stefan Schaal, and Sebastian Trimpe. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1557–1563. IEEE, 2017.

Barry L Nelson. On control variate estimators. *Computers & Operations Research*, 14(3):219–225, 1987.

Haoyi Niu, Yiwen Qiu, Ming Li, Guyue Zhou, Jianming Hu, Xianyuan Zhan, et al. When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning. *Advances in Neural Information Processing Systems*, 35:36599–36612, 2022.

Haoyi Niu, Jianming Hu, Guyue Zhou, and Xianyuan Zhan. A comprehensive survey of cross-domain policy transfer for embodied agents. *arXiv preprint arXiv:2402.04580*, 2024.

Haoyi Niu, Tianying Ji, Bingqi Liu, Haocheng Zhao, Xiangyu Zhu, Jianying Zheng, Pengfei Huang, Guyue Zhou, Jianming Hu, and Xianyuan Zhan. H2o+: an improved framework for hybrid offline-and-online rl with dynamics gaps. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1421–1428. IEEE, 2025.

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

Art B. Owen. *Monte Carlo theory, methods and examples.* `https://artowen.su.domains/mc/`, 2013.

Sergey Pankov. Reward-estimation variance elimination in sequential decision processes. *arXiv preprint arXiv:1811.06225*, 2018.

Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. In *International conference on machine learning*, pp. 4026–4035. PMLR, 2018.

Matteo Papini, Matteo Pirotta, and Marcello Restelli. Smoothing policies and safe policy gradients. *Machine Learning*, 111(11):4081–4137, 2022.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Siam Review*, 60(3):550–591, 2018.

Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2219–2225, 2006. doi: 10.1109/IROS.2006.282564.

Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100(2):255–283, 2015.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL `http://jmlr.org/papers/v22/20-1364.html`.

Fabio Ramos, Rafael Carvalhaes Possas, and Dieter Fox. Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators. *arXiv preprint arXiv:1906.01728*, 2019.

Herbert Robbins and David Siegmund. A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing methods in statistics*, pp. 233–257. Elsevier, 1971.

Andrei A Rusu, Matej Večerík, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. In *Conference on robot learning*, pp. 262–270. PMLR, 2017.

Gilhyun Ryou, Geoffrey Wang, and Sertac Karaman. Multi-fidelity reinforcement learning for time-optimal quadrotor re-planning. *arXiv preprint arXiv:2403.08152*, 2024.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Laura Smith, J. Chase Kew, Xue Bin Peng, Sehoon Ha, Jie Tan, and Sergey Levine. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 1593–1599, 2022. doi: 10.1109/ICRA46639.2022.9812166.

Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

Varun Suryan, Nahush Gondhalekar, and Pratap Tokekar. Multi-fidelity reinforcement learning with gaussian processes. *arXiv preprint arXiv:1712.06489*, 2017.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard Stuart Sutton. *Temporal credit assignment in reinforcement learning*. University of Massachusetts Amherst, 1984.

Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. *Annual Review of Control, Robotics, and Autonomous Systems*, 8, 2024.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Matthew E Taylor, Peter Stone, and Yaxin Liu. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(9), 2007.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30. IEEE, 2017.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. In *International conference on machine learning*, pp. 5015–5024. PMLR, 2018.

Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, pp. 538–545, 2001.

RJ Williams. Toward a theory of reinforcement-learning connectionist systems. *Technical Report*, 1988.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*, 2018.

Kang Xu, Chenjia Bai, Xiaoteng Ma, Dong Wang, Bin Zhao, Zhen Wang, Xuelong Li, and Wei Li. Cross-domain policy adaptation via value-guided data filtering. *Advances in Neural Information Processing Systems*, 36:73395–73421, 2023.

Patrick Yin, Tyler Westenbroek, Simran Bagaria, Kevin Huang, Ching-an Cheng, Andrey Kobolov, and Abhishek Gupta. Rapidly adapting policies to the real world via simulation-guided fine-tuning. *arXiv preprint arXiv:2502.02705*, 2025.

Kaiqing Zhang, Alec Koppel, Hao Zhu, and Tamer Basar. Global convergence of policy gradient methods to (almost) locally optimal policies. *SIAM Journal on Control and Optimization*, 58(6):3586–3612, 2020.

Tingting Zhao, Hirotaka Hachiya, Gang Niu, and Masashi Sugiyama. Analysis and improvement of policy gradient estimation. *Advances in Neural Information Processing Systems*, 24, 2011.

Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744. IEEE, 2020.

Yuanyi Zhong, Yuan Zhou, and Jian Peng. Coordinate-wise control variates for deep policy gradients. *arXiv preprint arXiv:2107.04987*, 2021.

A Complete MFPG REINFORCE Algorithm

Algorithm 2 presents the complete algorithm block for the proposed multi-fidelity variant of the REINFORCE algorithm.

Algorithm 2: Multi-Fidelity Policy Gradients (MFPGs) - REINFORCE

Input: Initial policy π_θ, multi-fidelity environments $\mathcal{M}^h$ and $\mathcal{M}^l$, batch size for correlated trajectory samples N^h, batch size for control variate sample mean estimate N^l.
Output: Trained policy π_θ.
while *maximum high-fidelity environment step not reached* **do**

Collect N^h correlated trajectory samples $\mathcal{D}_1 = \{\tau_i^h, \tau_i^l\}_{i=1}^{N^h}$. ; `// See Algorithm 1`
Collect N^l additional uncorrelated low-fidelity trajectory samples $\mathcal{D}_2 = \{\tau_j^l\}_{j=1}^{N^l}$.
Compute correlated policy gradient random variables using $\mathcal{D}_1$:

$$X^{\pi_\theta}_{\tau^h,i} = \frac{1}{T}\sum_{t=0}^{T-1}(G^h_{i,t} - V_\phi(s^h_{i,t}))\,\log\pi_\theta(a^h_{i,t} \mid s^h_{i,t}),$$

$$X^{\pi_\theta}_{\tau^l,i} = \frac{1}{T}\sum_{t=0}^{T-1}(G^l_{i,t} - V_\phi(s^l_{i,t}))\,\log\pi_\theta(a^l_{i,t} \mid s^l_{i,t}).$$

Estimate the optimal control variate coefficient $\hat{c}^*$ using $\mathcal{D}_1$ and Eq. (2).
Estimate the control variate sample mean using $\mathcal{D}_2$: $\hat{\mu}^l = \frac{1}{N^l}\sum_{j=1}^{N^l} X^{\pi_\theta}_{\tau^l,j}$.
Construct multi-fidelity policy gradient estimator: $\mathbb{E}[\nabla_\theta Z^{\pi_\theta}] \approx \frac{1}{N^h}\sum_{i=1}^{N^h}\nabla_\theta\big[X^{\pi_\theta}_{\tau^h,i} + \hat{c}^*(X^{\pi_\theta}_{\tau^l,i} - \hat{\mu}^l)\big]$.
Perform policy optimization.
Fit the value function network $V_\phi(s^l_{i,t})$ using high-fidelity samples in $\mathcal{D}_1$.

return *policy* π_θ

B Proof of Theorem 1

The proof of Theorem 1 leverages the Lipschitz continuity of the policy gradient for a policy deployed in the MDP $\mathcal{M}^h$, under assumptions Assumptions 1 to 4. Under these assumptions, the Lipschitzness of the policy gradient was proved for the infinite horizon setting by Zhang et al. (2020). We follow a similar procedure to establish the Lipschitzness for the finite-horizon case, stated below.

Lemma 1 (Adapted from Zhang et al. (2020))**.** *Under Assumptions 1 to 4, for a problem of horizon T, the high-fidelity only policy gradient $\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\mathcal{M}^h(\theta)}[R(\tau)]$ is L_T-Lipschitz, where*

$$L_T = \frac{1-(T+2)\gamma^{T+1}+(T+1)\gamma^{T+2}}{(1-\gamma)^2}\cdot U_R^h L_\Theta + \frac{\left(1-(T+2)^2\gamma^{T+1}+(T+3)(T+1)\gamma^{T+2}\right)(1-\gamma)+2\gamma\left(1-(T+2)\gamma^{T+1}+(T+1)\gamma^{T+2}\right)}{(1-\gamma)^3}\cdot U_R^h B_\Theta^2.$$

Proof. Note that the high-fidelity only policy gradient can be written as

$$\nabla J(\theta) = \sum_{t=0}^{T}\sum_{\kappa=0}^{T-t}\gamma^{t+\kappa}\int r^h(s^h_{t+\kappa}, a^h_{t+\kappa})\cdot\nabla_\theta\log\pi_\theta\cdot p_{\theta,0:t+\kappa}\cdot ds^h_{0:t+\kappa}\cdot da^h_{0:t+\kappa},$$

$$\text{where } p_{\theta,0:t+\kappa} = \left(\prod_{u=0}^{t+\kappa-1} p(s^h_{u+1}|s^h_u, a^h_u)\right)\cdot\left(\prod_{u=0}^{t+\kappa}\pi_\theta(a^h_u|s^h_u)\right).$$

Thus, for $\theta_1, \theta_2 \in \mathbb{R}^d$, we have for the high-fidelity only policy gradient,

$$\|\nabla J(\theta_1) - \nabla J(\theta_2)\| = \Bigg\| \sum_{t=0}^{T} \sum_{\kappa=0}^{T-t} \gamma^{t+\kappa} \int \Big(\underbrace{r^h(s^h_{t+\kappa}, a^h_{t+\kappa}) \left[\nabla \log \pi_{\theta_1}(a^h_t \mid s^h_t) - \nabla \log \pi_{\theta_2}(a^h_t \mid s^h_t)\right] p^h_{\theta_1,0:t+\kappa}}_{A}$$

$$+ \underbrace{r^h(s^h_{t+\kappa}, a^h_{t+\kappa}) \nabla \log \pi_{\theta_2}(a^h_t \mid s^h_t) \left(p^h_{\theta_1,0:t+\kappa} - p^h_{\theta_2,0:t+\kappa}\right)}_{B} \Big) ds^h_{0:t+\kappa} da^h_{0:t+\kappa} \Bigg\|$$

$$\leq \sum_{t=0}^{\infty} \sum_{\kappa=0}^{\infty} \gamma^{t+\kappa} \Bigg[\underbrace{\int \Big(\left|r^h(s^h_{t+\kappa}, a^h_{t+\kappa})\right| \left\|\nabla \log \pi_{\theta_1}(a^h_t \mid s^h_t) - \nabla \log \pi_{\theta_2}(a^h_t \mid s^h_t)\right\| p^h_{\theta_1,0:t+\kappa} ds^h_{0:t+\kappa} da^h_{0:t+\kappa}}_{I_1}$$

$$+ \underbrace{\int \left|r^h(s^h_{t+\kappa}, a^h_{t+\kappa})\right| \left\|\nabla \log \pi_{\theta_2}(a^h_t \mid s^h_t)\right\| \left|p^h_{\theta_1,0:t+\kappa} - p^h_{\theta_2,0:t+\kappa}\right| ds^h_{0:t+\kappa} da^h_{0:t+\kappa}}_{I_2} \Bigg] \tag{3}$$

From Assumption 4, we have

$$\begin{aligned} \|A\| &\leq U^h_R L_\Theta \|\theta^1 - \theta^2\| \\ \implies I_1 &\leq U^h_R L_\Theta \|\theta^1 - \theta^2\|. \end{aligned} \tag{4}$$

Further, note that

$$p^h_{\theta_1,0:t+\kappa} - p^h_{\theta_2,0:t+\kappa} = \left(\prod_{u=0}^{t+\kappa-1} p(s^h_{u+1} | s^h_u, a^h_u) \right) \cdot \left(\prod_{u=0}^{t+\kappa} \pi_{\theta^1}(a^h_u | s^h_u) - \prod_{u=0}^{t+\kappa} \pi_{\theta^2}(a^h_u | s^h_u) \right)$$

From Taylor's theorem, $\exists\ \lambda \in (0,1)$ such that for $\tilde{\theta} := \lambda\theta^1 + (1-\lambda)\theta^2$, we have

$$\begin{aligned} \left| \prod_{u=0}^{t+\kappa} \pi_{\theta^1}(a^h_u | s^h_u) - \prod_{u=0}^{t+\kappa} \pi_{\theta^2}(a^h_u | s^h_u) \right| &= \left| \left(\theta^1 - \theta^2\right)^\top \left(\sum_{i=0}^{t+\kappa} \nabla_\theta \pi_{\tilde{\theta}}(a^h_i | s^h_i) \prod_{u=0, u\neq i}^{t+\kappa} \pi_{\tilde{\theta}}(a^h_u | s^h_u) \right) \right| \\ &\leq \|\theta^1 - \theta^2\| \cdot \sum_{i=0}^{t+\kappa} \left\|\nabla_\theta \log \pi_{\tilde{\theta}}(a^h_i | s^h_i)\right\| \cdot \prod_{u=0}^{t+\kappa} \pi_{\tilde{\theta}}(a^h_u | s^h_u) \\ &\leq (t+\kappa+1) \cdot B_\Theta \cdot \|\theta^1 - \theta^2\| \cdot \prod_{u=0}^{t+\kappa} \pi_{\tilde{\theta}}(a^h_u | s^h_u) \qquad \text{(Assumption 3)} \\ \implies I_2 &\leq (t+\kappa+1) \cdot U^h_R \cdot B^2_\Theta \cdot \|\theta^1 - \theta^2\|. \end{aligned} \tag{5}$$

From Equations (3) to (5), we get

$$\|\nabla J(\theta_1) - \nabla J(\theta_2)\| \leq \left(\sum_{t=0}^{T} \sum_{\kappa=0}^{T-t} \left[U^h_R L_\Theta \gamma^{t+\kappa} + U^h_R B^2_\Theta (t+\kappa+1) \gamma^{t+\kappa} \right] \right) \|\theta^1 - \theta^2\|$$

Using the fact that

$$\begin{aligned} \sum_{t=0}^{T} \sum_{\kappa=0}^{T-t} \gamma^{t+\kappa} &= \frac{1 - (T+2)\gamma^{T+1} + (T+1)\gamma^{T+2}}{(1-\gamma)^2}, \\ \sum_{t=0}^{T} \sum_{\kappa=0}^{T-t} (t+\kappa+1)\gamma^{t+\kappa} &= \frac{\left(1 - (T+2)^2\gamma^{T+1} + (T+3)(T+1)\gamma^{T+2}\right)(1-\gamma)}{(1-\gamma)^3} \\ &\quad + \frac{2\gamma\left(1 - (T+2)\gamma^{T+1} + (T+1)\gamma^{T+2}\right)}{(1-\gamma)^3}, \end{aligned}$$

we get that

$$L_T = \frac{1-(T+2)\gamma^{T+1}+(T+1)\gamma^{T+2}}{(1-\gamma)^2}\cdot U_R^h L_\Theta + \frac{\left(1-(T+2)^2\gamma^{T+1}+(T+3)(T+1)\gamma^{T+2}\right)(1-\gamma)+2\gamma\left(1-(T+2)\gamma^{T+1}+(T+1)\gamma^{T+2}\right)}{(1-\gamma)^3}\cdot U_R^h B_\Theta^2.$$

Note that in the limit of infinite time horizon, we recover the Lipschitz bounds proposed by Zhang et al. (2020) for the infinite horizon setting, given by

$$\lim_{T\to\infty} L_T = \frac{1}{(1-\gamma)^2}\cdot U_R^h L_\Theta + \frac{(1+\gamma)}{(1-\gamma)^3}\cdot U_R^h B_\Theta^2.$$

□

To establish the asymptotic almost sure convergence of $\|\nabla J(\theta^{MFPG})\|$ to 0, we also need the following preliminary result, which is an adaptation of a similar result from Zhang et al. (2020), but for the multi-fidelity case.

Lemma 2. *Under Assumptions 1 to 4, the norm of the Multi-Fidelity Policy Gradient estimator is bounded, i.e.,* $\exists\ \hat{\ell} > 0$ *such that* $\|\nabla \hat{J}^{MFPG}(\theta)\| \leq \hat{\ell}\ \forall\ \theta \in \mathbb{R}^d$. *Further, Consider the random variable* W_k*, defined as*

$$W_k := J(\theta_k^{MFPG}) - L_T \hat{\ell}^2 \sum_{j=k}^{\infty} \alpha_j^2.$$

Let $\mathcal{F}_k$ *be a filtration denoting all randomness in* k *iterations. Then, we have that*

$$\mathbb{E}[W_{k+1}|\mathcal{F}_k] \geq W_k + \alpha_k \|\nabla J(\theta_k^{MFPG})\|^2.$$

Proof. We first show the boundedness of the MFPG estimator. We have

$$\left\|\nabla X^{\pi_\theta}_{\tau^h,i}\right\| = \left\|\frac{1}{T}\sum_{t=0}^{T-1} G^h_{i,t}\,\nabla\log\pi_\theta(a^h_{i,t}\mid s^h_{i,t})\right\| \leq \frac{U_R^h B_\Theta}{T}\sum_{t=0}^{T-1}\gamma^t = \frac{U_R^h B_\Theta}{T}\cdot\frac{1-\gamma^T}{1-\gamma}$$

$$\left\|\nabla X^{\pi_\theta}_{\tau^l,i}\right\| = \left\|\frac{1}{T}\sum_{t=0}^{T-1} G^l_{i,t}\,\nabla\log\pi_\theta(a^l_{i,t}\mid s^l_{i,t})\right\| \leq \frac{U_R^l B_\Theta}{T}\sum_{t=0}^{T-1}\gamma^t = \frac{U_R^l B_\Theta}{T}\cdot\frac{1-\gamma^T}{1-\gamma}$$

$$\implies \left\|\frac{1}{N^h}\sum_{i=1}^{N^h}\left[\nabla X^{\pi_\theta}_{\tau^h,i} + c^*\left(\nabla X^{\pi_\theta}_{\tau^l,i} - \mathbb{E}\left[\nabla X^{\pi_\theta}_{\tau^l,i}\right]\right)\right]\right\| \leq \frac{U_R^h B_\Theta}{TN^h}\cdot\frac{1-\gamma^T}{1-\gamma} + 2c^*\frac{U_R^l B_\Theta}{TN^h}\cdot\frac{1-\gamma^T}{1-\gamma} := \hat{\ell}$$

We now show the stochastic ascent property associated with W_k. By Taylor's theorem we have

$$\begin{aligned}
W_{k+1} &= J(\theta_k^{MFPG}) + \left(\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\right)^\top \nabla J(\tilde{\theta_k}^{MFPG}) - L_T \hat{\ell}^2 \sum_{j=k+1}^{\infty} \alpha_j^2 \\
&= J(\theta_k^{MFPG}) + \left(\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\right)^\top \nabla J(\theta_k^{MFPG}) \\
&+ \left(\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\right)^\top \left(\nabla J(\tilde{\theta_k}^{MFPG}) - \nabla J(\theta_k^{MFPG})\right) - L_T \hat{\ell}^2 \sum_{j=k+1}^{\infty} \alpha_j^2 \\
&\geq J(\theta_k^{MFPG}) + \left(\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\right)^\top \nabla J(\theta_k^{MFPG}) - L_T \left\|\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\right\|^2 \\
&- L_T \hat{\ell}^2 \sum_{j=k+1}^{\infty} \alpha_j^2 \\
\implies \mathbb{E}\left[W_{k+1}|\mathcal{F}_k\right] &\geq J(\theta_k^{MFPG}) + \mathbb{E}\left[\theta_{k+1}^{MFPG} - \theta_k^{MFPG}|\mathcal{F}_k\right]^\top \nabla J(\theta_k^{MFPG}) \\
&- L_T \mathbb{E}\left[\left\|\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\right\|^2 |\mathcal{F}_k\right] - L_T \hat{\ell}^2 \sum_{j=k+1}^{\infty} \alpha_j^2 \\
&= J(\theta_k^{MFPG}) + \mathbb{E}\left[\theta_{k+1}^{MFPG} - \theta_k^{MFPG}|\mathcal{F}_k\right]^\top \nabla J(\theta_k^{MFPG}) \\
&- L_T \alpha_k^2 \mathbb{E}\left[\|\nabla J(\theta_k^{MFPG})\|^2 |\mathcal{F}_k\right] - L_T \hat{\ell}^2 \sum_{j=k+1}^{\infty} \alpha_j^2 \\
&\geq J(\theta_k^{MFPG}) + \mathbb{E}\left[\theta_{k+1}^{MFPG} - \theta_k^{MFPG}|\mathcal{F}_k\right]^\top \nabla J(\theta_k^{MFPG}) \\
&- L_T \alpha_k^2 \hat{\ell}^2 - L_T \hat{\ell}^2 \sum_{j=k+1}^{\infty} \alpha_j^2 \\
\implies \mathbb{E}\left[J(\theta_{k+1}^{MFPG})|\mathcal{F}_k\right] &\geq J(\theta_k^{MFPG}) + \mathbb{E}\left[\theta_{k+1}^{MFPG} - \theta_k^{MFPG}|\mathcal{F}_k\right]^\top \nabla J(\theta_k^{MFPG}) - L_T \alpha_k^2 \hat{\ell}^2.
\end{aligned}$$

To show the bound, we have $\mathbb{E}\left[\theta_{k+1}^{MFPG} - \theta_k^{MFPG}|\mathcal{F}_k\right] = \mathbb{E}\left[\alpha_k \nabla \hat{J}^{MFPG}(\theta^{MFPG})|\mathcal{F}_k\right] = \alpha_k \nabla J(\theta_k^{MFPG})$. *This is the crucial step dependent on unbiasedness of control variates, which allows us to conduct this analysis.* Putting this in the above inequality gives us the desired bound. □

Using Lemmas 1 and 2, we can now formally prove Theorem 1, which we restate for the reader's convenience.

Theorem 1. *Under Assumptions 1 to 6, let $\{\theta_k^{MFPG}\}_{k\in\mathbb{Z}^+}$, $\{\theta_k^h\}_{k\in\mathbb{Z}^+}$ be the sequence of policy parameters generated by running the REINFORCE algorithm with MFPG and high-fidelity only policy gradient estimators. Then, for a problem with time horizon T, after N iterations, with step sizes for both iterates taken to be a sequence α_k satisfying $\sum_k \alpha_k = \infty, \sum_k \alpha_k^2 < \infty$, we have*

$$\min_{k\in[N]} \mathbb{E}\left[\|\nabla J(\theta_k^{MFPG})\|^2\right] \leq \frac{J(\theta^*) - J(\theta_1) + \left(\mathbf{1} - \boldsymbol{\rho^2}\right) \frac{\sigma^2 L_T}{2} \sum_{k=1}^N \alpha_k^2}{\sum_{k=1}^N \left(\alpha_k - \frac{\alpha_k^2 L_T}{2}\right)},$$

where L_T is the Lipschitz constant of the high-fidelity policy gradient, established in Lemma 1. Moreover, we recover the corresponding rate for the high-fidelity only policy iterates $\{\theta_k^h\}_{k\in\mathbb{Z}^+}$ by substituting $\rho = 0$. Finally, we have that $\lim_{k\to\infty} \mathbb{E}\left[\|\nabla J(\theta_k^{MFPG})\|\right] = 0$ almost surely.

Proof. Let the MFPG estimator be written as $F^{MFPG}(\theta) := \nabla J(\theta) + \xi$, and the high fidelity only policy gradient estimator be written as $F^h(\theta) := \nabla J(\theta) + \tilde{\xi}$. We analyze the sequences $\{\theta_k^{MFPG}\}_{k\in\mathbb{Z}^+}$ and $\{\theta_k^h\}_{k\in\mathbb{Z}^+}$ generated by the two estimators respectively,

$$\begin{aligned}
\theta_{k+1}^{MFPG} &= \theta_k^{MFPG} + \alpha_k F^{MFPG}, k \in \mathbb{Z}^+, \\
\theta_{k+1}^h &= \theta_k^h + \alpha_k F^h, k \in \mathbb{Z}^+, \\
\theta_1^h &= \theta_1^{MFPG} := \theta_1.
\end{aligned}$$

$$
\begin{aligned}
J(\theta^h_{k+1}) &\geq J(\theta^h_k) + \langle \nabla J(\theta^h_k), \theta^h_{k+1} - \theta^h_k \rangle - \frac{L_T}{2}\|\theta^h_{k+1} - \theta^h_k\|^2 \\
&= J(\theta^h_k) + \alpha_k \langle \nabla J(\theta^h_k), F^h \rangle - \frac{\alpha_k^2 L_T}{2}\|F^h\|^2 \\
&= J(\theta^h_k) + \alpha_k \langle \nabla J(\theta^h_k), \nabla J(\theta^h_k) + \xi_k \rangle - \frac{\alpha_k^2 L_T}{2}\|\nabla J(\theta^h_k) + \xi_k\|^2 \\
&= J(\theta^h_k) + \left(\alpha_k - \frac{\alpha_k^2 L_T}{2}\right)\|\nabla J(\theta^h_k)\|^2 + \left(\alpha_k - \alpha_k^2 L_T\right)\langle \nabla J(\theta^h_k), \xi_k \rangle - \frac{\alpha_k^2 L_T}{2}\|\xi_k\|^2 \\
\implies \left(\alpha_k - \frac{\alpha_k^2 L_T}{2}\right)\|\nabla J(\theta^h_k)\|^2 &\leq J(\theta^h_{k+1}) - J(\theta^h_k) - \left(\alpha_k - \alpha_k^2 L_T\right)\langle \nabla J(\theta^h_k), \xi_k \rangle + \frac{\alpha_k^2 L_T}{2}\|\xi_k\|^2 \\
\implies \sum_{k=1}^N \left(\alpha_k - \frac{\alpha_k^2 L_T}{2}\right)\|\nabla J(\theta^h_k)\|^2 &\leq J(\theta^h_{N+1}) - J(\theta^h_1) - \sum_{k=1}^N \left(\alpha_k - \alpha_k^2 L_T\right)\langle \nabla J(\theta^h_k), \xi_k \rangle \\
&\qquad + \sum_{k=1}^N \frac{\alpha_k^2 L_T}{2}\|\xi_k\|^2 \\
&\leq J(\theta^*) - J(\theta_1) - \sum_{k=1}^N \left(\alpha_k - \alpha_k^2 L_T\right)\langle \nabla J(\theta^h_k), \xi_k \rangle + \sum_{k=1}^N \frac{\alpha_k^2 L_T}{2}\|\xi_k\|^2.
\end{aligned}
$$

Similarly, for the MFPG iterates, we get

$$
\begin{aligned}
\implies \sum_{k=1}^N \left(\alpha_k - \frac{\alpha_k^2 L_T}{2}\right)\|\nabla J(\theta^{MFPG}_k)\|^2 &\leq \\
J(\theta^*) - J(\theta_1) - \sum_{k=1}^N \left(\alpha_k - \alpha_k^2 L_T\right)\langle \nabla J(\theta^{MFPG}_k), \tilde{\xi_k} \rangle &+ \sum_{k=1}^N \frac{\alpha_k^2 L_T}{2}\|\tilde{\xi_k}\|^2.
\end{aligned}
$$

By taking expectation over the randomness for both sequences, from Assumptions 5 and 6 and the fact that the MFPG is an unbiased estimator for true policy gradient, we get

$$
\begin{aligned}
\sum_{k=1}^N \left(\alpha_k - \frac{\alpha_k^2 L_T}{2}\right)\mathbb{E}\left[\|\nabla J(\theta^h_k)\|^2\right] &\leq J(\theta^*) - J(\theta_1) + \frac{\sigma^2 L_T}{2}\sum_{k=1}^N \alpha_k^2 \\
\implies \min_{k\in[T]} \mathbb{E}\left[\|\nabla J(\theta^h_k)\|^2\right] &\leq \frac{J(\theta^*) - J(\theta_1) + \frac{\sigma^2 L_T}{2}\sum_{k=1}^N \alpha_k^2}{\sum_{k=1}^N \left(\alpha_k - \frac{\alpha_k^2 L_T}{2}\right)}
\end{aligned}
$$

$$
\begin{aligned}
\sum_{k=1}^N \left(\alpha_k - \frac{\alpha_k^2 L_T}{2}\right)\mathbb{E}\left[\|\nabla J(\theta^{MFPG}_k)\|^2\right] &\leq J(\theta^*) - J(\theta_1) + \left(1-\rho^2\right)\frac{\sigma^2 L_T}{2}\sum_{k=1}^N \alpha_k^2 \\
\implies \min_{k\in[T]} \mathbb{E}\left[\|\nabla J(\theta^{MFPG}_k)\|^2\right] &\leq \frac{J(\theta^*) - J(\theta_1) + \left(1-\rho^2\right)\frac{\sigma^2 L_T}{2}\sum_{k=1}^N \alpha_k^2}{\sum_{k=1}^N \left(\alpha_k - \frac{\alpha_k^2 L_T}{2}\right)}.
\end{aligned}
$$

We establish the global asymptotic convergence of the multi-fidelity policy gradient using the arguments developed in Zhang et al. (2020). To show that $\lim_{k\to\infty}\|\nabla J(\theta^{MFPG}_k)\| = 0$ almost surely, we first establish that $\liminf_{k\to\infty}\|\nabla J(\theta^{MFPG}_k)\| = 0$. Let J^* be the global optimum, we have that $W_k \leq J^*$ by definition of W_k. From the stochastic ascent property and corresponding bound established in Lemma 2, we have

$$
\begin{aligned}
&\mathbb{E}\left[J^* - W_{k+1}|\mathcal{F}_k\right] \leq (J^* - W_k) - \alpha_k\|\nabla J(\theta^{MFPG}_k)\|^2 \\
\implies &\sum_{k=1}^\infty \alpha_k\|\nabla J(\theta^{MFPG}_k)\|^2 < \infty \text{ almost surely.} \\
&\text{(supermartingale convergence theorem (Robbins \& Siegmund, 1971) as } J^* - W_k \geq 0.) \\
\implies &\liminf_{k\to\infty}\|\nabla J(\theta^{MFPG}_k)\| = 0. \qquad (\because \textstyle\sum_{k=0}^\infty \alpha_k = \infty)
\end{aligned}
$$

Finally, we will establish by contradiction, that $\limsup_{k\to\infty} \|\nabla J(\theta_k^{MFPG})\| = 0$. Assume that $\exists\ \epsilon > 0$ such that $\limsup_{k\to\infty} \|\nabla J(\theta_k^{MFPG})\| = \epsilon$. Thus the set $N_1 = \{\theta_k^{MFPG} : \|\nabla J(\theta_k^{MFPG})\| \geq {}^{2\epsilon}/_{3}\}$ is an infinite set. Because $\liminf_{k\to\infty} \|\nabla J(\theta_k^{MFPG})\| = 0$, the set $N_2 = \{\theta_k^{MFPG} : \|\nabla J(\theta_k^{MFPG})\| \leq {}^{\epsilon}/_{3}\}$ is also an infinite set. We can define the distance between the two sets $D(N_1, N_2) = \inf_{\theta_1 \in N_1} \inf_{\theta_2 \in N_2} \|\theta_1 - \theta_2\|$, which by assumption will be positive. Because both N_1, N_2 have an infinite number of sequence members, there must be an index set $\mathcal{I}$ such that the sequence $\{\theta_k^{MFPG}\}_{k\in\mathcal{I}}$ switches between N_1 and N_2 infinite times, and there must be indices $\{s_i\}_{i\geq 0}, \{t_i\}_{i\geq 0}$ such that $\theta_{s_i} \in N_1, i \geq 0, \theta_{t_i} \in N_2, i \geq 0$, and the iterates $\{\theta_k^{MFPG}\}_{k\in\mathcal{I}} = \{\theta_{s_i+1}^{MFPG}, \dots \theta_{t_i-1}^{MFPG}\}$ are neither in N_1 nor in N_2. Thus, we have

$$\begin{aligned}
\underbrace{\sum_{i\geq 0} D(N_1, N_2)}_{=\infty} = \sum_i \|\theta_{s_i}^{MFPG} - \theta_{t_i}^{MFPG}\| &= \sum_i \left\| \sum_{k=s_i}^{t_i-1} \left(\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\right) \right\| \\
&\leq \sum_{i\geq 0} \sum_{k=s_i}^{t_i-1} \|\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\| \\
&= \sum_{k\in\mathcal{I}} \|\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\| \qquad (6)
\end{aligned}$$

We know that

$$\begin{aligned}
\frac{\epsilon}{3} &\leq \|\nabla J(\theta_k)\| \ \forall\ k \in \mathcal{I} \\
\implies \sum_{k\in\mathcal{I}} \frac{\alpha_k \epsilon^2}{9} &\leq \sum_{k\in\mathcal{I}} \alpha_k \|\nabla J(\theta_k^{MFPG})\|^2 < \infty \\
\implies \sum_{k\in\mathcal{I}} \alpha_k &< \infty \\
\implies \sum_{k\in\mathcal{I}} \mathbb{E}\left[\|\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\|\right] &= \sum_{k\in\mathcal{I}} \alpha_k \mathbb{E}\left[\|\nabla \hat{J}(\theta_k^{MFPG})\|\right] < \infty \qquad \text{(because } \mathbb{E}[\|\nabla \hat{J}\|] \leq \hat{\ell}\text{)} \\
\implies \sum_{k\in\mathcal{I}} \|\theta_{k+1}^{MFPG} - \theta_k^{MFPG}\| &< \infty \text{ almost surely,} \qquad \text{(monotone convergence theorem)}
\end{aligned}$$

which leads us to a contradiction in light of Equation (6), and thus $\limsup_{k\to\infty} \|\nabla J(\theta_k^{MFPG})\| = 0$ almost surely. Thus, we get that $\lim_{k\to\infty} \|\nabla J(\theta_k^{MFPG})\| = 0$.

□

C Complete Experimental Results on Off-Dynamics Tasks

Section 6.3 presents a representative subset of 15 task settings for clarity. Here, we report the complete results across all 39 task settings, spanning diverse types and levels of multi-fidelity dynamics gaps. The overall trends are consistent with those in Section 6.3. Figures 12 and 13 summarize the final evaluation return and AUC metrics, respectively, across the full set of off-dynamics tasks.

D Implementation Details and Hyperparameter Setup

This section provides implementation details and the hyperparameter setup for all evaluated approaches. Details of the off-dynamics task settings described in Section 6.1 are available in the ODRL benchmark (Lyu et al., 2024b). Table 1 summarizes the hyperparameter configurations for all evaluated methods. To ensure reproducibility, we include our code in the supplementary materials and will release the project as open source upon acceptance.

We build our implementation on top of the RL library Stable-Baselines3 (Raffin et al., 2021). For experimentation, we use random seeds {0, 1, 2} for hyperparameter tuning and {3, 4, 5, 6, 7} for evaluation. The

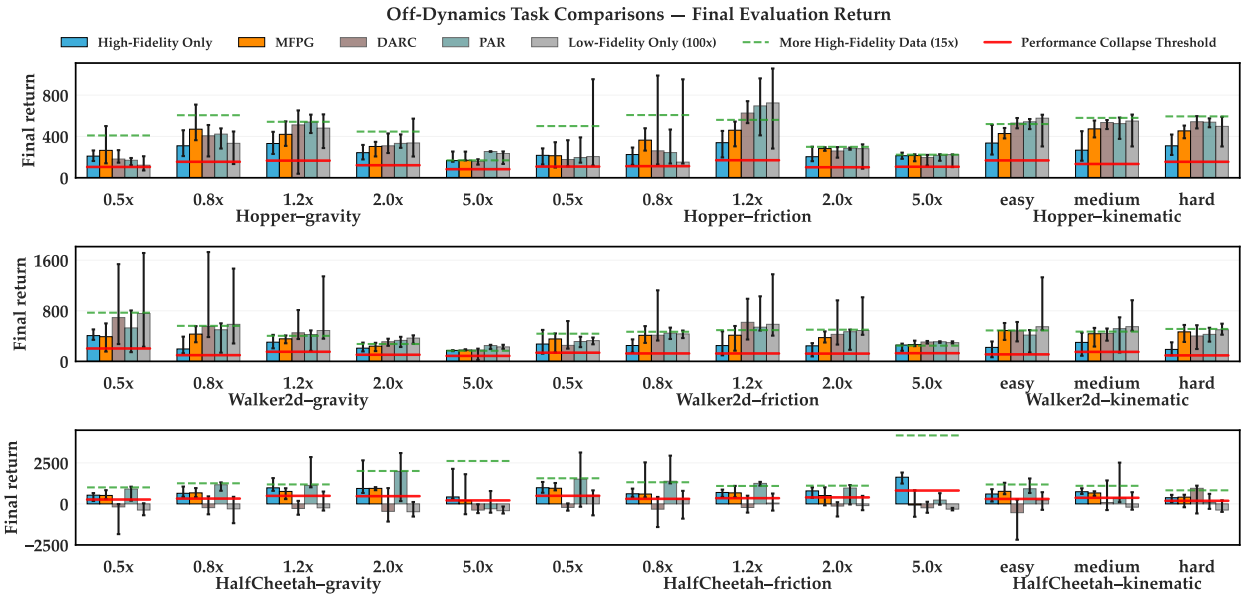

Figure 12: Final evaluation returns of MFPG versus baselines across the full set of off-dynamics task settings described in Section 6.1.

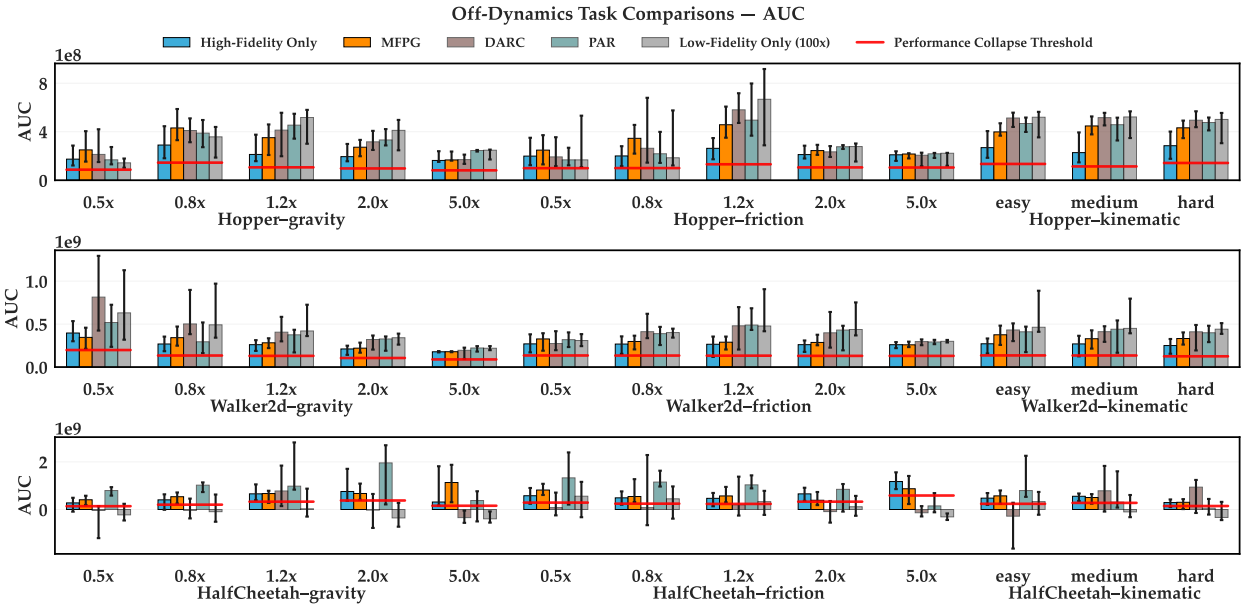

Figure 13: AUC of MFPG versus baselines across the full set of off-dynamics task settings described in Section 6.1.

High-Fidelity Only baseline is tuned using 3 seeds {0, 1, 2}. The resulting configuration serves as the shared backbone for the other methods, following standard practice in the off-dynamics RL literature (Xu et al., 2023; Lyu et al., 2024a). All additional hyperparameters are tuned with 2 seeds {0, 1}, either over more task settings (for the multi-fidelity methods) or longer training steps (for More High-Fidelity Data (15×)). All single-fidelity methods are tuned on the MuJoCo Hopper-v3, Walker2d-v3, and HalfCheetah-v3 tasks with original dynamics, chosen because they represent the midpoint among the varied dynamics settings. Multi-fidelity methods are tuned on the same three tasks under a representative subset of dynamics variations, namely 0.8× and 2.0× gravity. We verified that the tuned hyperparameters generalize well to other variation types and levels (cf. Section C). For the extra sensitive hyperparameters of PAR (reward-augmentation weight) and DARC (the standard deviation of Gaussian noise added to classifier inputs), we additionally tune them on the 0.8× and 2.0× friction variations of the three tasks. We note that this protocol grants the baselines more tuning privilege than MFPG, highlighting the simplicity and minimum tuning overhead of MFPG.

High-Fidelity Only. We implement the standard REINFORCE algorithm (Williams, 1992) with state-value subtraction. Subject to the high-fidelity sample restrictions per policy update described in Section 6.1, we perform grid search over the learning rate $\{1\times10^{-4}, 2\times10^{-4}, 5\times10^{-4}, 7\times10^{-4}, 9\times10^{-4}, 1\times10^{-3}, 2\times10^{-3}\}$, discount factor γ {0.95, 0.97, 0.98, 0.99, 0.995}, advantage normalization {True, False}, maximum gradient norm {0.5, 1, 2}, and state-value loss weight {0.25, 0.5, 1}. We adopt the default network architectures for the policy and state-value networks in Stable-Baselines3 (Raffin et al., 2021). The tuned REINFORCE configuration is then adopted as the shared algorithmic backbone for all multi-fidelity methods. The full hyperparameter configuration, along with those for all other methods, is provided in Table 1.

DARC. DARC (Eysenbach et al.) trains two domain classifiers $q_{\theta_{\text{SAS}}}(\text{high} \mid s_t, a_t, s_{t+1})$ and $q_{\theta_{\text{SA}}}(\text{high} \mid s_t, a_t)$ to predict how likely an observed transition (s_t, a_t, s_{t+1}) or state-action pair (s_t, a_t) comes from the high-fidelity environment. The classifiers are trained by minimizing cross-entropy losses over the replay buffers $\mathcal{D}_{\text{high}}$ and $\mathcal{D}_{\text{low}}$ of high- and low-fidelity samples, respectively:

$$\begin{aligned}\mathcal{L}_{\text{SAS}}(\theta_{\text{SAS}}) &= -\mathbb{E}_{\mathcal{D}_{\text{low}}}\left[\log q_{\theta_{\text{SA}}}(\text{low} \mid s_t, a_t, s_{t+1})\right] - \mathbb{E}_{\mathcal{D}_{\text{high}}}\left[\log q_{\theta_{\text{SAS}}}(\text{high} \mid s_t, a_t, s_{t+1})\right] \\ \mathcal{L}_{\text{SA}}(\theta_{\text{SA}}) &= -\mathbb{E}_{\mathcal{D}_{\text{low}}}\left[\log q_{\theta_{\text{SA}}}(\text{low} \mid s_t, a_t)\right] - \mathbb{E}_{\mathcal{D}_{\text{high}}}\left[\log q_{\theta_{\text{SA}}}(\text{high} \mid s_t, a_t)\right]. \end{aligned} \tag{7}$$

The logits of the trained classifiers are then combined to approximate the dynamics gap, $\log \frac{p(s_{t+1}|s_t,a_t,\text{high})}{p(s_{t+1}|s_t,a_t,\text{low})}$, for each observed (low-fidelity) transition (s_t, a_t, s_{t+1}):

$$\Delta r(s_t, a_t, s_{t+1}) = \log p(\text{high} \mid s_t, a_t, s_{t+1}) - \log p(\text{high} \mid s_t, a_t) - \log p(\text{low} \mid s_t, a_t, s_{t+1}) + \log p(\text{low} \mid s_t, a_t). \tag{8}$$

This estimated dynamics gap is used to augment low-fidelity rewards $r^l_{\text{augment},t}(s^l_t, a^l_t, s^l_{t+1}) = r^l_t(s^l_t, a^l_t, s^l_{t+1}) + \Delta r(s^l_t, a^l_t, s^l_{t+1})$. The augmented rewards penalize the agent for exploiting low-fidelity transitions that are implausible in the high-fidelity environment.

We adopt the standard implementation from the ODRL benchmark (https://github.com/OffDynamicsRL/off-dynamics-rl), cross-validated against the authors' original code (https://github.com/google-research/google-research/tree/master/darc). To adapt DARC to the on-policy REINFORCE algorithm, we retain the original classifier-learning and reward-augmentation mechanisms, substituting only the policy optimization backbone. At each update, classifiers are trained using off-policy samples from the high- and low-fidelity replay buffers, as in the original implementation, by minimizing the cross-entropy losses in Eq. (7). The learned classifiers are then used to augment the rewards, cf., Eq. (8), for low-fidelity, on-policy samples, which are subsequently employed to compute the REINFORCE updates and learn the state-value function.

Building on the shared REINFORCE backbone used by the High-Fidelity Only baseline, we further tune several DARC-specific parameters to which the method is more sensitive: the standard deviation of Gaussian noise added to classifier inputs as a regularizer (an effective stabilization trick noted by the authors (Eysenbach et al.)) {0.5, 1, 2}, and the number of warm-start, high-fidelity environment steps {2000, 10000, 50000} prior to classifier learning and reward augmentation. For other hyperparameters—such as classifier architecture, learning rate, and batch size—we adopt the default values from the original paper (Eysenbach et al.). The resulting classifier training curves are stable.

PAR. Similar in spirit to DARC, PAR (Lyu et al., 2024a) also estimates the dynamics gap of low-fidelity transitions for reward augmentation. However, instead of domain classification, PAR adopts a representation-learning approach: it trains a state encoder $z_{1,t} = f_\zeta(s_t)$ and a state–action encoder $z_{2,t} = g_\nu(z_{1,t}, a_t)$ by minimizing a latent dynamics consistency loss over the *high-fidelity* replay buffer:

$$\mathcal{L}(\zeta, \nu) = \mathbb{E}_{(s_t^h, a_t^h, s_{t+1}^h)\sim\mathcal{D}_{\text{high}}} \left[\| g_\nu(f_\zeta(s_t^h), a_t^h) - \texttt{NoGradient}(f_\zeta(s_{t+1}^h)) \|^2 \right], \tag{9}$$

which essentially learns a dynamics model for the high-fidelity environment in a latent space. The trained encoders are then used to estimate dynamics mismatch for low-fidelity transitions and augment the rewards:

$$r^l_{\text{augment},t}(s_t^l, a_t^l, s_{t+1}^l) = r_t^l(s_t^l, a_t^l, s_{t+1}^l) - \beta \cdot \| g_\nu(f_\zeta(s_t^l), a_t^l) - f_\zeta(s_{t+1}^l) \|^2, \tag{10}$$

where β is a weighting hyperparameter. Furthermore, unlike DARC, which trains its policy and critics only on low-fidelity samples with augmented rewards, PAR trains its policy and critics on both high- and low-fidelity samples, with reward augmentation applied to the latter.

We adopt the standard implementation from the ODRL benchmark (https://github.com/OffDynamicsRL/off-dynamics-rl), cross-validated against the authors' original code (https://github.com/dmksjfl/PAR). Analogous to our adaptation of DARC to the on-policy REINFORCE algorithm, we adapt PAR by retaining its original representation-learning and reward-augmentation mechanisms while substituting only the policy optimization backbone. At each update, we draw off-policy, high-fidelity samples from the replay buffer to train the encoders by minimizing Eq. (9). The trained encoders are then applied to augment rewards for on-policy low-fidelity samples, as in Eq. (10). Unlike DARC, which relies solely on augmented low-fidelity samples for policy and value function learning, PAR leverages both on-policy high-fidelity and augmented low-fidelity samples for REINFORCE policy updates and state-value function learning, consistent with the original PAR algorithm.

On top of the shared REINFORCE backbone, we further tune the reward-augmentation weight β in Eq. (10) over the grid {0.1, 0.5, 1, 2}, as PAR is particularly sensitive to this parameter; cf. Fig. 14. The original PAR (Lyu et al., 2024a) employs a *per-task* weight for this reason, and we follow the same convention, reporting the selected β for each task in Table 1. We note, however, that this requirement introduces a computationally expensive adaptation overhead, unlike other algorithms, which do not require per-task tuning. For other hyperparameters—such as encoder architecture, learning rate, and batch size—we adopt the defaults from the original paper (Lyu et al., 2024a). The resulting encoder training curves are stable.

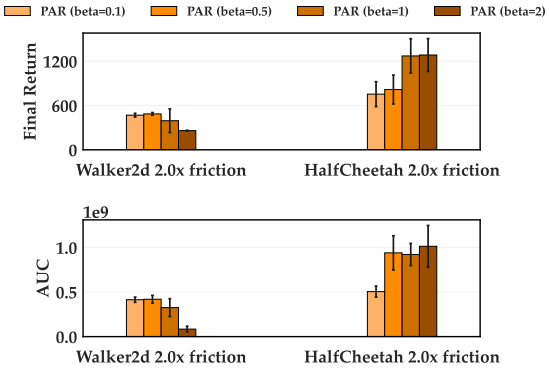

Figure 14: Sweep of the reward-augmentation weight β for PAR on two tasks. The relative performance across β values varies substantially between tasks.

MFPG. Building on the shared REINFORCE backbone, we perform a grid search over the exponential moving-average weight η_{ma} in Eq. (2) using values {0.92, 0.95, 0.99}. MFPG is found to be relatively robust to this hyperparameter, as illustrated in Section 6.5. In off-dynamics task settings, we drop the control variate term $c(X_{\tau^l}^{\pi_\theta} - \hat{\mu}^l)$ whenever the estimated correlation coefficient for the current batch, i.e., $\hat{\rho}(X_{\tau^h}^{\pi_\theta}, X_{\tau^l}^{\pi_\theta})_{\text{new}}$ in Eq. (2), is negative. Since high- and low-fidelity samples should in principle be positively correlated in off-dynamics settings, a negative correlation suggests either low-quality samples or noise in correlation estimation. In practice, such cases occur only rarely. This engineering trick helps mitigate noise in estimating the control variate coefficient, which can arise from extreme low-fidelity samples. Developing more principled estimation schemes remains an orthogonal direction for future work.

Low-Fidelity Only (100×). The Low-Fidelity Only baseline serves as an ablation of high-fidelity samples from the multi-fidelity approaches. Accordingly, we adopt the same hyperparameter configuration as the

shared REINFORCE backbone, except that each policy and state-value function update is performed using abundant low-fidelity data in place of limited high-fidelity samples.

More High-Fidelity Data (15×). To ensure fair benefit from the additional high-fidelity samples, we tune the sensitive hyperparameters for the oracle baseline More High-Fidelity Data (15×) separately from the shared REINFORCE backbone. Specifically, we tune the learning rate over the grid $\{5 \times 10^{-4}, 7 \times 10^{-4}, 9 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}, 4 \times 10^{-3}\}$ and the batch size (i.e., samples per policy update) over the grid {1024, 2048, 3072, 4096, 6144}. Because of the expensive, longer training time for this baseline, we first run all configurations for 3 million steps, then select the best 7 configurations and extend them to 15 million steps to determine the final configuration. Other hyperparameters are kept consistent with the shared REINFORCE backbone, as they are relatively insensitive.

Low-fidelity data amount. In baseline comparisons, we assume that low-fidelity samples are cheap to generate, and all multi-fidelity methods may supplement limited high-fidelity data with up to 100× additional low-fidelity samples per policy update. While PAR has been empirically observed to benefit from more low-fidelity data in some task settings (Lyu et al., 2024a), we also found it to be sensitive to the amount of low-fidelity data in certain cases. To maximize baseline performance, we additionally tune the ratio of low- to high-fidelity samples per policy update over the grid $\{10\times, 20\times, 50\times, 100\times\}$ for PAR and DARC. For DARC, performance is generally maximized with 100× low-fidelity samples. For PAR, performance is maximized across tasks at 20×. The Low-Fidelity Only baseline is assigned 100× low-fidelity samples to match the budget. We implement parallelized low-fidelity environments, each generating 10× low-fidelity samples per policy update; hence, DARC and Low-Fidelity Only employ 10 environments, while PAR uses 2 environments. For MFPG, we did not tune the amount of low-fidelity samples, since MFPG is in theory stable with respect to low-fidelity sample amounts, so long as the low-fidelity sample amount is sufficient to estimate the low-fidelity sample mean $\hat{\mu}^l$. Our empirical results in Fig. 8 support this point. Therefore, instead of using 10 parallel low-fidelity environments (which would exceed the 100× budget, given that MFPG also requires 1× correlated low-fidelity samples), we adopt 9 parallel environments in all main results, corresponding to 90× low-fidelity samples (plus 1× correlated low-fidelity samples). In some auxiliary variance and sensitivity analysis in Sections 6.2 and 6.5, we use 100× (uncorrelated) low-fidelity samples; we specify the low-fidelity sample amount separately in those sections. Finally, we emphasize that DARC and PAR receive more tuning budget than MFPG, underscoring the simplicity and low adaptation overhead of MFPG.

Complete hyperparameter configuration. Table 1 reports the full hyperparameter configuration for all methods in the baseline comparison results. As noted above, the High-Fidelity Only configuration serves as the shared backbone for other methods, with algorithm-specific hyperparameters listed on top of this backbone. Blue entries indicate hyperparameters that replace the corresponding values in the backbone.

Table 1: Hyperparameter configurations for evaluated methods.

High-Fidelity Only (REINFORCE backbone)	
Optimizer	Adam (Kingma, 2014)
Learning rate	7×10^{-4}
High-fidelity batch size	100
Discount factor γ	0.97
State-value loss weight vf_coef	1.0
Maximum gradient norm	1.0
Policy network	(64, 64)
State value network	(64, 64)
Nonlinearity	Tanh
DARC	
Classifier network	(256, 256)
Classifier nonlinearity	ReLU
Classifier Gaussian input noise std. σ	0.5
Classifier optimizer	Adam
Classifier learning rate	3×10^{-4}
Classifier batch size	128
Replay buffer size (classifier)	1×10^{6}
Low-fidelity data amount / policy update	100× (relative to high-fidelity data)
Warm-up high-fidelity steps	2000
PAR	
Encoder network	(256, 256)
Encoder nonlinearity	ELU
Representation dimension	256
Encoder optimizer	Adam
Encoder learning rate	3×10^{-4}
Encoder batch size	128
Replay buffer size (encoder)	1×10^{6}
Polyak averaging coefficient for encoder updates	0.995 (for the target encoder network)
Low-fidelity data amount / policy update	20×
Reward augmentation weight β	0.5 (Hopper), 0.1 (Walker2d), 1.0 (HalfCheetah)
MFPG	
Low-fidelity data amount / policy update	90× uncorrelated + 1× correlated
Exponential moving-average weight η_{ma}	0.95
Low-Fidelity Only (100×)	
Low-fidelity data amount / policy update	100× (no high-fidelity data)
More High-Fidelity Data (15×)	
Learning rate	9×10^{-4}
High-fidelity batch size	3072