MULTI-FIDELITY POLICY GRADIENTS STABILIZE DATA-SCARCE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Policy gradient methods for on-policy reinforcement learning (RL) can become unstable when expensive, scarce target-domain data yield noisy gradient estimates. We address this challenge by complementing limited *high-fidelity (HF)* target-domain data with abundant, cheap, but biased *low-fidelity (LF)* data, e.g., from a simplified simulator. Most existing methods directly optimize biased objectives based on LF data. In contrast, the recently introduced multi-fidelity policy gradient (MFPG) framework uses LF data solely to construct a control variate that reduces variance and improves HF data efficiency without biasing the policy gradient estimator. However, published work on MFPG is limited to REINFORCE on small-scale simulation tasks. We develop MFPG for modern actor-critic learning in GPU-parallel simulation and on a physical robot. Our analysis and experiments show that naive extensions to proximal policy optimization (PPO) can lose cross-fidelity correlation or inflate variance. Our MFPG-PPO addresses these failures by redesigning the sampling, advantage estimation, and control variate construction to preserve cross-fidelity correlation, and by monitoring estimator uncertainty to prevent variance inflation. We also introduce a budget-aware MFPG-PPO to divide a fixed sampling budget among high- and low-fidelity data sources. Across simulated robot locomotion tasks of varying LF-to-HF transfer difficulty and HF data budgets, MFPG-PPO improves upon PPO trained on HF data alone in nearly all settings, and consistently matches the performance of PPO trained with $16\times$ more HF data on the hardest task at the smallest HF budgets. In contrast, most baselines that use LF data perform well only where direct LF-to-HF transfer succeeds. MFPG-PPO enables stable learning on a physical Franka arm using only 4 real-robot episodes per update and no human demonstrations.

1 INTRODUCTION

On-policy reinforcement learning (RL) with policy gradient methods underpins advances in areas ranging from language-model post-training (Ouyang et al., 2022) to fine-tuning vision-language-action models (Lu et al., 2025) for robot control. Yet in many real-world settings, target-domain data are expensive to collect and therefore scarce, e.g., a physical robot may collect only a handful of rollouts per policy update. In such data-scarce regimes, standard on-policy methods can break down because their policy gradient estimates become too noisy to support stable learning.

We study how to enable stable on-policy RL in data-scarce regimes, ultimately using as few as 4 real-robot episodes per update, or target-domain data batches up to $512\times$ smaller than in standard GPU-simulation RL pipelines (Rudin et al., 2022). The key is access to *multi-fidelity* data: we assume that the target environment (a physical system or high-accuracy simulator) provides limited, expensive high-fidelity (HF) samples, whereas a simplified simulator provides abundant, inexpensive, but biased low-fidelity (LF) samples. Exploiting LF data requires care, however, and can easily bias policy updates and degrade performance in the target HF environment.

Existing approaches use LF data in different ways. Off-dynamics RL methods (Eysenbach et al., 2021; Lyu et al., 2024) learn to correct LF rewards for the dynamics mismatch, and sim-and-real co-training (Wei et al., 2025) mixes LF and HF data. However, *reliability* is a major challenge. Most of these methods still directly optimize biased objectives based on LF data, so their performance in the target HF environment is largely tied to how well policies learned from LF data transfer.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

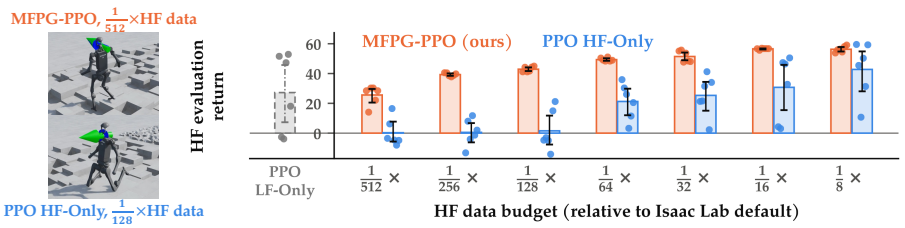


Figure 1: **In a simulated humanoid robot locomotion task on rough terrain, MFPG-PPO (orange) *scales monotonically* with the HF data budget and, where PPO trained on HF data alone (blue) collapses, matches the mean return of PPO trained with $16\times$ more HF data.** HF data budgets are set by the number of parallel HF training environments and expressed as fractions of the Isaac Lab default (4096 environments). At each HF budget, MFPG-PPO uses the same amount of HF data as PPO HF-Only, plus abundant LF data from a simplified simulator. PPO LF-Only (gray) trains on abundant LF data alone and is evaluated in the HF environment. Bars show mean HF evaluation returns of final policies across 6 seeds, with 95% seed-bootstrapped confidence intervals. Dots denote individual seeds. The snapshots show the MFPG-PPO policy walking across the terrain, whereas the PPO HF-Only policy remains on its knees despite using $4\times$ more HF training data.

The recent multi-fidelity policy gradient (MFPG) framework (Liu et al., 2026) offers a principled alternative. Here, LF samples are used to construct a control variate (Nelson, 1987)—a correlated auxiliary random variable with a known mean—for the HF policy gradient estimator. The MFPG estimator remains unbiased with respect to the HF objective, and the policy gradient surrogate variance shrinks as the cross-fidelity data correlation grows. MFPG is therefore robust to LF data biases.

However, the MFPG construction of Liu et al. (2026) is limited to the classic REINFORCE algorithm (Williams, 1992) and small-scale, single-environment simulation tasks. How MFPG interacts with modern actor-critic learning and GPU-parallel simulation (Mittal et al., 2025; Tao et al., 2024), and how it translates to real-robot learning, remain unresolved. An earlier preprint (Liu et al., 2025) directly substituted proximal policy optimization (PPO) for REINFORCE. The resulting method did not consistently improve over the REINFORCE instantiation and sometimes even underperformed single-fidelity PPO. Hence, an effective extension requires more than algorithm substitution.

We address this gap with two contributions: (i) *MFPG-PPO, a multi-fidelity PPO algorithm.* Through analysis and experiments, we show how naive PPO instantiations of MFPG can lose cross-fidelity correlation or inflate variance by orders of magnitude. We address these failures by redesigning the sampling, advantage estimation, and control variate construction to preserve cross-fidelity correlation, and by monitoring estimator uncertainty to prevent variance inflation. Preserving correlation comes at the price of potential bias, a tradeoff we characterize analytically. (ii) *A budget-aware MFPG-PPO.* When HF and LF samples incur different costs, we allocate a fixed per-iteration sampling budget across fidelities, guided by the variance-minimizing allocation framework of Peherstorfer et al. (2016).

We test MFPG-PPO in GPU-parallel simulation (Mittal et al., 2025; Tao et al., 2024) and on a physical robot arm. Across simulation tasks and HF data budgets, we compare with 5 baselines that use LF data in different ways and with PPO and soft actor-critic (SAC) (Haarnoja et al., 2018) trained on HF data alone. MFPG-PPO is the only method in our comparison that improves over PPO trained on HF data alone in nearly all settings, is never distinguishably worse, and scales monotonically with more HF data. Most baselines that use LF data perform well only where direct LF-to-HF transfer succeeds. On the hardest task at the smallest HF budgets, MFPG-PPO consistently matches the mean return of PPO trained with $16\times$ more HF data, cf. Fig. 1. Finally, MFPG-PPO enables stable RL on a Franka arm with only 4 episodes per policy update and no human demonstrations.

2 Related Work

LF simulation can support policy learning for an HF target domain through system identification (Ljung, 1998; Ramos et al., 2019), zero-shot transfer with domain randomization (Tobin et al., 2017; Peng et al., 2018), adapting simulation-trained priors with scarce HF data (Smith et al., 2022;

Levy et al., 2026), and combining multi-fidelity data within policy updates. We focus on the last category. A complementary line of multi-fidelity RL research (Cutler et al., 2015; Marco et al., 2017; Ryou et al., 2026; Bhola et al., 2023; Sifaou & Simeone, 2025) studies which fidelity to query at each training stage rather than how to combine fidelities within a policy update.

Three mechanisms for combining multi-fidelity data are particularly relevant. *Pooling* multi-fidelity data is a simple and effective recipe for sim-and-real co-training in imitation learning (Maddukuri et al., 2025; Wei et al., 2025; Lei et al., 2026) and in RL (Di-Castro et al., 2021). Off-dynamics RL *corrects* for estimated dynamics mismatch in LF data, e.g., through reward penalties from domain classifiers (Eysenbach et al., 2021) or representation mismatch (Lyu et al., 2024), filtering by value discrepancy (Xu et al., 2023), support extension (Le Pham Van et al., 2024), or imitation-based transfer (Guo et al., 2024), with extensions to offline and hybrid offline-online settings (Liu et al., 2022; Niu et al., 2022; 2025; Lyu et al., 2025; Guo et al., 2026).

Pooling and correction optimize biased objectives based on LF data, whereas multi-fidelity *control variates* use LF data to reduce variance in HF estimates. Control variates are a classical tool in single-fidelity RL (Greensmith et al., 2004; Gu et al., 2017; Cheng et al., 2020). Recently, Khairy & Balaprakash (2024) develop multi-fidelity control variates for state-action value estimation in tabular Markov decision processes (MDPs), while Liu et al. (2026) propose a policy gradient framework based on REINFORCE with function approximation. Building on these advances, we develop multi-fidelity control variates for actor-critic learning in GPU-parallel simulation and on a real robot.

3 PRELIMINARIES

Multi-fidelity MDPs. We consider a *high-fidelity (HF)* and a *low-fidelity (LF)* environment, modeled as infinite-horizon discounted MDPs $\mathcal{M}^f := (S, A, P^f, r^f, \Delta_{s_I}, \gamma)$ for $f \in \{\text{high}, \text{low}\}$. Both share the state and action spaces S and A, initial-state distribution Δ_{s_I}, and the discount factor $\gamma \in [0, 1)$, but may differ in their next-state distributions $P^f(\cdot \mid s, a)$ and reward functions $r^f \colon S \times A \to \mathbb{R}$, where $s \in S$ denotes the current state and $a \in A$ the action. We write $\tau \sim (\mathcal{M}^f, \pi_\theta)$ for a trajectory $\tau := (s_0, a_0, s_1, a_1, \dots)$ generated by a stochastic policy $\pi_\theta(a \mid s)$ with parameters $\theta \in \mathbb{R}^d$ in $\mathcal{M}^f$, and seek the policy that maximizes the expected discounted HF return $J^{\text{high}}(\theta) := \mathbb{E}_{\tau \sim (\mathcal{M}^{\text{high}}, \pi_\theta)}\big[\sum_{t=0}^{\infty} \gamma^t \, r^{\text{high}}(s_t, a_t)\big]$. We assume the LF simulator can be reset to recorded HF states for sampling correlated multi-fidelity trajectories, cf. Section 4.1.

Policy gradient surrogates. Policy gradient methods estimate $\nabla_\theta J^{\text{high}}(\theta)$ from trajectories $\tau \sim (\mathcal{M}^{\text{high}}, \pi_\theta)$. We form estimates by averaging $\nabla_\theta X(\theta)$, where X is a scalar surrogate statistic defined per trajectory or transition, depending on the algorithm. REINFORCE (Williams, 1992) uses the *trajectory-level* statistic $X_\tau(\theta) := \sum_{t=0}^{H-1} \gamma^t \, \log \pi_\theta(a_t \mid s_t) \, \big(\hat{G}_t - b(s_t)\big)$, where H is the episode length, $\hat{G}_t := \sum_{l=t}^{H-1} \gamma^{l-t} \, r^{\text{high}}(s_l, a_l)$ the Monte Carlo return, and b a baseline. During differentiation, $\hat{G}_t$ and b are fixed. Episodic REINFORCE requires collecting complete episodes.

Proximal policy optimization (PPO) (Schulman et al., 2017) operates on *transitions* by replacing returns $\hat{G}_t$ with advantage estimates $\hat{A}_t$ commonly computed via generalized advantage estimation (GAE) (Schulman et al., 2015). Using a learned value function (critic) V, GAE sums temporal-difference residuals over a T-step rollout within an episode as $\hat{A}_t := \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \delta_{t+l}$, where $\delta_t := r^{\text{high}}(s_t, a_t) + \gamma V(s_{t+1}) - V(s_t)$ and $\lambda \in [0, 1]$ sets the bias-variance tradeoff. T sets the GAE horizon. Bootstrapping with state values at rollout boundaries allows updates from parallel, short *rollout segments*. PPO reuses batches across epochs of minibatch updates on the surrogate $X_t(\theta) := \min\big(\omega_t \hat{A}_t, \, \text{clip}(\omega_t, 1-\epsilon, 1+\epsilon) \, \hat{A}_t\big)$, where $\omega_t := \pi_\theta(a_t \mid s_t) / \pi_{\theta_{\text{old}}}(a_t \mid s_t)$ is the importance ratio relative to the data-collecting policy and ϵ the clipping range. These features make PPO standard in GPU-based RL (Rudin et al., 2022) but depart from the MFPG-REINFORCE setting.

Multi-fidelity policy gradients. Following Gu et al. (2017) and Liu et al. (2026), we target the variance of the scalar surrogate statistic $X(\theta)$ before differentiation. Liu et al. (2026) form a control-variate estimator (Nelson, 1987) on REINFORCE's trajectory-level statistic X_τ over N^{high} HF trajectories: $\nabla_\theta J^{\text{high}}(\theta) \approx (N^{\text{high}})^{-1} \sum_{i=1}^{N^{\text{high}}} \nabla_\theta \big[X_i^{\text{high}}(\theta) + c \, \big(X_i^{\text{low}}(\theta) - \hat{\mu}^{\text{low}}(\theta)\big)\big]$, where $c \in \mathbb{R}$ is the control variate coefficient and $c \, (X_i^{\text{low}} - \hat{\mu}^{\text{low}})$ is a corrector for X_i^{high}. Each X_i^{low} is computed on the LF trajectory *coupled* to its HF counterpart: the LF simulator is reset to the HF initial state

and the two rollouts share the same action-noise samples via a reparameterization trick (Kingma & Welling, 2013) adapted to RL. The mean $\hat{\mu}^{\text{low}}(\theta) := (N^{\text{low}})^{-1} \sum_{j=1}^{N^{\text{low}}} X_{\tilde{\tau}_j^{\text{low}}}(\theta)$ is estimated from $N^{\text{low}} \gg N^{\text{high}}$ additional, independently sampled LF trajectories $\tilde{\tau}_j^{\text{low}}$. Intuitively, X^{low} is an LF "twin" of the HF statistic. Formally, for $Z(c) := X^{\text{high}} + c\,(X^{\text{low}} - \mu^{\text{low}})$ with $\mu^{\text{low}} := \mathbb{E}[X^{\text{low}}]$, Liu et al. (2026) establish (i) *unbiasedness*, $\mathbb{E}[Z(c)] = \mathbb{E}[X^{\text{high}}]$ for fixed c, since the LF corrector has zero mean, and (ii) *variance reduction*, with the optimal $c^* = -\operatorname{Cov}(X^{\text{high}}, X^{\text{low}})/\operatorname{Var}(X^{\text{low}})$ yielding $\operatorname{Var}\big(Z(c^*)\big) = \big(1-\rho^2\big)\operatorname{Var}(X^{\text{high}})$, where ρ is the correlation between coupled X^{high} and X^{low}. Variance reduction grows with $|\rho|$, and c^* is estimated from data across iterations.

4 Approach

We develop MFPG-PPO by addressing two challenges that arise when extending the control variate construction of Liu et al. (2026) to PPO, then introduce a budget-aware MFPG-PPO.

Challenge 1: What to pair as $(X^{\text{high}}, X^{\text{low}})$, and how to keep them correlated. Liu et al. (2026) pair the complete-episode REINFORCE statistic X_τ, which is unbiased but has two limitations. First, PPO collects short rollouts from desynchronized environments, which need not contain any complete episode. Second, since the multi-fidelity environments have different dynamics, coupled trajectories can diverge over long episodes despite shared initial states and action noise, weakening or destabilizing their correlation ρ on which the variance reduction $(1-\rho^2)$ relies. One trajectory may even terminate while the other continues.

Challenge 2: How much to trust the estimates. The correlation ρ and the coefficient c^* are unknown and estimated from scarce HF and coupled LF data. However, a poorly estimated coefficient can make the estimator *noisier* than using HF data alone. Moreover, PPO reuses each batch across policy updates, so coefficient estimates must account for changes in the surrogate statistics. The control variate should therefore be applied cautiously only when the evidence supports its use.

4.1 MFPG-PPO

Workflow. Each iteration of MFPG-PPO follows four steps, cf. Algorithm 1 in Appendix A.4. First, *multi-fidelity data collection* gathers HF, coupled LF, and uncoupled LF batches of transitions. Within each batch, k indexes independent rollouts, e.g., from parallel environments, and t their steps. Second, *constructing surrogate statistics* computes the PPO statistics defined in Section 3 for all three batches. Let $\mathcal{P}$ index the paired statistics $(X_{k,t}^{\text{high}}, X_{k,t}^{\text{low}})$ and $\mathcal{U}$ the uncoupled LF statistics. Third, *coefficient fitting* estimates and smooths the coefficient c from the pairs in $\mathcal{P}$ for the estimator

$$L^{\text{MF}}(\theta) := \frac{1}{|\mathcal{P}|} \sum_{(k,t)\in\mathcal{P}} \Big[X_{k,t}^{\text{high}}(\theta) + c\,\big(X_{k,t}^{\text{low}}(\theta) - \hat{\mu}^{\text{low}}(\theta)\big)\Big], \quad \hat{\mu}^{\text{low}}(\theta) := \frac{1}{|\mathcal{U}|} \sum_{(k,t)\in\mathcal{U}} X_{k,t}^{\text{low}}(\theta). \tag{1}$$

The policy gradient estimate is $\nabla_\theta L^{\text{MF}}$, with gradients flowing through all three terms, including $\hat{\mu}^{\text{low}}$, and the advantages and c held fixed. Finally, *monitoring* decides whether to apply the control variate or to fall back to a PPO update on HF data alone by setting $c = 0$. The first two steps address Challenge 1 and the last two Challenge 2.

Multi-fidelity data collection. To preserve correlation between the paired statistics $(X_{k,t}^{\text{high}}, X_{k,t}^{\text{low}})$, we limit divergence between HF and coupled LF rollouts by (i) sharing controllable sources of randomness and (ii) periodically re-synchronizing the coupled LF rollouts to recorded HF states, cf. lines 3 to 6 of Algorithm 1. Figure 2 illustrates the multi-fidelity data collection in two settings.

When HF and LF environments are GPU-parallel simulations of different accuracy, we collect short T-step rollouts from N^{high} HF, N^{high} coupled LF, and $N^{\text{low}} \gg N^{\text{high}}$ uncoupled LF environments. Figure 2(a) illustrates one representative rollout from each batch. We first collect the HF rollouts, then reset each coupled LF environment to its counterpart's rollout start state, including task variables such as terrain. We assume the policy is *reparameterizable* (Kingma & Welling, 2013) as $a_t^f = g_\theta(s_t^f, \varepsilon_t)$ with a deterministic map g_θ and noise ε_t drawn from a fixed distribution independent of θ. Coupled rollouts share the noise sample at corresponding steps, with the policy evaluated at each fidelity's own state. When one environment of a pair terminates while the other continues,

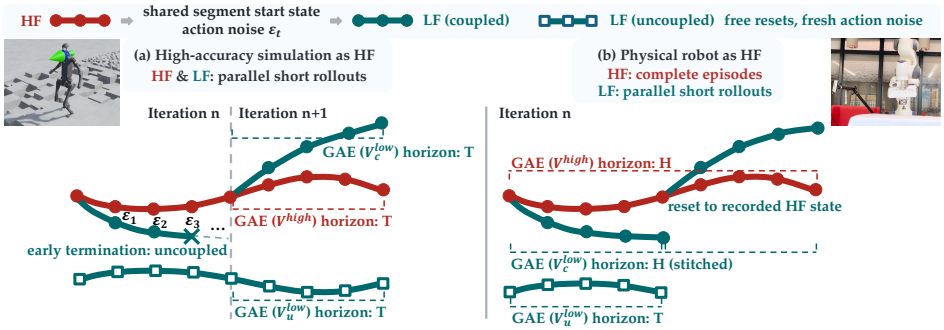


Figure 2: **MFPG-PPO collects three batches of transitions.** Representative HF, coupled LF, and uncoupled LF rollouts are shown with red circles, filled teal circles, and hollow teal squares. **(a)** When HF and LF are simulations of different accuracy, each batch contains parallel T-step rollouts per iteration. **(b)** When the HF environment is a robot collecting H-step episodes ($H > T$), coupled LF simulation rollouts restart from recorded HF states every T steps. In both cases, the LF simulation runs only parallel T-step rollouts, while the coupled LF GAE horizon matches the HF horizon, T in (a) and H in (b). In (b), we stitch the coupled LF rollouts of each episode for GAE.

the remaining steps of the continuing environment are uncoupled and excluded from $\mathcal{P}$ until the next re-synchronization, i.e., an environment reset or the start of the next iteration. The rollout length T sets both PPO's rollout horizon and the re-synchronization interval. Uncoupled LF environments draw their action noise independently and reset freely to $s_0^{\text{low}} \sim \Delta_{s_I}$.

When the HF environment is a physical robot and the LF environment is simulation, a robot instead collects K complete H-step episodes sequentially ($H > T$). We divide each episode into T-step segments and launch a coupled LF rollout from each segment's *measured* initial state, cf. Fig. 2(b). These rollouts run in parallel after HF collection and re-synchronize frequently to HF states.

Constructing paired statistics $(X_{k,t}^{\text{high}}, X_{k,t}^{\text{low}})$**.** We construct PPO statistics by computing GAE with critics V^{high}, $V_{\text{c}}^{\text{low}}$, and $V_{\text{u}}^{\text{low}}$ for the HF, coupled LF, and uncoupled LF batches, respectively. The separate LF critics accommodate the different state distributions of the coupled and uncoupled LF rollouts. We match the GAE horizons of the coupled LF rollouts to the HF rollouts to preserve cross-fidelity correlation, cf. lines 7 and 8 of Algorithm 1 and Fig. 2.

In simulation, all three batches use a T-step GAE horizon matching their rollout length, cf. Fig. 2(a). On a real robot, we match the H-step HF GAE horizon by stitching the coupled LF rollout segments of the same episode, cf. Fig. 2(b). Uncoupled LF rollouts use an ordinary T-step GAE horizon in both settings. The critics, in contrast, are trained only on transitions their own environment produced, with targets that never span a re-synchronized state, so that each remains a valid value estimate of its own environment. Since the coupled and uncoupled LF statistics are computed along different trajectory distributions, the LF correction in Eq. (1) can introduce bias, a cost of preserving correlation. We analyze the tradeoff for this design in Appendix A.2 and show that re-synchronizing the uncoupled LF rollouts as well would remove the bias but leave part of the variance irreducible.

Fitting the coefficient c**.** We fit c by estimating the optimal value c^* of Section 3 from the pairs $(X_{k,t}^{\text{high}}, X_{k,t}^{\text{low}})$, $(k,t) \in \mathcal{P}$, and smooth the estimates with an exponential moving average (EMA). We re-fit c as the policy is updated across minibatches, cf. lines 9 and 11 of Algorithm 1.

Since the pairs are scarce (Challenge 2), one option, taken by the released implementation of Liu et al. (2026), treats all pairs $(k,t) \in \mathcal{P}$ as independent samples and fits *per step*, $\hat{c}_{\text{step}} := -\widehat{\text{Cov}}(X_{k,t}^{\text{high}}, X_{k,t}^{\text{low}}) / \widehat{\text{Var}}(X_{k,t}^{\text{low}})$. The per-step fit multiplies the sample size and stabilizes the estimate, but it can miss cross-step correlation relevant to variance reduction. On our real-robot manipulation task, cf. Fig. 4(c), key events such as grasping occur asynchronously in the two fidelities, and the per-step correlation is weak ($\hat{\rho} \approx 0.17$). Fitting on the *episode sums* $B_k^f := \sum_t X_{k,t}^f$, i.e., $\hat{c}_{\text{ep}} := -\widehat{\text{Cov}}(B_k^{\text{high}}, B_k^{\text{low}}) / \widehat{\text{Var}}(B_k^{\text{low}})$, instead retains the cross-step correlation ($\hat{\rho} \approx 0.6$). In simulation, per-step correlations are substantially higher, cf. Fig. 11 in Appendix A.7. We therefore

fit c per step in simulation and on episode sums on the robot. The fitting unit trades off sample size against the cross-step correlation it captures, as formalized in Lemma 1, with proof in Appendix A.1.

Lemma 1 (Per-step fitting omits cross-step covariance)**.** *For a fixed policy, consider paired statistics* $\{(X_{k,t}^{\text{high}},\ X_{k,t}^{\text{low}})\}_{k=1,\,t=1}^{K,\,H}$ *from K i.i.d. episodes of H steps, with the episode sums* B_k^{high} *and* B_k^{low} *defined above. In the moments below, we drop the episode index k. Assume the per-step means* $\mathbb{E}[X_t^{\text{high}}]$ *and* $\mathbb{E}[X_t^{\text{low}}]$ *are constant in t. Treating* $\hat{\mu}^{\text{low}}$ *as fixed and under the regularity conditions of Appendix A.1, the variance of the estimator in Eq.* (1) *is minimized by* $c^* = -\operatorname{Cov}(B^{\text{high}}, B^{\text{low}})/\operatorname{Var}(B^{\text{low}})$*, where* $\operatorname{Cov}(B^{\text{high}}, B^{\text{low}}) = \sum_{t=1}^{H}\sum_{t'=1}^{H}\operatorname{Cov}(X_t^{\text{high}}, X_{t'}^{\text{low}})$*. The per-step fit estimates* $c_{\text{step}} = -\sum_t \operatorname{Cov}(X_t^{\text{high}}, X_t^{\text{low}}) \,/\, \sum_t \operatorname{Var}(X_t^{\text{low}})$*, retaining only the same-step terms. Hence, $c_{\text{step}} = c^*$ when cross-step covariances vanish, but not in general.*

Monitoring the control variate. Since the coefficient c is fitted on scarce data, we apply the control variate only when two checks pass, cf. lines 9 and 12 of Algorithm 1. Otherwise, the update falls back to HF-only by setting $c = 0$. First, *statistical evidence*. The correlation between the paired statistics must be distinguishable from zero at a chosen confidence level, which we set to 90%. In simulation, a bootstrap confidence interval (Efron, 1992) on the fitted c must exclude zero. On the real robot, bootstrapping from scarce episode pairs can be unstable. We instead use a heuristic check on the estimated correlation $\hat{\rho}$ between episode sums $(B_k^{\text{high}}, B_k^{\text{low}})$. Since the distribution of $\hat{\rho}$ is skewed, we average its Fisher-z transform $\operatorname{atanh}(\hat{\rho})$ (Fisher, 1921), which is approximately normal, over a sliding window of recent iterations and require the resulting normal confidence interval to exclude zero, cf. Appendix A.5. Second, *empirical variance reduction*. The corrected statistic $X_{k,t}^{\text{high}} + c\left(X_{k,t}^{\text{low}} - \hat{\mu}^{\text{low}}\right)$ must have lower sample variance than $X_{k,t}^{\text{high}}$ alone over the pairs.

4.2 Budget-Aware MFPG-PPO

The estimator in Eq. (1) so far treats LF data as effectively free with $N^{\text{low}} \gg N^{\text{high}}$. When LF sampling is not free, we ask: *given a fixed per-iteration budget and heterogeneous per-sample costs, how many samples should be drawn from each fidelity to minimize the variance of the control variate estimator?* Writing $w^{\text{high}}, w^{\text{low}}$ for the per-sample costs (per T-step rollout), p for the budget, and $m^{\text{high}}, m^{\text{low}}$ for the HF and *total* coupled and uncoupled LF sample counts, the question becomes

$$\min_{c,\, m^{\text{high}},\, m^{\text{low}}} \operatorname{Var}\left[L_{\text{all}}^{\text{MF}}(\theta)\right] \quad \text{s.t.} \quad w^{\text{high}} m^{\text{high}} + w^{\text{low}} m^{\text{low}} = p, \quad m^{\text{low}} \geq m^{\text{high}} > 0, \qquad (2)$$

a two-fidelity case of the multi-fidelity Monte Carlo (MFMC) allocation problem (Peherstorfer et al., 2016). Here, $L_{\text{all}}^{\text{MF}}$ is Eq. (1) with the baseline mean $\hat{\mu}_{\text{all}}^{\text{low}}$ over *all* coupled and uncoupled LF samples, as in the MFMC estimator. Let $\nu := m^{\text{low}}/m^{\text{high}}$ denote the ratio of total LF to HF sample counts. Under the idealized sampling model in Appendix A.3, the variance-minimizing allocation (Peherstorfer et al., 2016, Thm. 3.4) satisfies $\nu^* = \max\left\{1,\ \sqrt{(w^{\text{high}}/w^{\text{low}})\,\rho^2/(1-\rho^2)}\right\}$ and $m^{\text{high},*} = p/(w^{\text{high}} + w^{\text{low}}\,\nu^*)$. Appendix A.3 gives a derivation and details on flooring the real-valued solution to integers within the budget. For $\nu^* > 1$, the coefficient remains the c^* of Section 3. At $\nu^* = 1$, the LF correction vanishes. The ratio ν^* grows as LF rollouts become cheaper or more strongly correlated with their HF counterparts.

Online budget allocation. As in multi-fidelity optimization under uncertainty (Ng & Willcox, 2014; Hyun et al., 2026), we adapt the allocation as the policy changes. We track ρ with the smoothing scheme of Section 4.1, re-solve Eq. (2) every iteration *before* data collection, and set $N^{\text{high}} = m^{\text{high},*}$ and $N^{\text{low}} = m^{\text{low},*} - m^{\text{high},*}$ after flooring to integers. Coefficient fitting and monitoring carry over unchanged, and the policy objective becomes $L_{\text{all}}^{\text{MF}}$. Whenever $\nu^* = 1$, i.e., LF data do not warrant more samples than HF data, the iteration collects only coupled HF-LF data, which preserves our ability to estimate ρ, and the update falls back to HF-only PPO, cf. Appendix A.3.

5 Experiments

Evaluation criteria. A practitioner rarely knows in advance how informative the LF data will be. We therefore evaluate two criteria. (i) *Monotone scaling*, i.e., an approach's performance should improve monotonically as more HF data become available, until it plateaus. (ii) *Reliability*, i.e.,

compared to training with limited HF data alone, an approach should perform better when the LF data are informative and no worse when they are misleading.

5.1 Experimental Setup

Simulation setup and HF data budgets. We evaluate locomotion on rough and flat terrain with the Unitree H1 humanoid and the ANYmal-D quadruped in Isaac Lab (Mittal et al., 2025). We sweep a *sample-scarcity* axis by varying the number of parallel HF environments N^{high} from 8 to 512, up to $512\times$ fewer than the Isaac Lab default of 4096, while fixing $N^{\text{low}} = 4096$ uncoupled LF environments. The LF simulation simplifies the robot collision model, coarsens the physics time step, reduces the number of physics solver iterations, and disables self-collisions. Since exploration from a random initialization is difficult with scarce HF data, all on-policy methods that use HF data begin from a shared policy briefly pre-trained with PPO in the LF environment. SAC starts from its own LF-pretrained policy with matched LF training return. Unless otherwise stated, we evaluate each final policy deterministically over 2048 HF episodes and report mean returns with 95% bootstrap confidence intervals across 6 seeds. Appendix A.6 reports details.

Real-robot setup. We evaluate cube picking with a physical Franka arm and ManiSkill simulation (Tao et al., 2024) as the HF and LF environments, respectively. The task is harder in the HF environment because the robot's side-mounted camera loses sight of the object during contact (around 69% of the steps), and grasp detection relies on gripper width and may be delayed. By contrast, the simulator provides ground-truth object and grasp states. Friction and controller stiffness also differ between fidelities. We run 3 random seeds per real-robot method, each for 80 iterations of 4 episodes. MFPG-PPO collects extra coupled LF rollouts and 8192 uncoupled 20-step LF rollouts.

Baselines and shared tuning protocol. We evaluate two MFPG-PPO variants. *MFPG-PPO-FR* in Section 4.1 uses a *fixed* HF-to-LF sampling *ratio*, and *MFPG-PPO-BA* in Section 4.2 performs *budget allocation* online. Baselines cover several uses of LF data. *PPO HF-Only* uses no LF data. *Co-training* (Wei et al., 2025) pools HF and LF data. *DARC* (Eysenbach et al., 2021) and *PAR* (Lyu et al., 2024) correct LF rewards. *PPO LF-Only* uses no HF data. *MFPG-REINFORCE* (Liu et al., 2026) uses control variates. To isolate the effect of LF data use from the algorithm backbone, we adapt DARC and PAR to the PPO backbone of MFPG-PPO, following the practice of Liu et al. (2026), and Co-training to RL on the same backbone. We additionally compare to an off-policy baseline *SAC HF-Only* (Haarnoja et al., 2018). Each baseline matches the data budget of MFPG-PPO-FR at each fidelity it consumes. Appendix A.6 describes the baselines and the shared hyperparameter tuning protocol. Appendix A.7 reports extended comparisons.

5.2 How does MFPG-PPO scale with HF data scarcity?

The bar chart in Fig. 1 compares MFPG-PPO-FR with PPO HF-Only at matched HF data budgets on H1 Rough, the hardest task in our suite. PPO HF-Only collapses to near-zero return at HF budgets of $\frac{1}{128}\times$ and below, whereas MFPG-PPO-FR still attains a mean return around 40 at $\frac{1}{256}\times$. At each of the 3 smallest HF budgets, MFPG-PPO-FR matches or exceeds the mean return of PPO HF-Only trained with $16\times$ more HF data while exhibiting substantially lower cross-seed variability. PPO LF-Only transfers poorly on half of the seeds, and even the seeds that transfer are brittle, with roughly twice the cross-episode return variability of MFPG-PPO-FR.

5.3 How do different uses of LF data scale across tasks and HF budgets?

Figure 3 reports returns relative to PPO HF-Only across tasks and HF budgets from $\frac{1}{64}\times$ to $\frac{1}{8}\times$ the Isaac Lab default. LF-to-HF transfer is *easy* on ANYmal-D, where PPO LF-Only matches the best methods, but *hard* on H1, where transfer is poor and seed-sensitive. Baselines that exploit LF data greedily follow this pattern: DARC and Co-training achieve among the largest gains on ANYmal-D but fall below or provide no distinguishable gain over PPO HF-Only on H1. Both MFPG-REINFORCE and SAC HF-Only generally outperform PPO HF-Only on H1 Rough but underperform in most settings on the other 3 tasks.

MFPG-PPO leverages LF data reliably. MFPG-PPO-FR outperforms PPO HF-Only in 14 of the 16 task-budget settings and is never distinguishably worse. The remaining 2, on H1 Flat, are not distinguishable because PPO HF-Only already performs among the best. MFPG-PPO-BA, assuming

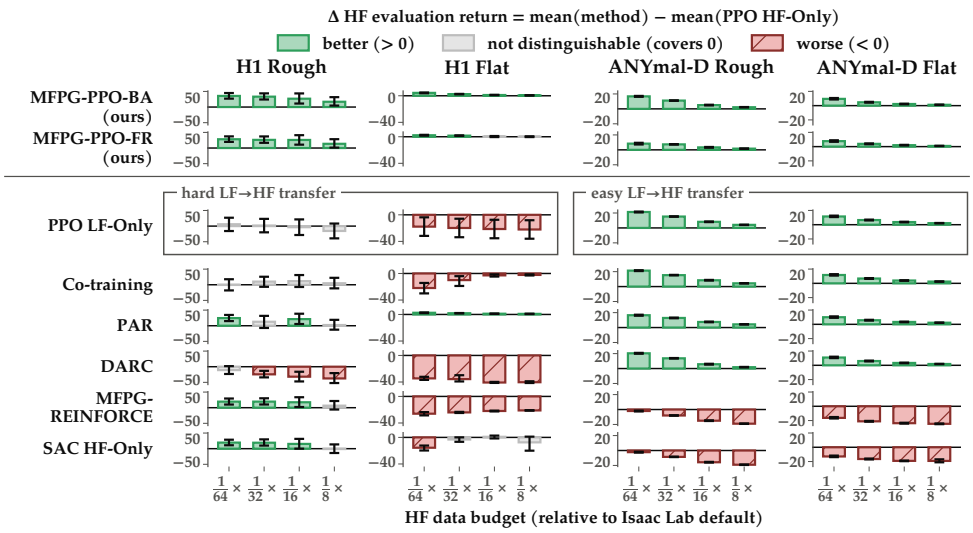


Figure 3: **MFPG-PPO is the only method in our comparison that improves upon PPO HF-Only in nearly every setting, is never distinguishably worse, and scales monotonically with more HF data.** Bars denote differences in mean HF evaluation return relative to PPO HF-Only. Error bars denote 95% bootstrapped confidence intervals for the mean of the difference across 6 seeds.

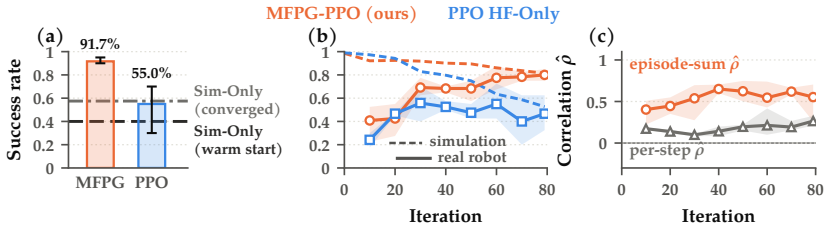


Figure 4: **Neither limited real-world data nor abundant simulation data alone yield a good real-robot policy, whereas MFPG-PPO succeeds by combining them.** Bars and curves denote means across 3 seeds. Whiskers and shaded bands show their ranges. **(a)** Final success rate of the deterministic policy over 60 real-robot episodes (20 per seed). Horizontal lines denote the shared warm-start checkpoint performance and that of a simulation-trained policy well past convergence (100% simulation success), each evaluated over 40 real-robot episodes. **(b)** Training success rate on the robot (solid lines denote running averages over 40 episodes per seed) and in uncoupled LF simulation (dashed lines). **(c)** Pearson correlation between the coupled LF and HF paired statistics, computed on episode sums (4 pairs) and per step (240 pairs), averaged over the past 10 iterations.

a cost ratio $w^{\text{high}}/w^{\text{low}} = 10$, outperforms PPO HF-Only in all 16 settings. Its HF budget labels are nominal, since it allocates a sampling budget online. The gains of MFPG-PPO are largest on the hardest task, H1 Rough, at the smallest HF budgets. This reliability comes with a tradeoff. In easy LF-to-HF settings, the greedier methods gain more from the LF data.

PAR also improves in 14 settings and is never distinguishably worse than PPO HF-Only, but it scales less consistently with HF data. On H1 Rough, its mean return decreases at 2 of the 3 budget increases. MFPG-PPO-FR's mean return instead rises with more HF data in every task, plateauing only at the largest budget on H1 Rough, as shown in Fig. 8 in Appendix A.7.

5.4 Can MFPG-PPO learn stably on a real robot?

We test MFPG-PPO on the Franka cube-picking task in Section 5.1. Simulation data are near-free relative to real-robot data, so we use MFPG-PPO-FR. We compare it with PPO HF-Only, initializing both from the same simulation-trained policy, which has near-perfect simulation success but retains exploration noise. Figure 4(a) reports the final evaluation results. Starting from 40.0% real-robot

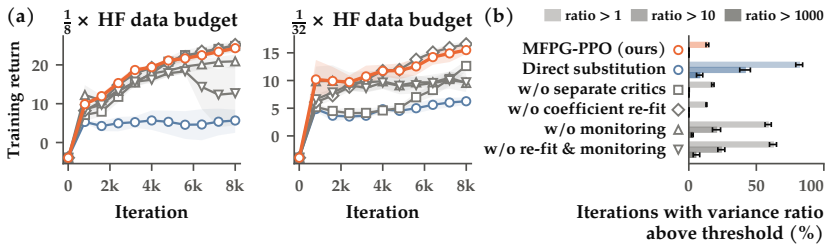


Figure 5: Curves and bars denote means over 6 seeds. Bands and whiskers denote seed-bootstrapped 95% confidence intervals for the means. **(a)** HF training return on H1 Rough. **(b)** Fraction of training iterations whose variance ratio $\mathrm{Var}[X^{\mathrm{high}}_{k,t} + c\,(X^{\mathrm{low}}_{k,t} - \hat{\mu}^{\mathrm{low}})] \,/\, \mathrm{Var}[X^{\mathrm{high}}_{k,t}]$, averaged over minibatches *before* the monitoring check, exceeds 1, 10, and 1000 at the $\frac{1}{8}\times$ HF budget.

success, MFPG-PPO-FR reaches 91.7%, compared with 55.0% for PPO HF-Only, and every MFPG-PPO-FR seed (90–95%) exceeds every PPO HF-Only seed (30–70%). Figure 4(b) shows a consistent pattern during training: the real-robot success rate of MFPG-PPO-FR trends above PPO HF-Only after 30 iterations. Meanwhile, the same policies' success rates in simulation, cf. the dashed lines in Fig. 4(b), decline toward their real-robot performance, consistent with adaptation to the hardware.

Figure 4(c) shows the fitting-unit tradeoff for the coefficient c in Lemma 1. The same policy tends to grasp earlier in simulation than in the real world, cf. Fig. 11 in Appendix A.7. On the same coupled samples, the episode-sum correlation is higher but fluctuates more than the per-step correlation.

5.5 Ablations: naive control variate construction amplifies variance

We compare MFPG-PPO-FR with component ablations and Direct substitution, which substitutes PPO for REINFORCE in MFPG, mirroring the preprint (Liu et al., 2025). Direct substitution lacks the estimator-side machinery of Section 4.1 (separate critics, monitoring, per-minibatch coefficient re-fitting) but shares our data-collection scheme. In Fig. 5(a), Direct substitution yields the lowest training return at both HF budgets on H1 Rough. Per-minibatch coefficient re-fitting and monitoring act *jointly*. Removing either alone at most lowers the return, while removing both causes learning to collapse at the $\frac{1}{8}\times$ budget. Removing separate critics has little effect at $\frac{1}{8}\times$ but lowers returns at $\frac{1}{32}\times$. Figure 5(b) shows that all variants without monitoring *amplify* variance in most iterations (variance ratio above 1), with ratios sometimes exceeding 1000. For the full MFPG-PPO, the ratio exceeds 1 in only a small fraction of iterations. Minibatches failing the variance reduction check fall back to HF-only PPO updates. Appendix A.7 reports further diagnostics.

6 Conclusions and Discussion

We introduced MFPG-PPO and its budget-aware variant to stabilize data-scarce on-policy RL. Through designs that preserve cross-fidelity correlation and monitor estimator uncertainty, MFPG-PPO enables stable learning with per-update target-domain data batches as small as 4 real-robot episodes or up to $512\times$ smaller than in common GPU-simulation RL pipelines (Rudin et al., 2022). Unlike methods that directly optimize biased objectives based on LF data, MFPG-PPO uses LF data to construct a variance-reducing control variate. In our comparison across tasks and HF budgets, MFPG-PPO is the only method that improves upon PPO trained on HF data alone in nearly all settings, is never distinguishably worse, and scales monotonically with more HF data.

Three directions call for future research. First, re-synchronizing coupled LF rollouts to recorded HF states while uncoupled LF rollouts evolve freely can introduce bias in Eq. (1) if the two LF statistics differ in mean. Re-synchronizing both LF batches would remove this bias but leave the variance associated with the re-synchronization states *irreducible*, cf. Lemma 2 in Appendix A.2. Our construction keeps the full variance reducible, but doing so without bias remains an open challenge. Second, raising cross-fidelity correlation, e.g., through representation alignment (Lei et al., 2026), could strengthen variance reduction. Third, real-to-sim tools that turn scene scans into interactive simulators (Jain et al., 2025; Team, 2026) could ease the construction of coupled LF simulators.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods, clean and reformat datasets, interpret results, assist in the writing of proofs, and design or provide feedback on research methodology or experiments. Additionally, we used generative AI tools for sourcing/searching for information and to create or modify scientific figures or images, summarize or analyze existing literature, edit this paper to improve readability, draft parts of this paper, and suggest a structure for this paper.

We did not use generative AI tools to generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, or discover research topics or identify gaps. Using generative AI tools to assist with translation, formulate questions for surveys or interviews, and support qualitative and thematic data analysis is not applicable to this work.

We have reviewed all AI-assisted work. In particular, we reviewed and tested all AI-written code, checked every AI-assisted proof step, and verified all reported experimental results against the experiment logs. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Algorithm 1 presents the complete MFPG-PPO algorithm, and Appendix A.5 specifies its monitoring checks for the control variate estimator. Appendix A.6.1 reports the shared hyperparameter tuning protocol and the final hyperparameters, Appendix A.6.2 describes the implementation of MFPG-PPO and all baselines, and Appendix A.6.3 details the simulation and real-robot tasks, their HF and LF environments, and the training and evaluation protocols. Appendix A.1 proves Lemma 1, Appendix A.2 analyzes why the uncoupled LF rollouts are not re-synchronized, and Appendix A.3 derives the budget allocation. We will release our software implementation upon publication.

REFERENCES

Sid Bharthulwar, Stone Tao, and Hao Su. Staggered environment resets improve massively parallel on-policy reinforcement learning. *Advances in Neural Information Processing Systems*, 38: 133342–133375, 2026.

Sahil Bhola, Suraj Pawar, Prasanna Balaprakash, and Romit Maulik. Multi-fidelity reinforcement learning framework for shape optimization. *Journal of Computational Physics*, 482:112018, 2023.

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.

Shirli Di-Castro, Dotan Di Castro, and Shie Mannor. Sim and real: Better together. *Advances in Neural Information Processing Systems*, 34:6868–6880, 2021.

Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics: Methodology and distribution*, pp. 569–593. Springer, 1992.

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*, 2021. URL https://openreview.net/forum?id=eqBwg3AcIAK.

Ronald A Fisher. On the" probable error" of a coefficient of correlation deduced from a small sample. *Metron*, 1:3–32, 1921.

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. URL https://openreview.net/forum?id=SJ3rcZcxl.

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.

Yihong Guo, Yu Yang, Pan Xu, and Anqi Liu. Mobody: Model-based off-dynamics offline reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 6985–7008, 2026.

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.

Jaeyub Hyun, Anirban Chaudhuri, Karen E. Willcox, and H. Alicia Kim. Robust topology optimization of continuum structures under material uncertainties using multifidelity monte carlo. *Computer Methods in Applied Mechanics and Engineering*, 451:118714, 2026. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2025.118714. URL https://www.sciencedirect.com/science/article/pii/S0045782525009867.

Arhan Jain, Mingtong Zhang, Kanav Arora, William Chen, Marcel Torne, Muhammad Zubair Irshad, Sergey Zakharov, Yue Wang, Sergey Levine, Chelsea Finn, et al. Polaris: Scalable real-to-sim evaluations for generalist robot policies. *arXiv preprint arXiv:2512.16881*, 2025.

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

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

Linh Le Pham Van, Hung The Tran, and Sunil Gupta. Policy learning for off-dynamics rl with deficient support. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, pp. 1093–1100, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

Yu Lei, Minghuan Liu, Abhiram Maddukuri, Zhenyu Jiang, and Yuke Zhu. A mechanistic analysis of sim-and-real co-training in generative robot policies. *arXiv preprint arXiv:2604.13645*, 2026.

Jacob Levy, Tyler Westenbroek, Kevin Huang, Fernando Palafox, Patrick Yin, Shayegan Omidshafiei, Dong-Ki Kim, Abhishek Gupta, and David Fridovich-Keil. Simulation distillation: Pretraining world models in simulation for rapid real-world adaptation. *arXiv preprint arXiv:2603.15759*, 2026.

Jinxin Liu, Zhang Hongyin, and Donglin Wang. DARA: Dynamics-aware reward augmentation in offline reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9SDQB3b68K.

Xinjie Liu, Cyrus Neary, Kushagra Gupta, Christian Ellis, Ufuk Topcu, and David Fridovich-Keil. Multi-fidelity policy gradient algorithms. *arXiv preprint arXiv:2503.05696v1*, 2025. URL https://arxiv.org/abs/2503.05696v1.

Xinjie Liu, Cyrus Neary, Kushagra Gupta, Wesley A. Suttle, Christian Ellis, ufuk topcu, and David Fridovich-Keil. A multi-fidelity control variate approach for policy gradient estimation. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=zAo0L7Dcqt. J2C Certification.

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

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.

Jiafei Lyu, Chenjia Bai, Jing-Wen Yang, Zongqing Lu, and Xiu Li. Cross-domain policy adaptation by capturing representation mismatch. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33638–33663. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/lyu24a.html.

Jiafei Lyu, Mengbei Yan, Zhongjian Qiao, Runze Liu, Xiaoteng Ma, Deheng Ye, Jing-Wen Yang, Zongqing Lu, and Xiu Li. Cross-domain offline policy adaptation with optimal transport and dataset constraint. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LRrbD8EZJl.

Abhiram Maddukuri, Zhenyu Jiang, Lawrence Yunliang Chen, Soroush Nasiriany, Yuqi Xie, Yu Fang, Wenqi Huang, Zu Wang, Zhenjia Xu, Nikita Chernyadev, Scott Reed, Ken Goldberg, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Sim-and-Real Co-Training: A Simple Recipe for Vision-Based Robotic Manipulation. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.109.

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.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, Lukasz Wawrzyniak, Milad Rakhsha, Alain Denzler, Eric Heiden, Ales Borovicka, Ossama Ahmed, Iretiayo Akinola, Abrar Anwar, Mark T. Carlson, Ji Yuan Feng, Animesh Garg, Renato Gasoto, Lionel Gulich, Yijie Guo, M. Gussert, Alex Hansen, Mihir Kulkarni, Chenran Li, Wei Liu, Viktor Makoviychuk, Grzegorz Malczyk, Hammad Mazhar, Masoud Moghani, Adithyavairavan Murali, Michael Noseworthy, Alexander Poddubny, Nathan Ratliff, Welf Rehberg, Clemens Schwarke, Ritvik Singh, James Latham Smith, Bingjie Tang, Ruchik Thaker, Matthew Trepte, Karl Van Wyk, Fangzhou Yu, Alex Millane, Vikram Ramasamy, Remo Steiner, Sangeeta Subramanian, Clemens Volk, CY Chen, Neel Jawale, Ashwin Varghese Kuruttukulam, Michael A. Lin, Ajay Mandlekar, Karsten Patzwaldt, John Welsh, Huihua Zhao, Fatima Anes, Jean-Francois Lafleche, Nicolas Moënne-Loccoz, Soowan Park, Rob Stepinski, Dirk Van Gelder, Chris Amevor, Jan Carius, Jumyung Chang, Anka He Chen, Pablo de Heras Ciechomski, Gilles Daviet, Mohammad Mohajerani, Julia von Muralt, Viktor Reutskyy, Michael Sauter, Simon Schirm, Eric L. Shi, Pierre Terdiman, Kenny Vilella, Tobias Widmer, Gordon Yeoman, Tiffany Chen, Sergey Grizan, Cathy Li, Lotus Li, Connor Smith, Rafael Wiltz, Kostas Alexis, Yan Chang, David Chu, Linxi "Jim" Fan, Farbod Farshidian, Ankur Handa, Spencer Huang, Marco Hutter, Yashraj Narang, Soha Pouya, Shiwei Sheng, Yuke Zhu, Miles Macklin, Adam Moravanszky, Philipp Reist, Yunrong Guo, David Hoeller, and Gavriel State. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025. URL https://arxiv.org/abs/2511.04831.

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

Leo W. T. Ng and Karen E. Willcox. Multifidelity approaches for optimization under uncertainty. *International Journal for Numerical Methods in Engineering*, 100(10):746–772, 2014. doi:

https://doi.org/10.1002/nme.4761. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4761.

Haoyi Niu, shubham sharma, Yiwen Qiu, Ming Li, Guyue Zhou, Jianming HU, and Xianyuan Zhan. When to trust your simulator: Dynamics-aware hybrid offline-and-online reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 36599–36612. Curran Associates, Inc., 2022. doi: 10.52202/068431-2652. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/ed3cd2520148b577039adfade82a5566-Paper-Conference.pdf.

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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Optimal model management for multifidelity monte carlo estimation. *SIAM Journal on Scientific Computing*, 38(5):A3163–A3194, 2016.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 3803–3810. IEEE, 2018.

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

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on robot learning*, pp. 91–100. PMLR, 2022.

Gilhyun Ryou, Geoffrey Wang, and Sertac Karaman. Multi-fidelity reinforcement learning for time-optimal quadrotor re-planning. *The International Journal of Robotics Research*, 45(4):645–669, 2026.

Gianluca Sabatini, Chenhao Li, and Marco Hutter. Bridging the gap: Enabling soft actor critic for high performance legged locomotion. *arXiv preprint arXiv:2605.24975*, 2026.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

Clemens Schwarke, Mayank Mittal, Nikita Rudin, David Hoeller, and Marco Hutter. Rsl-rl: A learning library for robotics research. *arXiv preprint arXiv:2509.10771*, 2025.

Houssem Sifaou and Osvaldo Simeone. Multi-fidelity hybrid reinforcement learning via information gain maximization. *arXiv preprint arXiv:2509.14848*, 2025.

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. IEEE, 2022.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-kai Chan, et al. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

World Labs Team. Atlas: A world model for spatial intelligence. *World Labs Blog*, 2026. https://www.worldlabs.ai/blog/atlas.

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.

Adam Wei, Abhinav Agarwal, Boyuan Chen, Rohan Bosworth, Nicholas Pfaff, and Russ Tedrake. Empirical analysis of sim-and-real cotraining of diffusion policies for planar pushing from pixels. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5625–5632. IEEE, 2025.

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

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.

A Appendix

A.1 Proof of Lemma 1

Lemma 1 holds under the following regularity conditions: (i) the per-step statistics have finite second moments; (ii) $\mathrm{Var}(B^{\mathrm{low}}) > 0$ (if $\mathrm{Var}(B^{\mathrm{low}})$ vanishes, the variance in question does not depend on c); (iii) every step of every episode is coupled, so the pair set of Eq. (1) contains exactly one pair per (k, t) and $|\mathcal{P}| = KH$.

Proof. By condition (iii), $|\mathcal{P}| = KH$. Grouping the sum by episode,

$$L^{\mathrm{MF}}(\theta) = \frac{1}{KH}\sum_{k=1}^{K}\sum_{t=1}^{H}\Big[X_{k,t}^{\mathrm{high}} + c\big(X_{k,t}^{\mathrm{low}} - \hat{\mu}^{\mathrm{low}}\big)\Big] = \frac{1}{KH}\sum_{k=1}^{K}\big(B_k^{\mathrm{high}} + c\,B_k^{\mathrm{low}}\big) \;-\; c\,\hat{\mu}^{\mathrm{low}}.$$

Since $\hat{\mu}^{\mathrm{low}}$ contributes no variance, and since the episodes are independent and identically distributed,

$$\begin{aligned}\mathrm{Var}\left(L^{\mathrm{MF}}\right) &= \frac{1}{K^2H^2}\sum_{k=1}^{K}\mathrm{Var}\left(B_k^{\mathrm{high}} + c\,B_k^{\mathrm{low}}\right)\\ &= \frac{1}{KH^2}\Big[\,\mathrm{Var}(B^{\mathrm{high}}) + 2c\;\mathrm{Cov}(B^{\mathrm{high}}, B^{\mathrm{low}}) + c^2\,\mathrm{Var}(B^{\mathrm{low}})\Big],\end{aligned}$$

a strictly convex quadratic in c whose unique minimizer is $c^* = -\,\mathrm{Cov}(B^{\mathrm{high}}, B^{\mathrm{low}})/\,\mathrm{Var}(B^{\mathrm{low}})$. Covariance is linear in each of its two arguments, so

$$\begin{aligned}\mathrm{Cov}(B^{\mathrm{high}}, B^{\mathrm{low}}) &= \mathrm{Cov}\left(\sum_{t=1}^{H} X_t^{\mathrm{high}}, \sum_{t'=1}^{H} X_{t'}^{\mathrm{low}}\right)\\ &= \sum_{t=1}^{H}\mathrm{Cov}\left(X_t^{\mathrm{high}}, \sum_{t'=1}^{H} X_{t'}^{\mathrm{low}}\right)\\ &= \sum_{t=1}^{H}\sum_{t'=1}^{H}\mathrm{Cov}(X_t^{\mathrm{high}}, X_{t'}^{\mathrm{low}}).\end{aligned}$$

Likewise, $\mathrm{Var}(B^{\mathrm{low}}) = \mathrm{Cov}(B^{\mathrm{low}}, B^{\mathrm{low}}) = \sum_{t=1}^{H}\sum_{t'=1}^{H}\mathrm{Cov}(X_t^{\mathrm{low}}, X_{t'}^{\mathrm{low}})$.

The per-step fit pools the KH pairs and computes moments centered at the means $\bar{X}^{\mathrm{high}} := \frac{1}{KH}\sum_{k,t} X_{k,t}^{\mathrm{high}}$ and $\bar{X}^{\mathrm{low}}$:

$$\widehat{\mathrm{Cov}}_{\mathrm{step}} := \frac{1}{KH}\sum_{k,t}\big(X_{k,t}^{\mathrm{high}} - \bar{X}^{\mathrm{high}}\big)\big(X_{k,t}^{\mathrm{low}} - \bar{X}^{\mathrm{low}}\big), \qquad \widehat{\mathrm{Var}}_{\mathrm{step}} := \frac{1}{KH}\sum_{k,t}\big(X_{k,t}^{\mathrm{low}} - \bar{X}^{\mathrm{low}}\big)^2.$$

The per-step fit is the ratio $\hat{c}_{\mathrm{step}} := -\widehat{\mathrm{Cov}}_{\mathrm{step}}/\widehat{\mathrm{Var}}_{\mathrm{step}}$. By the strong law of large numbers over the independent episodes, as $K \to \infty$, almost surely

$$\widehat{\mathrm{Cov}}_{\mathrm{step}} \;\longrightarrow\; \frac{1}{H}\sum_{t=1}^{H}\Big[\,\mathrm{Cov}\left(X_t^{\mathrm{high}}, X_t^{\mathrm{low}}\right) + \big(\mu_t^{\mathrm{high}} - \bar{\mu}^{\mathrm{high}}\big)\big(\mu_t^{\mathrm{low}} - \bar{\mu}^{\mathrm{low}}\big)\Big],$$

where $\mu_t := \mathbb{E}[X_t]$ and $\bar{\mu} := \frac{1}{H}\sum_t \mu_t$, and analogously for $\widehat{\mathrm{Var}}_{\mathrm{step}}$. Under the assumption that the per-step means are constant in t

$$-\frac{\widehat{\mathrm{Cov}}_{\mathrm{step}}}{\widehat{\mathrm{Var}}_{\mathrm{step}}} \;\longrightarrow\; -\frac{\sum_t \mathrm{Cov}(X_t^{\mathrm{high}}, X_t^{\mathrm{low}})}{\sum_t \mathrm{Var}(X_t^{\mathrm{low}})} = c_{\mathrm{step}}.$$

Splitting each double sum above into its same-step terms with $t = t'$ and its cross-step terms with $t \neq t'$, and using $\mathrm{Cov}(X_t^{\mathrm{low}}, X_t^{\mathrm{low}}) = \mathrm{Var}(X_t^{\mathrm{low}})$, gives

$$\mathrm{Cov}(B^{\mathrm{high}}, B^{\mathrm{low}}) = \sum_{t}\mathrm{Cov}(X_t^{\mathrm{high}}, X_t^{\mathrm{low}}) + \sum_{t\neq t'}\mathrm{Cov}(X_t^{\mathrm{high}}, X_{t'}^{\mathrm{low}}),$$

$$\mathrm{Var}(B^{\mathrm{low}}) = \sum_t \mathrm{Var}(X_t^{\mathrm{low}}) + \sum_{t \neq t'} \mathrm{Cov}(X_t^{\mathrm{low}}, X_{t'}^{\mathrm{low}}).$$

If all cross-step covariances vanish, both off-diagonal sums are zero and $c^* = c_{\mathrm{step}}$. They do not coincide in general: with $H = 2$, zero-mean statistics, and $X_1^{\mathrm{low}} := X_2^{\mathrm{high}}$, $X_2^{\mathrm{low}} := X_1^{\mathrm{high}}$ for independent $X_1^{\mathrm{high}}, X_2^{\mathrm{high}}$ with positive variance, every same-step covariance vanishes, so $c_{\mathrm{step}} = 0$, while $B^{\mathrm{low}} = B^{\mathrm{high}}$ gives $c^* = -1$. □

On the constant-means assumption. Without the assumption that the per-step means $\mathbb{E}[X_t^{\mathrm{high}}]$ and $\mathbb{E}[X_t^{\mathrm{low}}]$ are constant in t, the limits retain the terms $\frac{1}{H}\sum_t(\mu_t^{\mathrm{high}} - \bar{\mu}^{\mathrm{high}})(\mu_t^{\mathrm{low}} - \bar{\mu}^{\mathrm{low}})$ in the numerator and $\frac{1}{H}\sum_t(\mu_t^{\mathrm{low}} - \bar{\mu}^{\mathrm{low}})^2$ in the denominator. The per-step fit then mixes the dependence structure with the variation of the mean over the horizon. The assumption isolates the former, which is the effect of interest.

A.2 Why Are the Uncoupled LF Rollouts Not Re-synchronized?

The coupled LF rollouts are re-synchronized to recorded HF states to preserve correlation (Section 4.1), whereas the uncoupled LF rollouts (which supply the LF baseline $\hat{\mu}^{\mathrm{low}}$ in Eq. (1)) evolve freely. Re-synchronizing the uncoupled LF rollouts to the same states would remove the bias noted in Section 4.1, since the coupled and uncoupled LF rollouts would then follow the same distribution. However, we show that a re-synchronized LF baseline leaves a variance component that no fixed coefficient c can reduce, no matter how faithful the LF simulator is.

Fix the policy and let C collect the recorded HF states and the matched task variables for re-synchronization, and let $(X^{\mathrm{high}}, X^{\mathrm{low}})$ be the paired HF and coupled LF statistics, with finite positive variances and correlation ρ. By the law of total variance, $\mathrm{Var}(X^{\mathrm{high}}) = \mathrm{Var}(\mathbb{E}[X^{\mathrm{high}} \mid C]) + \mathbb{E}[\mathrm{Var}(X^{\mathrm{high}} \mid C)]$. We call the two terms the *between-state* and the *within-state* variance, with respect to the re-synchronization states in C. With freely evolving uncoupled LF rollouts, the LF baseline is a constant $\mu_{\mathrm{u}}^{\mathrm{low}}$, the mean of the uncoupled LF statistic. If the uncoupled LF rollouts are instead re-synchronized to C, the LF baseline becomes the conditional mean $\mu_C^{\mathrm{low}} := \mathbb{E}[X^{\mathrm{low}} \mid C]$. For a coefficient c, the control-variate estimators are

$$Z_{\mathrm{u}} := X^{\mathrm{high}} + c\,\big(X^{\mathrm{low}} - \mu_{\mathrm{u}}^{\mathrm{low}}\big), \qquad Z_C := X^{\mathrm{high}} + c\,\big(X^{\mathrm{low}} - \mu_C^{\mathrm{low}}\big).$$

We treat both LF baselines as exact. Finitely many uncoupled LF rollouts add estimation noise to either.

Lemma 2 (A re-synchronized LF baseline cannot reduce the between-state variance)**.** *For every c,* $\mathbb{E}[Z_C] = \mathbb{E}[X^{\mathrm{high}}]$ *and*

$$\mathrm{Var}(Z_C) = \mathrm{Var}\big(\mathbb{E}[X^{\mathrm{high}} \mid C]\big) + \mathbb{E}\big[\mathrm{Var}(X^{\mathrm{high}} + c\,X^{\mathrm{low}} \mid C)\big] \;\geq\; \mathrm{Var}\big(\mathbb{E}[X^{\mathrm{high}} \mid C]\big),$$

whereas $\mathbb{E}[Z_{\mathrm{u}}] = \mathbb{E}[X^{\mathrm{high}}] + c\,\big(\mathbb{E}[X^{\mathrm{low}}] - \mu_{\mathrm{u}}^{\mathrm{low}}\big)$ *and* $\min_c \mathrm{Var}(Z_{\mathrm{u}}) = (1-\rho^2)\mathrm{Var}(X^{\mathrm{high}})$.

Proof. Since μ_C^{low} is a function of C, $\mathbb{E}[Z_C \mid C] = \mathbb{E}[X^{\mathrm{high}} \mid C]$ and $\mathrm{Var}(Z_C \mid C) = \mathrm{Var}(X^{\mathrm{high}} + c\,X^{\mathrm{low}} \mid C)$. The laws of total expectation and variance give the claims for Z_C. The LF baseline of Z_{u} is a constant, so $\mathrm{Var}(Z_{\mathrm{u}}) = \mathrm{Var}(X^{\mathrm{high}} + c\,X^{\mathrm{low}})$, a quadratic in c minimized at $c^* = -\mathrm{Cov}(X^{\mathrm{high}}, X^{\mathrm{low}})/\mathrm{Var}(X^{\mathrm{low}})$ with the stated value. □

The between-state variance reflects which states the scarce HF rollouts start from. A re-synchronized LF baseline removes the between-state part of X^{low} from the LF corrector, so the LF correction acts only on the within-state variance and leaves $\mathbb{E}[X^{\mathrm{high}} \mid C]$ unchanged, whereas the free LF baseline keeps both parts, so that one coefficient can reduce both. The idealized case of perfect LF statistics makes this concrete. If $X^{\mathrm{low}} = X^{\mathrm{high}}$ and $\mu_{\mathrm{u}}^{\mathrm{low}} = \mathbb{E}[X^{\mathrm{high}}]$, then $c = -1$ gives $Z_{\mathrm{u}} = \mathbb{E}[X^{\mathrm{high}}]$ with zero variance, whereas $Z_C = \mathbb{E}[X^{\mathrm{high}} \mid C]$ retains the full between-state variance.

In our design, we leave the uncoupled LF rollouts free to keep both variance components within reach of the control-variate estimator, at the price of a possible bias that vanishes when the coupled and uncoupled LF statistics have the same mean. Reducing both variance components without bias would require an LF baseline estimated from additional re-synchronization states. Such states come only from the HF environment, precisely the data that are scarce.

A.3 BUDGET ALLOCATION

Sampling model and estimator. Fix the policy θ and consider an idealized sampling model. Each sample is a T-step rollout, represented by its mean surrogate $X_i^f := T^{-1}\sum_{t=1}^{T} X_{i,t}^f$. We draw m^{high} i.i.d. coupled pairs $(X_i^{\text{high}}, X_i^{\text{low}})$ with Pearson correlation ρ, $\rho^2 < 1$, and finite positive variances, together with $m^{\text{low}} - m^{\text{high}}$ additional independent LF samples drawn from the same marginal distribution as the coupled LF samples. Let $\hat{\mu}_{\text{all}}^{\text{low}} := \frac{1}{m^{\text{low}}}\sum_{j=1}^{m^{\text{low}}} X_j^{\text{low}}$ be the baseline mean over all LF samples. The estimator in Eq. (2) is

$$L_{\text{all}}^{\text{MF}}(\theta) := \frac{1}{m^{\text{high}}}\sum_{i=1}^{m^{\text{high}}}\left[X_i^{\text{high}} + c\left(X_i^{\text{low}} - \hat{\mu}_{\text{all}}^{\text{low}}\right)\right]. \tag{3}$$

Variance and coefficient. For fixed c, accounting for the shared samples in the coupled LF mean and the baseline gives

$$\text{Var}\left[L_{\text{all}}^{\text{MF}}\right] = \frac{\text{Var}(X^{\text{high}})}{m^{\text{high}}} + \left(\frac{1}{m^{\text{high}}} - \frac{1}{m^{\text{low}}}\right)\left[2c\,\text{Cov}(X^{\text{high}}, X^{\text{low}}) + c^2\text{Var}(X^{\text{low}})\right]. \tag{4}$$

For $m^{\text{low}} > m^{\text{high}}$, minimizing this quadratic over c at fixed counts gives

$$c^* = -\frac{\text{Cov}(X^{\text{high}}, X^{\text{low}})}{\text{Var}(X^{\text{low}})}, \tag{5}$$

independent of $(m^{\text{high}}, m^{\text{low}})$. Substituting c^*,

$$\text{Var}\left[L_{\text{all}}^{\text{MF}}\right] = \text{Var}(X^{\text{high}})\left[\frac{1-\rho^2}{m^{\text{high}}} + \frac{\rho^2}{m^{\text{low}}}\right]. \tag{6}$$

Constrained allocation. For positive costs and budget, set $\nu = m^{\text{low}}/m^{\text{high}} \geq 1$ and substitute $m^{\text{high}} = p/(w^{\text{high}} + w^{\text{low}}\nu)$ into Eq. (6) to obtain

$$\begin{aligned} F(\nu) &:= \frac{\text{Var}(X^{\text{high}})}{p}(w^{\text{high}} + w^{\text{low}}\nu)\left(1 - \rho^2 + \frac{\rho^2}{\nu}\right), \\ F'(\nu) &= \frac{\text{Var}(X^{\text{high}})}{p}\left[w^{\text{low}}(1-\rho^2) - \frac{w^{\text{high}}\rho^2}{\nu^2}\right]. \end{aligned} \tag{7}$$

Since F' is nondecreasing, the minimizer on $\nu \geq 1$ is $\nu^* = \max\{1, \tilde{\nu}\}$, where $\tilde{\nu} := \sqrt{w^{\text{high}}\rho^2/(w^{\text{low}}(1-\rho^2))}$, recovering the closed-form solution of Section 4.2. For $\nu^* > 1$, this agrees with Peherstorfer et al. (2016, Thm. 3.4). At $\nu^* = 1$, $m^{\text{high}} = m^{\text{low}} = p/(w^{\text{high}}+w^{\text{low}})$, and the LF correction vanishes for any c. Optimality is restricted to layouts retaining the paired statistics ($\nu \geq 1$), which keeps ρ observable even when the control variate is disabled. Having $\nu^* > 1$ alone does not guarantee lower variance than an equal-cost HF-only allocation (Peherstorfer et al., 2016, Corollary 3.5).

Integer counts and implementation. Equation (2) is solved over continuous counts and the solution is floored to integers. Flooring preserves ordering and the budget cap but may leave budget unused or give $m^{\text{high}} = 0$. We clamp $m^{\text{high}} \geq 1$ and repair the counts to ensure $m^{\text{low}} \geq m^{\text{high}} \geq 1$ and $w^{\text{high}}m^{\text{high}} + w^{\text{low}}m^{\text{low}} \leq p$. Infeasible budgets are rejected. While $m^{\text{low}} < m^{\text{high}}$, we increment m^{low} if the budget permits, otherwise decrement m^{high}. While over budget, we decrement m^{low} if $m^{\text{low}} > m^{\text{high}}$, otherwise m^{high}, always retaining $m^{\text{high}} \geq 1$. Unused budget is not redistributed. The resulting counts map onto the inputs of Algorithm 1 as $N^{\text{high}} = m^{\text{high}}$ and $N^{\text{low}} = m^{\text{low}} - m^{\text{high}}$, with the coupled LF samples included in m^{low}.

Scope of the result. In practice, coupled LF rollouts start from HF states while uncoupled rollouts evolve freely, so their statistics need not share a marginal distribution. We also use per-transition correlation estimates, which omit cross-step covariance relevant to rollout-level allocation (Lemma 1). The moments are estimated and smoothed online as the policy changes. The optimality statement therefore applies to the idealized scalar-estimator variance above, not to the deployed algorithm.

A.4 COMPLETE ALGORITHM

Algorithm 1 summarizes MFPG-PPO, with each iteration following the four steps of Section 4.1. Lines 3 to 6 collect the multi-fidelity data, lines 7 and 8 construct the paired statistics, lines 9 and 11 fit the coefficient c, and lines 9 and 12 monitor the control variate. The policy is reparameterized as in Section 4.1. For a Gaussian policy with mean action $\bar{a}_\theta(s)$ and standard deviation σ_θ, $g_\theta(s, \varepsilon) = \bar{a}_\theta(s) + \sigma_\theta \odot \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I)$. Appendix A.5 specifies the monitoring procedures, and Appendix A.6 the implementation details. For each minibatch $\mathcal{B}$, $\mathcal{B}^{\text{high}}$ indexes all its HF transitions, including those without a valid coupled LF counterpart. $L^{\text{MF}}(\theta)\big|_{\mathcal{B}}$ denotes Eq. (1) evaluated on that minibatch, with the HF term averaged only over transitions with valid coupled LF counterparts.

Algorithm 1 MFPG-PPO

Require: Rollout counts $N^{\text{high}}, N^{\text{low}}$; rollout length T; PPO hyperparameters; iteration count $\ell_{\max}$

1: Initialize policy π_θ, critics $V^{\text{high}}, V_{\text{c}}^{\text{low}}, V_{\text{u}}^{\text{low}}$, and the monitoring state, cf. Appendix A.5
2: **for** iteration $\in \{1, 2, \ldots, \ell_{\max}\}$ **do**
3: **Collect** N^{high} HF rollouts with actions $a_t^{\text{high}} = g_\theta(s_t^{\text{high}}, \varepsilon_t)$, recording the start states, task variables, and noise ε_t
4: **Reset** each coupled LF environment to its counterpart's recorded start state and task variables, then roll out T steps with actions $a_t^{\text{low}} = g_\theta(s_t^{\text{low}}, \varepsilon_t)$ using the recorded noise ε_t, cf. Fig. 2
5: **Collect** N^{low} uncoupled T-step LF rollouts with independently drawn noise and free resets
6: **Form** the index sets $\mathcal{P}$ of coupled pairs and $\mathcal{U}$ of uncoupled LF transitions
7: **Estimate advantages** for each batch with the corresponding critic over the GAE horizons of Fig. 2
8: **Evaluate** the statistics $X_{k,t}^{\text{high}}(\theta), X_{k,t}^{\text{low}}(\theta), \hat{\mu}^{\text{low}}(\theta)$ of Eq. (1) for all three batches
9: **Fit and smooth** c on the pairs in $\mathcal{P}$, cf. Section 4.1; run the iteration-level check, cf. Appendix A.5
10: **for** each epoch and minibatch $\mathcal{B}$ **do**
11: **Re-evaluate** the statistics $X_{k,t}^{\text{high}}(\theta), X_{k,t}^{\text{low}}(\theta), \hat{\mu}^{\text{low}}(\theta)$ of Eq. (1) on $\mathcal{B}$ under the current policy; re-fit c on the pairs in $\mathcal{B}$
12: **Monitor** with the minibatch checks of Appendix A.5; smooth c if the evidence check passes, apply the control variate only if all checks pass, and set $c \leftarrow 0$ otherwise
13: **Set** the policy objective $L_{\mathcal{B}}(\theta) \leftarrow \begin{cases} L^{\text{MF}}(\theta)\big|_{\mathcal{B}}, & c \neq 0, \\ \dfrac{1}{|\mathcal{B}^{\text{high}}|} \displaystyle\sum_{(k,t)\in\mathcal{B}^{\text{high}}} X_{k,t}^{\text{high}}(\theta), & c = 0. \end{cases}$
14: **Update** with a PPO step on $L_{\mathcal{B}}(\theta)$; fit each critic on its corresponding transitions with targets that never span a re-synchronized state
15: **end for**
16: **end for**

Budget-allocation variant. Before each iteration's data collection, we set N^{high} and N^{low} using the allocation and integer-rounding procedure in Appendix A.3. The objective $L_{\text{all}}^{\text{MF}}$ replaces the baseline mean in L^{MF} with the mean over all LF transitions in each minibatch.

A.5 MONITORING THE CONTROL VARIATE

Throughout, $\hat{c}$ denotes the coefficient fitted on the current fitting data, $\bar{c}$ the running smoothed coefficient, and c the coefficient applied in Eq. (1): $c = \bar{c}$ when all applicable checks pass and $c = 0$ otherwise.

Simulation. We fit and monitor the control variate at both batch and minibatch levels. At iteration start, we fit $\hat{c} = -\widehat{\text{Cov}}(X^{\text{high}}, X^{\text{low}})/\widehat{\text{Var}}(X^{\text{low}})$ on all paired transitions in the batch, update $\bar{c} \leftarrow (1-\alpha)\bar{c} + \alpha\hat{c}$, and check variance reduction. If the check fails, we disable the control variate for the iteration. For each minibatch of an eligible iteration, we re-fit $\hat{c}$ from its paired transitions and obtain a bootstrap confidence interval (Efron, 1992) by resampling those pairs. If the interval contains zero, we disable the control variate for that minibatch. Otherwise, we update $\bar{c}$ and apply the control variate using $\bar{c}$ only if the variance-reduction check passes.

Real robot. With only 4 paired episodes per iteration, within-iteration bootstrap intervals can be unstable. Thus, we assess statistical evidence once per iteration, using a Fisher-z heuristic to accumulate correlation evidence across iterations. The transform approximately normalizes the sampling distribution of correlations and stabilizes its variance. At iteration start, we average the Fisher-z (Fisher, 1921) transforms of the per-iteration correlations: $\bar{z} = |\mathcal{W}|^{-1} \sum_{m \in \mathcal{W}} \operatorname{atanh}\big(\widehat{\mathrm{Corr}}_k(B^{\mathrm{high}}_{m,k}, B^{\mathrm{low}}_{m,k})\big)$, where m indexes iterations, k indexes episodes, and $\mathcal{W}$ contains up to M most recent iterations. We form the approximate $(1-\delta)$ confidence interval $\bar{z} \pm z_{1-\delta/2}/\sqrt{|\mathcal{W}|}$, where $z_{1-\delta/2}$ is the standard-normal quantile. If the interval contains zero, we disable the control variate for the iteration. We update the control-variate coefficient at iteration start and, if the interval excludes zero, before each minibatch update: $\bar{c} = -\operatorname{EMA}_\alpha[\widehat{\mathrm{Cov}}_k(B^{\mathrm{high}}_k, B^{\mathrm{low}}_k)] / \operatorname{EMA}_\alpha[\widehat{\mathrm{Var}}_k(B^{\mathrm{low}}_k)]$ under the current policy on *all* paired episode sums from this iteration. Then, each minibatch applies the correction only if the variance check passes.

Variance check and fallback. We compare the sample variance of $X^{\mathrm{high}} + \bar{c}(X^{\mathrm{low}} - \hat{\mu}^{\mathrm{low}}_{\mathcal{B}})$ with that of X^{high}, using individual paired steps in simulation and sums over paired episodes on the robot. A variance ratio greater than or equal to 1 sets $c = 0$, yielding an HF-only actor objective.

Table 1: Control variate monitoring hyperparameters.

Hyperparameter	Simulation	Real robot
EMA rate α	0.002 (FR); 0.025 (BA)	0.05
Bootstrap resamples	10^3	—
Evidence-window length M	—	4
Confidence level $1-\delta$	0.9	0.9

A.6 Implementation and experiment details

A.6.1 Hyperparameter tuning protocol

Simulation tuning protocol. All methods share the same tuning protocol. We tune on H1 Rough and ANYmal-D Rough with $N^{\mathrm{high}} = 256$ and 3 seeds per task. Selected hyperparameters remain fixed across all locomotion tasks and HF data budgets. MFPG-PPO-FR adds $N^{\mathrm{low}} = 4096$ uncoupled LF environments, and the other LF-using baselines match its total LF count.

We select hyperparameters using both final and cumulative HF training performance, measured by final return and area under curve (AUC) (Osband et al., 2016; Hessel et al., 2018). Within each method, let $M_{d,k}(q)$ denote metric $k \in \{\mathrm{final}, \mathrm{AUC}\}$, averaged over seeds, for configuration q on task d. We standardize each metric across candidate hyperparameter configurations as $z_{d,k}(q) = [M_{d,k}(q) - \mu_{d,k}]/s_{d,k}$, where $\mu_{d,k}$ and $s_{d,k}$ are the mean and sample standard deviation across candidates. We select the configuration maximizing $S(q) = \sum_d \sum_k z_{d,k}(q)$.

We tune the PPO backbone independently for PPO HF-Only, MFPG-PPO-FR, Co-training, DARC, and PAR, then tune method-specific hyperparameters with the selected backbone fixed. For each PPO backbone we tune, we first search learning rate/minibatch pairs at the default of 5 epochs, then search the number of epochs for the 4 best pairs. The PPO grids are centered on the values of Isaac Lab's task-default RSL-RL configurations (Mittal et al., 2025; Schwarke et al., 2025). MFPG-PPO-BA inherits the selected MFPG-PPO-FR backbone and tunes only its EMA rate. For MFPG-REINFORCE, we first search the learning rate on a grid centered on the default of the released implementation of Liu et al. (2026), at its default discount factor of 0.97 and EMA rate of 0.05, then jointly search both for the selected learning rate. For SAC HF-Only, we first search pairs of learning rate and update-to-data (UTD) ratio at the default target-entropy scale of 0.167, then search this scale for the 4 best pairs. The number of minibatch updates per iteration sets the UTD ratio, and the target entropy equals the negative scale times the action dimension. Sabatini et al. (2026) use 200 minibatches per iteration with 8192 parallel environments, which implies a UTD ratio of about 8.3. We test UTD ratios of $\{8, 16, 32\}$ and this fixed 200-minibatch setting. Table 2 summarizes the grid values for the hyperparameter search.

Table 2: Grid values for simulation hyperparameter search.

Method	Candidate values
Backbone parameters	
PPO	Learning rate $\{5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-3}\}$; epochs $\{3, 5, 8\}$; minibatches per epoch $\{2, 4, 8\}$
REINFORCE	Learning rate $\{7 \times 10^{-5}, 2 \times 10^{-4}, 7 \times 10^{-4}, 2 \times 10^{-3}, 7 \times 10^{-3}\}$; discount factor $\{0.95, 0.97, 0.99\}$
SAC	Actor and critic learning rate $\{10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}\}$; UTD ratio $\{8, 16, 32\}$ or 200 minibatches per iteration; target-entropy scale $\{0.05, 0.167, 0.5\}$
Method-specific parameters	
MFPG-PPO-FR	EMA rate $\{0.0005, 0.002, 0.01\}$
MFPG-PPO-BA	EMA rate $\{0.025, 0.05, 0.1\}$
DARC / PAR	Reward-correction strength $\{0.005, 0.1, 0.5, 1, 2, 5\}$
MFPG-REINFORCE	EMA rate $\{0.025, 0.05, 0.1\}$

Simulation hyperparameters. Untuned PPO backbone hyperparameters follow Isaac Lab's task-default RSL-RL configurations (Mittal et al., 2025; Schwarke et al., 2025) for all PPO-based methods. PPO LF-Only uses the default backbone because it operates on the default 4096 environments. MFPG-REINFORCE uses the same actor and value network architectures as the PPO-based methods, and its other untuned hyperparameters, e.g., a value-loss coefficient and maximum gradient norm of 1, follow the released implementation of Liu et al. (2026). Untuned SAC hyperparameters follow the task-specific released configurations of Sabatini et al. (2026). Table 3 lists the adopted hyperparameters.

Real-robot hyperparameters. In real-robot experiments, we first tune the shared PPO backbone (adopted from ManiSkill's (Tao et al., 2024) PickCube PPO configuration) over learning rates $\{1 \times 10^{-5}, 4 \times 10^{-5}, 8 \times 10^{-5}, 1.5 \times 10^{-4}, 3 \times 10^{-4}\}$ and minibatch counts per epoch $\{4, 6, 8, 32\}$. We use the selected backbone for both MFPG-PPO and PPO HF-Only. On top of it, we tune the MFPG-specific EMA rate $\{0.05, 0.1\}$ and LF rollout segment length $\{10, 20, 60\}$ steps. We additionally adopt a learning-rate schedule for both methods. The learning rate steps down from 4×10^{-5} to 2.5×10^{-5} and then to 1.5×10^{-5} when the mean training success rate over the latest 10 updates reaches 0.8 and 0.925. Table 4 lists the adopted hyperparameters.

A.6.2 Implementation details

Two-process implementation. To support different simulator fidelities or a physical robot as the HF environment, we implement MFPG-PPO with two processes. An *HF worker* collects HF rollouts, and a *learner* collects coupled and uncoupled LF rollouts and optimizes the policy and critics. The learner sends the policy parameters and action noise to the worker, then uses the returned HF states to initialize coupled LF rollouts before collecting LF data. For locomotion, both processes use Isaac Lab (Mittal et al., 2025), and for Franka robot learning, the worker controls the robot and the learner runs ManiSkill (Tao et al., 2024). In locomotion, coupled LF environments match HF velocity-tracking commands, and coupled and uncoupled LF environments inherit HF terrain assignments. In manipulation, coupled LF environments match the measured HF object pose.

Co-training. We adapt Co-training method (Wei et al., 2025) to on-policy RL setting and train a shared policy and critic on independently collected HF and LF rollouts. Let $\mathcal{L}^f_{\mathrm{PPO}}$ be the mean loss in fidelity f, including policy, value, and entropy terms, and n^f its transition count. Co-training minimizes

$$\mathcal{L}_{\mathrm{co}} = \frac{n^{\mathrm{high}}\mathcal{L}^{\mathrm{high}}_{\mathrm{PPO}} + n^{\mathrm{low}}\mathcal{L}^{\mathrm{low}}_{\mathrm{PPO}}}{n^{\mathrm{high}} + n^{\mathrm{low}}},$$

giving each transition equal weight.

Table 3: Simulation hyperparameters.

PPO backbone hyperparameters inherited from Isaac Lab's task-default RSL-RL configurations

Parameter	Value
Rollout length per environment	24 steps
Optimizer	Adam
Discount factor / GAE parameter	0.99 / 0.95
PPO clipping parameter	0.2
Value-loss coefficient / value clipping	1 / enabled
Entropy coefficient	H1: 0.01; ANYmal-D: 0.005
Learning-rate schedule	KL-adaptive; target KL divergence 0.01
Maximum gradient norm	1
Actor and critic hidden-layer sizes	Rough: $(512, 256, 128)$; flat: $(128, 128, 128)$
Activation function	ELU
Observation normalization	Disabled
Initial policy standard deviation	1

SAC hyperparameters inherited from the released configurations of Sabatini et al. (2026)

Parameter	Value
Rollout length per environment	24 steps
Optimizer	Adam
Discount factor	Rough: 0.97; flat: 0.96
Multi-step return horizon	5 steps
Replay buffer size	5×10^6 transitions
Minibatch size	8192 transitions
Temperature	Initial 0.001; automatically tuned with learning rate 2×10^{-5}
Target smoothing coefficient	0.003
Maximum gradient norm	1
Actor and critic hidden-layer sizes	Rough: $(1024, 512, 256)$; flat: $(512, 256, 128)$
Activation function	SiLU
Observation normalization	Enabled
Initial policy standard deviation	0.15

Method configurations

Method	Learning rate	Epochs	Minibatches per epoch	Additional hyperparameters
PPO HF-Only	2×10^{-3}	5	8	—
MFPG-PPO-FR	5×10^{-4}	8	8	EMA rate 0.002
MFPG-PPO-BA	5×10^{-4}	8	8	EMA rate 0.025
Co-training	5×10^{-4}	8	8	—
DARC	2×10^{-3}	3	8	Reward-correction strength 0.005
PAR	2×10^{-3}	8	8	Reward-correction strength 0.005; equal HF/LF loss weights
MFPG-REINFORCE	7×10^{-5}	—	—	Discount factor 0.97; EMA rate 0.025
PPO LF-Only	10^{-3}	5	4	—
SAC HF-Only	5×10^{-4}	1	$\frac{32 \times 24 N^{\text{high}}}{8192}$	UTD ratio 32; target-entropy scale 0.5

DARC. Domain Adaptation with Rewards from Classifiers (DARC) (Eysenbach et al., 2021) corrects LF rewards using a classifier-based estimate of the dynamics mismatch. We adapt this correction to PPO. Two binary domain classifiers predict the fidelity label f: $D_{sa}(f \mid s, a)$ from state-action pairs and $D_{sas}(f \mid s, a, s')$ from transitions. Together, they estimate the relative likelihood

Table 4: Real-robot hyperparameters.

Parameter	Value
Shared	
Initial learning rate	4×10^{-5}
Epochs	4
Minibatches per epoch	4
HF episodes per update	4
Optimizer	Adam
Discount factor / GAE parameter	0.8 / 0.95
PPO clipping parameter	0.2
Value-loss coefficient / value clipping	0.5 / disabled
Entropy coefficient	0
Maximum gradient norm	0.5
Actor and critic hidden-layer sizes	$(256, 256)$
Activation function	ELU
Observation normalization	Disabled
MFPG-PPO-specific	
EMA rate	0.05
Coupled LF rollout-segment length	20 steps
Uncoupled LF environments	8192

of observing s' under HF versus LF dynamics, given (s, a). The resulting reward correction is

$$\widetilde{r}^{\text{low}} = r^{\text{low}} + \eta_{\text{DARC}} \left[\log \frac{D_{sas}(\text{high} \mid s, a, s')}{D_{sas}(\text{low} \mid s, a, s')} - \log \frac{D_{sa}(\text{high} \mid s, a)}{D_{sa}(\text{low} \mid s, a)} \right],$$

where η_{DARC} controls the correction strength. We follow Eysenbach et al. (2021) and the authors' released code[1], using the classifier architecture and batch size specified in Eysenbach et al. (2021, Appendix E). The policy and critic train only on LF rollouts with corrected rewards. Both fidelities train the classifiers. We use two networks with hidden layers $(256, 256)$ and ReLU activations, add the state-action network's logits to the transition network's logits, and apply Gaussian input noise with standard deviation 1 only during classifier training. At each classifier optimization step, we sample 128 transitions from each fidelity's replay buffer, forming minibatches $\mathcal{B}^{\text{high}}$ and $\mathcal{B}^{\text{low}}$, and minimize

$$\mathcal{L}_{\text{cls}} = -\frac{1}{256} \sum_{f \in \{\text{high,low}\}} \sum_{(s,a,s') \in \mathcal{B}^f} \left[\log D_{sa}(f \mid s, a) + \log D_{sas}(f \mid s, a, s') \right].$$

Classifier training starts in the first iteration. In our implementation, reward correction begins after the first 10% of training iterations.

PAR. Policy Adaptation by Representation Mismatch (PAR) (Lyu et al., 2024) penalizes LF transitions using the mismatch between the embedding of the observed next state and the embedding predicted by a representation model trained on HF data. We follow Lyu et al. (2024) and the authors' released code[2], adapting the policy and critic updates to PPO. Let $f_{\bar{\psi}}$ and $g_{\bar{\psi}}$ be target copies of a state encoder and a next-state embedding predictor, respectively, with parameters $\bar{\psi}$ maintained by an EMA of the online parameters ψ. For embedding dimension d_z and correction strength η_{PAR}, the corrected LF reward is

$$\widetilde{r}^{\text{low}} = r^{\text{low}} - \frac{\eta_{\text{PAR}}}{d_z} \left\| g_{\bar{\psi}}\big(f_{\bar{\psi}}(s), a\big) - f_{\bar{\psi}}(s') \right\|_2^2.$$

We train a shared policy and critic on HF rollouts with their original rewards and LF rollouts with corrected rewards. Let $\mathcal{L}_{\text{PPO}}^{f}$ denote the mean PPO loss over transitions from fidelity f, including policy, value, and entropy terms. Following PAR's equal weighting of the two domains, we minimize

$$\mathcal{L}_{\text{PAR}} = \tfrac{1}{2}\mathcal{L}_{\text{PPO}}^{\text{high}} + \tfrac{1}{2}\mathcal{L}_{\text{PPO}}^{\text{low}}.$$

[1] https://github.com/google-research/google-research/tree/master/darc

[2] https://github.com/dmksjfl/PAR

The encoder and predictor each have hidden layers $(256, 256)$ with ELU activations, and $d_z = 256$. We normalize each state embedding by dividing it by the mean absolute value of its components. On HF transitions (s, a, s'), we train the encoder and predictor to predict the next-state embedding from the current state and action, minimizing $\|g_\psi(f_\psi(s), a) - \mathrm{sg}(f_\psi(s'))\|_2^2/d_z$, where sg treats its argument as fixed during differentiation. In our implementation, after completing all PPO epochs, we perform the same number of gradient updates to the encoder and predictor as to the policy. Each update uses 128 transitions sampled from the HF replay buffer. After each gradient update, we update the target parameters as $\bar{\psi} \leftarrow 0.995\,\bar{\psi} + 0.005\,\psi$. Reward correction is active from the first iteration.

Shared reward correction details. DARC and PAR correct rewards before GAE computation. Auxiliary networks for reward correction remain fixed during policy optimization, followed by as many auxiliary gradient steps as policy update steps. Both DARC and PAR use Adam with learning rate 3×10^{-4} for auxiliary networks and replay buffer size 10^6 per stored fidelity: both fidelities for DARC, only HF for PAR. Table 3 gives the selected correction strengths.

MFPG-REINFORCE. We follow Liu et al. (2026) and the authors' released implementation[3]. Each iteration, MFPG-REINFORCE optimizes the objective with the episode-level statistic $X_\tau(\theta)$ defined in Section 3

$$\frac{1}{N^{\mathrm{high}}} \sum_{k=1}^{N^{\mathrm{high}}} \Big[X_{\tau_k^{\mathrm{high}}}(\theta) + c \left(X_{\tau_k^{\mathrm{low}}}(\theta) - \hat{\mu}_{\mathrm{all}}^{\mathrm{low}}(\theta) \right) \Big].$$

Here, k indexes coupled rollout pairs, and $\hat{\mu}_{\mathrm{all}}^{\mathrm{low}}$ averages the statistics over all coupled and uncoupled LF rollouts. Each iteration, MFPG-REINFORCE collects parallel 100-step multi-fidelity rollouts. Following the released implementation of Liu et al. (2026), it approximates full-episode Monte Carlo returns using rewards up to episode termination or the rollout boundary, without value bootstrapping. A shared value function V^{high}, fitted to returns on paired HF transitions, supplies the baseline for both fidelities. After each collection, MFPG-REINFORCE fits c per step as described in Section 4.1, smoothing the correlation and standard deviations separately with an EMA. It sets $c = 0$ when the current, unsmoothed correlation is negative. For the policy update, it differentiates all three terms, holding the returns, value baselines, and c fixed. Each collection produces one full-batch policy and value update.

SAC HF-Only. We use the SAC implementation and Isaac Lab configurations released by Sabatini et al. (2026)[4]. At the start of HF training, we restore the actor, critics, and observation-normalization statistics from the task's LF-pretrained checkpoint described in Appendix A.6.3, and reinitialize the optimizers and temperature. The replay buffer starts empty and stores only HF transitions. Each iteration collects a 24-step rollout from each of the N^{high} HF environments. The first $8192/N^{\mathrm{high}}$ iterations collect a total of 24×8192 transitions without gradient updates, matching the released configuration's single collection-only iteration with 8192 environments. These iterations count toward the HF sampling budget. Subsequent iterations perform the minibatch updates in Table 3.

Sampling budgets. Table 5 summarizes per-iteration sampling budgets for all the methods. For MFPG-PPO-BA, we assume costs $w^{\mathrm{high}} = 10$ and $w^{\mathrm{low}} = 1$ per rollout and set $p = 10N_{\mathrm{ref}}^{\mathrm{high}} + 4096$, where $N_{\mathrm{ref}}^{\mathrm{high}}$ is the nominal HF environment count of the MFPG-PPO-FR with the corresponding total cost.

A.6.3 Experiment details

Simulation locomotion tasks and fidelities. We use Isaac Lab's Unitree H1 and ANYmal-D velocity-tracking tasks on flat and rough terrain[5], with task-default observations, rewards, termination rules, and action space. Actions specify joint-position targets. Observations comprise base linear and angular velocities, projected gravity, velocity commands, joint positions relative to the default pose and joint velocities, and the previous action. Rough-terrain tasks additionally include a height scan. We disable injected observation noise. Both fidelities act every 0.02 s, with the

[3] `https://github.com/CLeARoboticsLab/MultiFidelityPolicyGradients`

[4] `https://github.com/leggedrobotics/rsl_rl_sac` and `https://github.com/sabagian/isaaclab-sac`

[5] The base task IDs are `Isaac-Velocity-Flat-H1-v0`, `Isaac-Velocity-Rough-H1-v0`, `Isaac-Velocity-Flat-Anymal-D-v0`, and `Isaac-Velocity-Rough-Anymal-D-v0`.

Table 5: Simulation rollouts per iteration. Rollouts contain 24 steps, except for MFPG-REINFORCE (100 steps).

Method	HF	Coupled LF	Uncoupled LF
PPO HF-Only	N^{high}	—	—
PPO LF-Only	—	—	4096
MFPG-PPO-FR	N^{high}	N^{high}	4096
MFPG-PPO-BA	m^{high}	m^{high}	$m^{\text{low}} - m^{\text{high}}$
Co-training, DARC, PAR	N^{high}	—	$4096 + N^{\text{high}}$
MFPG-REINFORCE	N^{high}	N^{high}	4096
SAC HF-Only	N^{high}	—	—

physics settings in Table 6. The LF assets use simplified collision geometry visualized in Fig. 6, and ANYmal-D LF additionally approximates cylinder and cone collisions. The cross-task comparison uses $N^{\text{high}} \in \{64, 128, 256, 512\}$. The H1 Rough scaling study additionally includes $N^{\text{high}} \in \{8, 16, 32\}$.

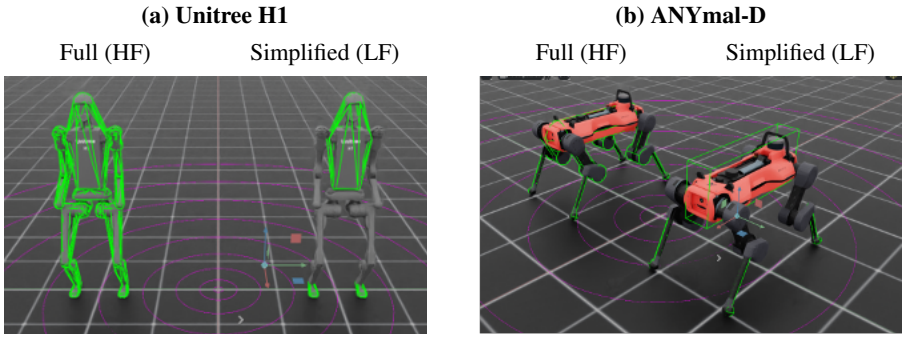


Figure 6: **Locomotion collision models. (a)** Unitree H1 and **(b)** ANYmal-D. Each panel compares the full HF collision model (left) with the simplified LF collision model (right). Green outlines indicate the collision geometry.

Table 6: Locomotion fidelity settings, shared by the flat and rough tasks of each robot.

	H1		ANYmal-D	
Parameter	HF	LF	HF	LF
Physics time step (s)	0.002	0.01	0.002	0.0025
Physics steps per action	10	2	10	8
Position solver iterations	4	2	4	2
Velocity solver iterations	4	0	0	0
Self-collisions	Enabled	Disabled	Enabled	Disabled
Collision geometry	Full	Simplified	Full	Simplified

Initialization and training length. For each task, all on-policy methods that use HF data initialize from the same policy checkpoint, pre-trained with PPO in the LF environment for 500 iterations on H1 Rough and 100 on the other tasks. PPO pretraining uses Isaac Lab's task-default RSL-RL hyperparameters. Since SAC uses different actor and critic network architectures, cf. Table 3, SAC HF-Only instead initializes from its own task-specific checkpoint, pre-trained with SAC in 4096 LF environments using the released configurations of Sabatini et al. (2026). Among checkpoints saved every 25 iterations, we select the earliest whose LF training return reaches that of the corresponding PPO LF-pretrained counterpart. The selected checkpoints are at iterations 150, 125, 200, and 225 on H1 Rough, H1 Flat, ANYmal-D Rough, and ANYmal-D Flat, respectively. We train for 8000 iterations on H1 Rough and 4000 on the other tasks, excluding pretraining. MFPG-REINFORCE uses 100-step rollouts and trains for 1920 and 960 iterations, respectively, matching the total HF transition counts of PPO at the same N^{high}.

Simulation evaluation. We use Isaac Lab's `Play` evaluation tasks[6], replacing their physics settings with the HF configuration in Table 6. We evaluate each checkpoint deterministically over 2048 completed episodes and compute its mean undiscounted episode return. We report means across seeds with 95% bootstrapped confidence intervals. For return differences from PPO HF-Only, we resample the two seed sets independently.

Franka task. Figure 7 shows the physical Franka robot HF environment and its ManiSkill LF simulation counterpart. In the HF environment, an Intel RealSense D435 camera estimates the cube's position and orientation from AprilTags attached to its six faces. The task is to grasp a cube with edge length $3.8\,\text{cm}$ and lift its center to the fixed goal $(x, y, z) = (0.45, 0, 0.2715)\,\text{m}$ in the robot base frame. The tabletop is at $z = 0.0545\,\text{m}$, placing the goal $0.198\,\text{m}$ above the cube center when the cube is on the table. Initial cube positions lie within $x \in [0.35, 0.50]\,\text{m}$ and $y \in [-0.075, 0.075]\,\text{m}$. Simulation samples these coordinates uniformly, with yaw sampled uniformly from $[-\pi/4, \pi/4]$. The physical cube may rest on any face and at any yaw. At reset, we relabel its axes using cube symmetry to express its initial orientation in the same convention. Between physical episodes, a scripted routine repositions the cube within this region if it remains grasped. Otherwise, a human operator randomly repositions it. Each episode comprises $H = 60$ actions at $5\,\text{Hz}$ and continues to the horizon after success.

The task is harder in the HF environment because the side-mounted camera can lose sight of the cube during contact. Across both methods, approximately 69% of training control steps lack a valid cube pose estimate. When perception fails, we retain the last valid pose, updating its position from the end effector if a grasp is detected. Grasp detection relies on gripper width and may be delayed. By contrast, the LF simulator provides ground-truth object and grasp states. Friction and controller stiffness also differ between fidelities.

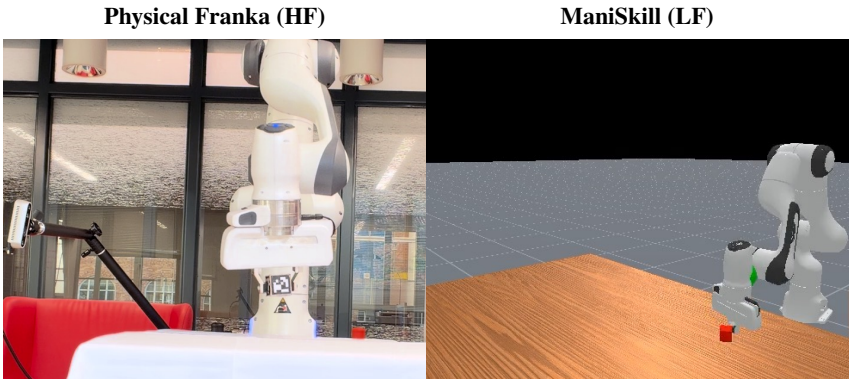


Figure 7: **Franka cube-picking environments.** The physical Franka setup (left) provides the HF environment, and ManiSkill simulation (right) provides the LF environment.

Robot observations and actions. The policy receives the 42-dimensional state observation in Table 7. Positions are expressed in the robot base frame. The five action components command end-effector translations, yaw rotation, and gripper opening or closing. The physical Franka arm uses Cartesian impedance control. The LF simulator uses ManiSkill's default physics settings and its `pd_ee_delta_pose` controller, which converts end-effector pose targets to joint targets through inverse kinematics. Joint PD control tracks these targets, with arm-joint stiffness and damping set to 300 and 200, respectively.

[6]The task IDs are `Isaac-Velocity-Flat-H1-Play-v0`, `Isaac-Velocity-Rough-H1-Play-v0`, `Isaac-Velocity-Flat-Anymal-D-Play-v0`, and `Isaac-Velocity-Rough-Anymal-D-Play-v0`.

Table 7: State observations for the Franka cube-picking task. Here, φ is cube yaw and t counts elapsed control steps.

Observation	Dimensions
Arm joint positions and velocities	$7 + 7$
Gripper width	1
End-effector position and quaternion	$3 + 4$
Cube position and quaternion	$3 + 4$
Goal position	3
End-effector-to-cube and cube-to-goal vectors	$3 + 3$
Grasp indicator	1
Cube-yaw features $\sin(4\varphi), \cos(4\varphi)$	2
Episode phase t/H	1

Robot reward and success. Let d_{reach} and d_{goal} be the distances from the cube center to the end effector and goal, respectively, $b_{\text{grasp}} \in \{0, 1\}$ the grasp indicator, and $b_{\text{place}} = \mathbf{1}\{d_{\text{goal}} \leq 0.05\,\text{m}\}$. With arm joint velocities $\dot{\mathbf{q}}$, both fidelities use

$$r = 1 - \tanh(5d_{\text{reach}}) + b_{\text{grasp}}\big[2 - \tanh(5d_{\text{goal}})\big] + b_{\text{place}}\big[1 - \tanh(5\|\dot{\mathbf{q}}\|_2)\big].$$

Success requires $d_{\text{goal}} \leq 0.05$ m and $\|\dot{\mathbf{q}}\|_\infty \leq 0.5$ rad/s, and replaces the reward with 5. An episode succeeds if this condition holds at least once.

Other training and evaluation details. Both methods start from the same simulation-trained policy and collect 4 physical episodes per update. MFPG-PPO additionally collects $T = 20$-step uncoupled LF simulation rollouts from 8192 independent parallel environments as well as coupled LF rollouts. The uncoupled LF environments are staggered across episode phases (Bharthulwar et al., 2026). We train both methods for 80 iterations per seed, using 3 seeds per method. Each trained policy is evaluated without exploration noise for 20 physical episodes. We report means and ranges across seeds. The simulation-only reference checkpoints are each evaluated for 40 physical episodes.

A.7 Extended Experimental Results

Figure 8 shows the absolute mean HF evaluation returns for the same settings as Fig. 3, which reports return differences relative to PPO HF-Only. White bars denote PPO HF-Only.

In Fig. 1, both MFPG-PPO and PPO HF-Only start from a common policy pretrained briefly for 500 iterations in the LF environment on H1 Rough, cf. Appendix A.6.3. Figure 9 repeats this comparison at the 5 smallest HF budgets after more extensive LF pretraining. For each seed, both methods start from that seed's final PPO LF-Only checkpoint, trained for 8000 iterations, and fine-tune for 8000 additional iterations at matched HF budgets using the hyperparameters in Table 3. MFPG-PPO additionally uses LF data during fine-tuning. The gray PPO LF-Only bar shows the starting checkpoints' mean HF evaluation return. MFPG-PPO achieves higher mean return than PPO HF-Only at all 5 budgets, with performance increasing before plateauing. In contrast, PPO HF-Only varies substantially across budgets and collapses to near-zero mean return at the 2 smallest budgets.

Figure 10 shows how often MFPG-PPO-FR applies the control variate during the Franka training runs in Fig. 4, following the monitoring checks in Appendix A.5.

In Isaac Lab simulation of the locomotion tasks, the timing of key events such as robot foot strikes generally aligns more closely across fidelities, whereas grasp events on the physical Franka robot and in simulation often occur at different steps. On the physical robot, per-step correlation between paired statistics is generally low. Figure 4(c) shows how aggregating statistics over an episode captures cross-step covariance and increases the correlation, supporting the analysis in Lemma 1. Figure 11 reports further event-timing results and cross-fidelity correlations between paired HF and LF statistics.

Figure 12 illustrates how event timing appears in the paired per-step statistics $(X_{k,t}^{\text{high}}, X_{k,t}^{\text{low}})$ for a physical Franka cube-picking episode and an H1 Rough simulation rollout.

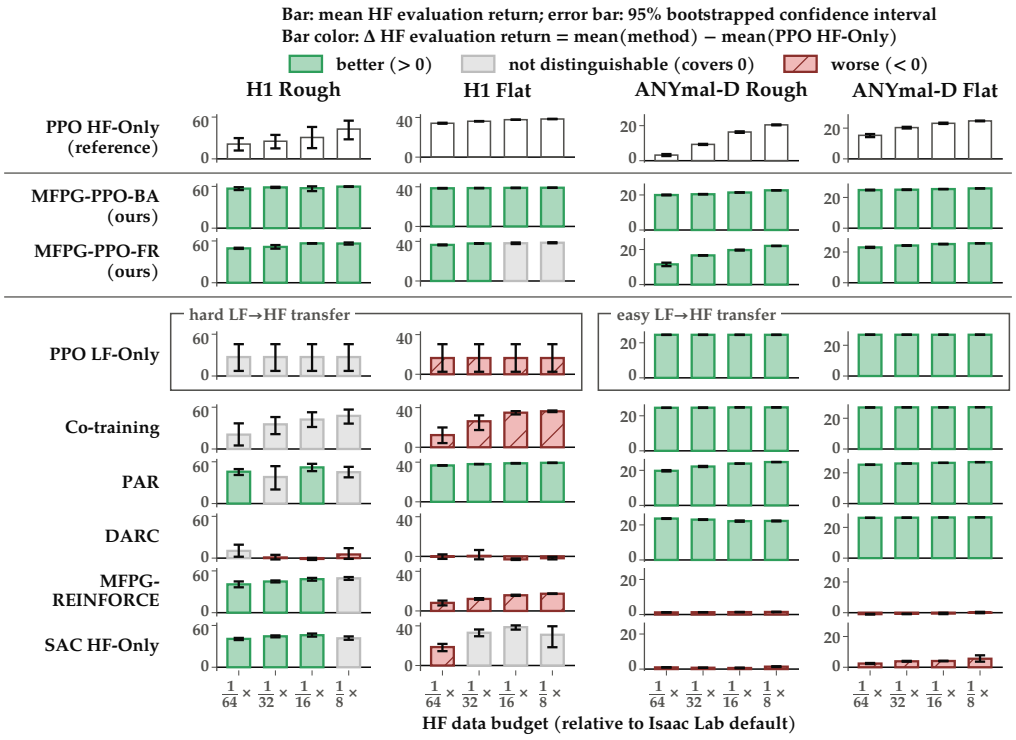


Figure 8: Mean final HF evaluation returns for the task-budget settings in Fig. 3. Error bars denote 95% bootstrap confidence intervals across 6 seeds.

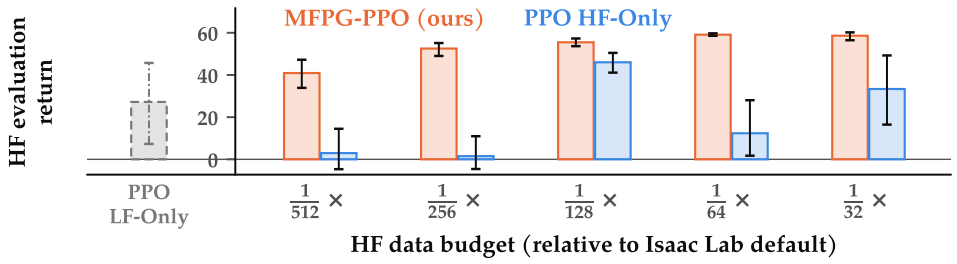


Figure 9: Fine-tuning after extensive LF pretraining on H1 Rough. Both methods start from each seed's final PPO LF-Only checkpoint, whose mean return the gray bar shows. HF budgets are expressed as fractions of 4096 parallel environments. Bars show HF evaluation returns over 2048 episodes per seed, averaged across 6 seeds. Whiskers show 95% seed-bootstrapped confidence intervals for the means.

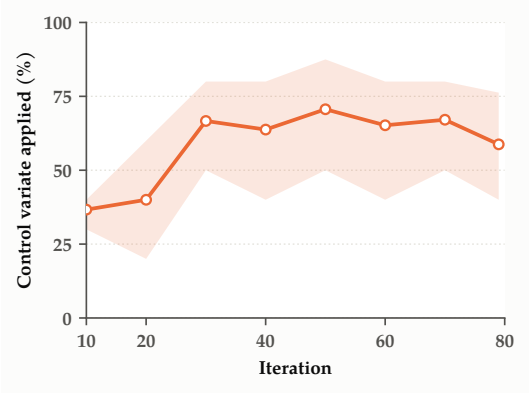


Figure 10: Percentage of minibatches on which the control variate was applied during Franka cube-picking training in Fig. 4. Each marker aggregates the past 10 iterations. The line is the mean over 3 seeds, the band their range.

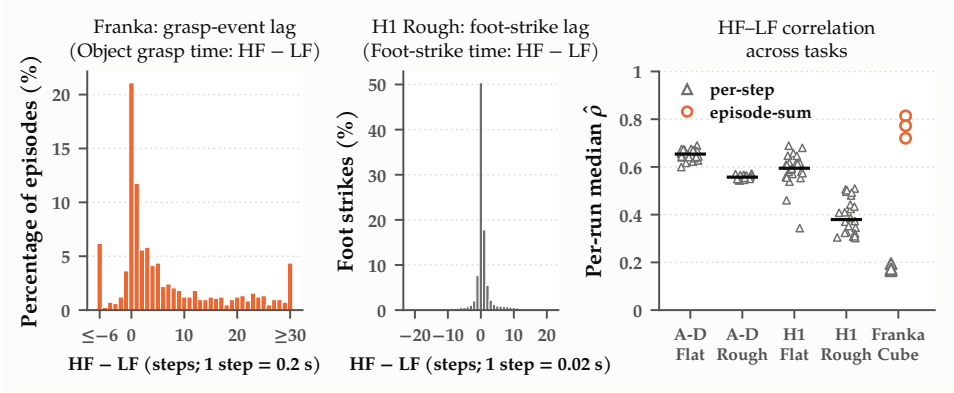


Figure 11: Cross-fidelity event timing and correlation between paired statistics. **Left.** Distribution of grasp-event timing differences between the physical Franka robot and its coupled ManiSkill simulation. The physical robot generally grasps later than its simulated counterpart due to fidelity gaps, causing low correlation between paired per-step statistics. **Middle.** Distribution of foot-strike timing differences between high- and low-fidelity simulations of the H1 Rough task in Isaac Lab. The timing generally aligns more closely than in the Franka case. **Right.** Cross-fidelity correlations between paired HF and LF statistics across the tasks in Sections 5.3 and 5.4. The first four x-axis entries are Isaac Lab simulation tasks and the last is the physical robot task. Each marker gives a run's median correlation over training. Gray triangles denote per-step correlations, orange circles denote episode-sum correlations, and black bars denote medians across runs. A-D denotes ANYmal-D.

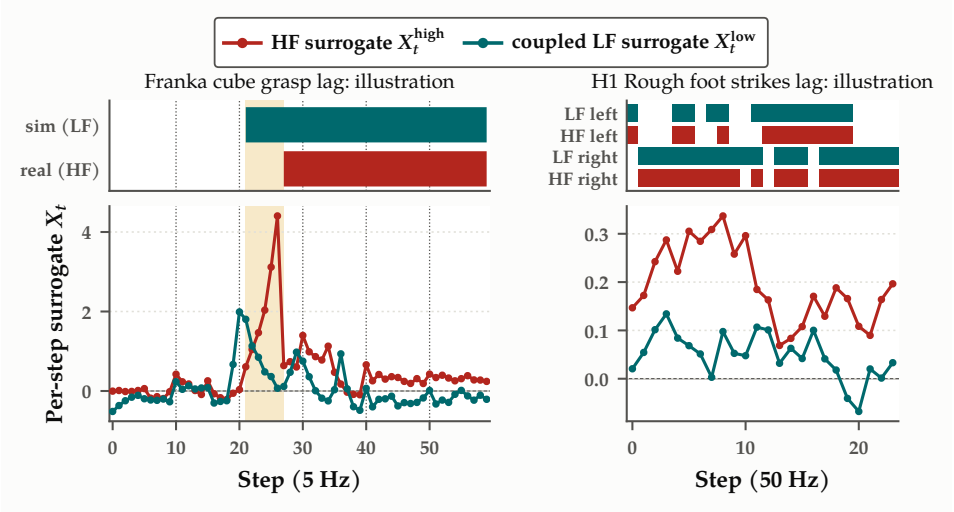


Figure 12: Event indicators (top) and paired per-step surrogate statistics $(X_{k,t}^{\text{high}}, X_{k,t}^{\text{low}})$ (bottom). **Left.** A Franka cube-picking episode. The upper bars indicate detected grasps, and the shaded interval marks the timing mismatch. **Right.** An H1 Rough rollout. The upper bars indicate left- and right-foot contact. In both panels, the lower curves show the HF and coupled LF surrogate statistics in red and teal, respectively.

Figure 13 extends the variance amplification diagnostic of Fig. 5(b) to the $\frac{1}{32}\times$ HF budget on H1 Rough. As in Fig. 5(b), variants without monitoring amplify variance in most iterations.

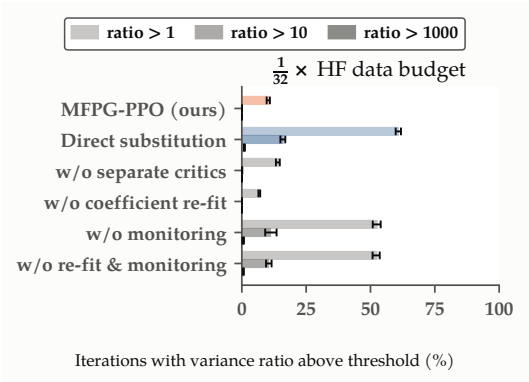


Figure 13: Fraction of training iterations whose variance ratio $\mathrm{Var}[X_{k,t}^{\text{high}} + c\,(X_{k,t}^{\text{low}} - \hat{\mu}^{\text{low}})] \,/\, \mathrm{Var}[X_{k,t}^{\text{high}}]$, averaged over minibatches *before* the monitoring check, exceeds 1, 10, and 1000 for the $\frac{1}{32}\times$ HF budget case of Fig. 5.

For the ablations in Section 5.5, Fig. 14 tracks the variance ratio $\mathrm{Var}[X_{k,t}^{\text{high}} + c\,(X_{k,t}^{\text{low}} - \hat{\mu}^{\text{low}})] \,/\, \mathrm{Var}[X_{k,t}^{\text{high}}]$, the fraction of minibatches failing the monitoring checks, and the estimated correlation between the paired per-step statistics $(X_{k,t}^{\text{high}}, X_{k,t}^{\text{low}})$. The ablation variants exhibit the same level of cross-fidelity correlation. For display, the curves for minibatch rejection and correlation are smoothed with a centered median over an 800-iteration window.

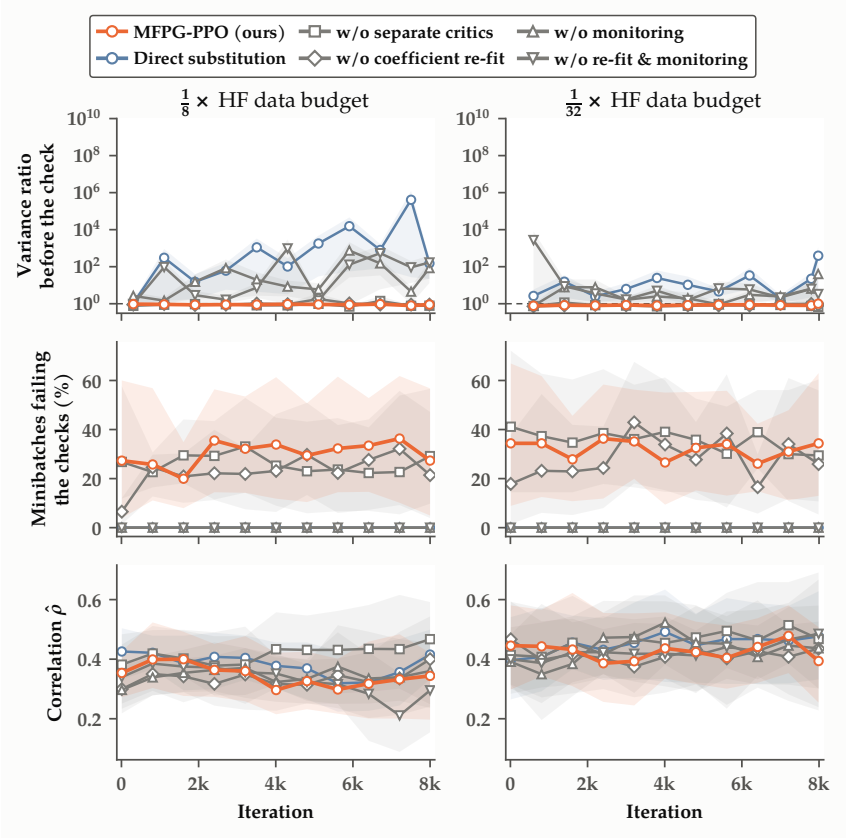


Figure 14: Training diagnostics for the ablations in Section 5.5 at the $\frac{1}{8}\times$ (left) and $\frac{1}{32}\times$ (right) HF budgets on H1 Rough. Rows show the variance ratio $\mathrm{Var}[X^{\mathrm{high}}_{k,t} + c\,(X^{\mathrm{low}}_{k,t} - \hat{\mu}^{\mathrm{low}})] \,/\, \mathrm{Var}[X^{\mathrm{high}}_{k,t}]$, the fraction of minibatches failing the monitoring checks, and the estimated correlation between the paired per-step statistics $(X^{\mathrm{high}}_{k,t}, X^{\mathrm{low}}_{k,t})$. Curves show seed means and bands show 95% bootstrapped confidence intervals for the means.