COHERENCE-GUIDED FACTORIZED FLOW POLICY FOR EFFICIENT HUMANOID WHOLE-BODY CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Flow-matching policies can represent complex, multimodal action distributions, providing an expressive policy model for humanoid whole-body control including locomotion, balance, and physical interaction. However, existing methods commonly optimize all joints within a single flow network, where gradients from heterogeneous body parts can interfere, obscuring reliable improvement directions and constraining effective policy updates. To address this problem, we propose Factorized Flow Policy Optimization (FFPO), which decomposes the whole-body flow policy into functional body agents and improves them sequentially while retaining parallel action generation during execution. Within this sequential process, likelihood-ratio prefixes account for preceding agents' policy updates when evaluating each body update. And body-wise gradient coherence, estimated using a gradient-SNR proxy, guides the allocation of a fixed optimization-epoch budget. FFPO then synchronously rescales the resulting candidate parameter updates according to their aggregate log-ratio magnitude, allowing reliable local improvements to accumulate under a controlled joint policy update. To implement these updates efficiently, FFPO uses a first-order flow-ratio estimator for the prefixes and a differentiable flow-matching proxy for current-agent gradients. Our theoretical analysis characterizes the policy-update cost of uncertain cross-body interactions and establishes conditions under which body-wise factorization yields greater expected improvement. Across 16 simulated locomotion, balance, and loco-manipulation tasks, FFPO improves mean normalized reward AUC by 14.4% and mean normalized final reward by 14.1% over the strongest evaluated baseline for each metric. Deployment on the Unitree G1 further demonstrates coordinated whole-body motion on hardware.

1 INTRODUCTION

Humanoid robots have become a promising platform for general-purpose embodied intelligence via their human-compatible morphology and the ability to combine locomotion with physical interaction (Gu et al., 2026). To control the nonlinear and dynamic humanoids, researchers have explored model-based planning, trajectory optimization, manually designed controllers, reinforcement learning (RL), and, more recently, large pretrained models (Bjorck et al., 2025). Among these approaches, online RL with massively parallel simulation has become a principal paradigm for joint-level whole-body control, since it can acquire task-specific behaviors through scalable trial-and-error interaction in simulation and deploy the policy in real-world environment (Mittal et al., 2025; Makoviychuk et al., 2021; Rudin et al., 2022), achieving substantial progress in locomotion, motion tracking, teleoperation, and physical interaction (Wang et al., 2025; Yin et al., 2025; He et al., 2024a; 2025).

Building on this RL learning paradigm, policy architectures have evolved to model increasingly complex whole-body behaviors. Early approaches typically use multilayer perceptrons (MLP) to predict joint targets directly (He et al., 2024b), whereas Transformer-based policies (Radosavovic et al., 2024) capture longer temporal and long-range dependencies in sensorimotor trajectories. To further represent the multimodal action distributions, recent studies introduce diffusion and flow policies that generate actions through iterative distribution transformation (Ren et al., 2025; McAllister et al., 2026; Yi et al., 2026). Among them, flow matching policies transport a simple base distribution toward the action distribution through a continuous velocity field, providing higher quality and efficiency (Lipman et al., 2022). Specifically, flow matching learns complex distribution via a

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.

tractable regression objective that propagates gradient signals through the generative model without repeatedly evaluating an exact normalized likelihood, making flow policy optimization an attractive approach to high-dimensional humanoid control (Urain et al., 2024).

Nonetheless, applying flow policy optimization to humanoid whole-body control introduces a fundamental scalability problem. Existing methods commonly formulate the robot as a single agent and optimize the complete joint action distribution using one monolithic policy network (Yi et al., 2026; Ding et al., 2025; Chi et al., 2025), even though different body parts have substantially different dynamics and control roles. For example, the legs maintain support and generate locomotion, the torso regulates balance, and the arms and hands perform task-dependent interaction. As the number of joints increases, jointly exploring these heterogeneous action subspaces requires rapidly growing amounts of interaction data and may lead to interfered gradients (Chatterjee, 2020). Meanwhile, a scalar global advantage evaluates the whole-body action without isolating each body's contribution (Foerster et al., 2018; Zhong et al., 2024). Gradients from heterogeneous action subspaces can therefore interfere within a monolithic update, obscuring coherent body-wise directions and consuming policy update budget without proportional improvement. This raises a central question: *how can flow-policy optimization exploit coherent body-wise gradients to improve whole-body behavior under a controlled joint policy update?*

To address this question, we formulate humanoid whole-body control as a multi-agent policy optimization problem and propose Factorized Flow Policy Optimization (FFPO). By factorizing the whole-body flow policy into functional body agents, FFPO improves the agents sequentially while generating their actions in parallel during execution. Within sequential improvement, likelihood-ratio prefixes account for preceding agents' policy updates when evaluating each body update, while body-wise gradient coherence, estimated using a gradient-SNR proxy, guides the allocation of a fixed optimization-epoch budget. FFPO then synchronously rescales the candidate parameter updates according to their aggregate log-ratio magnitude, allowing coherent local improvements to accumulate under a controlled joint policy update. To implement this procedure efficiently, FFPO uses a first-order tangent flow-ratio estimator for the prefixes and a differentiable flow-matching proxy for current-agent gradients. Our theoretical analysis characterizes a *cross-body noise tax* and identifies conditions under which body-wise factorization reduces its cost. Across 16 simulated locomotion, balance, and loco-manipulation tasks, FFPO improves mean normalized reward AUC by 14.4% (0.742 to 0.849) and mean normalized final reward by 14.1% (0.851 to 0.971) over the strongest baseline on each metric. Deployment on the Unitree G1 further demonstrates coordinated whole-body motion on hardware.

2 Related Work

Learning-Based Humanoid Whole-Body Control has progressed through motion priors, motion tracking, sim-to-real adaptation, and interaction-skill learning. Motion-prior methods (Peng et al., 2021; Mu et al., 2026) derive reusable style rewards from unstructured motion clips, whereas motion-tracking methods (Cheng et al., 2024; He et al., 2024b;a) combine retargeted human motions with reinforcement learning to realize expressive locomotion and whole-body teleoperation. These methods achieve broad motion coverage, but rely on curated feasible references, privileged teachers, or manually assigned tracking roles. ASAP (He et al., 2025) improves motion tracking under real-world dynamics by learning a delta-action model from robot rollouts, thereby addressing physics mismatch while retaining the original low-level policy optimizer. For loco-manipulation, SkillMimic (Wang et al., 2025) tracks coupled human-object motion to learn to interact, while HumanX (Wang et al., 2026) generates trajectories via data augmentation and applies interaction-aware rewards to learn generalizable interaction skills.

Flow Matching Policies model multimodal actions through iterative probability transport, offering expressive alternatives to unimodal Gaussian actors. Flow Matching (Lipman et al., 2022; Zhang et al., 2025a) learns a deterministic probability-flow ordinary differential equation and can reduce sampling cost when paired with suitable transport paths. While flow models are often designed for offline learning, online reinforcement learning further requires policy updates to be estimated from newly collected returns. ReinFlow (Zhang et al., 2025b) makes flow likelihoods tractable by injecting learnable noise and treating the resulting transitions as a Markov process, whereas GenPO (Ding et al., 2025) uses exact diffusion inversion and augmented dummy actions; both recover conven-

tional policy ratios but add process-specific stochastic dynamics or model structure. More closely related to our work, FPO (McAllister et al., 2026) and FPO++ (Yi et al., 2026) avoid likelihood evaluation through an advantage-weighted ratio derived from conditional flow-matching losses, yet still optimize the complete action vector as a single policy. PolicyFlow (Yang et al., 2026) provides another likelihood-free route through velocity-field variation along an interpolation path and controls mode collapse with a Brownian regularizer.

Multi-Agent Reinforcement Learning (MARL) addresses the curse of dimensionality caused by the exponential growth of the joint action space as the number of agents increases (Yuan et al., 2023; Feng et al., 2025; Zhang et al., 2021). To make joint optimization tractable, existing methods exploit structured factorization assumptions through two main directions: value-decomposition methods (Rashid et al., 2018; Wang et al., 2021a) factorize a global value function into agent-wise utilities, whereas policy-decomposition methods (Wang et al., 2021b; Zhong et al., 2024) factorize the joint policy into conditionally independent agent policies. A representative policy-decomposition framework is HARL (Kuba et al., 2022; Zhong et al., 2024), which derives multi-agent advantage decomposition and sequential trust-region updates, allowing heterogeneous agents to improve a shared return without parameter sharing. Sequential structures have also been explored architecturally: MAT (Wen et al., 2022) and MADT (Meng et al., 2023) autoregressively map agent observations to a sequence of actions for modeling cooperation.

3 PRELIMINARIES

The humanoid whole-body control could be modeled as a Markov decision process (MDP) (Sutton & Barto, 1998), which is a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ that describes the humanoid control process: given state s_t and whole-body action a_t, a policy $\pi_\theta(a_t \mid s_t)$ maximizes the discounted cumulative return $J(\theta) = \mathbb{E}_{\pi_\theta, P}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$. To further decompose the complex action space and model the heterogeneous body parts of humanoid, Heterogeneous-Agent Reinforcement Learning (Zhong et al., 2024) factorizes a cooperative policy as $\pi_\theta(a \mid s) = \prod_{i=1}^{M} \pi_{\theta_i}(a_i \mid s)$ and updates its agents sequentially under an order $\sigma = (\sigma_1, \ldots, \sigma_M)$. When optimizing agent $i = \sigma_m$, the policy updates of the preceding agents modify the global advantage through

$$R_{<m} = \prod_{j<m} \frac{\pi_{\theta^{\text{new}}_{\sigma_j}}(a_{\sigma_j} \mid s)}{\pi_{\theta^{\text{old}}_{\sigma_j}}(a_{\sigma_j} \mid s)}, \qquad \widetilde{A}_i = \text{stopgrad}(R_{<m})\widehat{A}, \qquad r_i(\theta_i) = \frac{\pi_{\theta_i}(a_i \mid s)}{\pi_{\theta^{\text{old}}_i}(a_i \mid s)}. \tag{1}$$

One feasible HAPPO objective based on the sequential advantage is

$$J_i^{\text{HAPPO}}(\theta_i) = \mathbb{E}_{(s,a)\sim\pi^{\text{old}}}\left[\min\left(r_i(\theta_i)\widetilde{A}_i, \text{clip}\left(r_i(\theta_i), 1-\epsilon, 1+\epsilon\right)\widetilde{A}_i\right)\right]. \tag{2}$$

Thus, the prefix $R_{<m}$ requires an accurate estimate to transmit the effects of preceding updates, whereas the current ratio r_i must remain differentiable to optimize agent i. FFPO preserves this sequential structure while adapting both quantities to implicit flow policies.

We use flow matching (Lipman et al., 2022) to model our policy, which offers an effective approach for modeling the complex whole-body control policy. A conditional flow policy transports a noise sample $\epsilon \sim \mathcal{N}(0, I)$ at flow time $\lambda = 1$ to an action a at $\lambda = 0$. For the linear path $x_\lambda = (1-\lambda)a + \lambda\epsilon$, the velocity field $v_\theta(x_\lambda, \lambda, s)$ is learned with

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{s,a,\epsilon,\lambda}\left[\|v_\theta(x_\lambda, \lambda, s) - (\epsilon - a)\|_2^2\right]. \tag{3}$$

Following FPO-style optimization, the per-sample loss difference $\tilde{\ell}_\theta^{(q)} = \mathcal{L}_{\text{CFM}}^{(q)}(\theta_{\text{old}}) - \mathcal{L}_{\text{CFM}}^{(q)}(\theta)$ acts as an efficient and differentiable first-order pseudo log-ratio (McAllister et al., 2026).

4 METHOD

As shown in Figure 1, FFPO optimizes a body-factorized flow policy through three components. First, FFPO decomposes the policy and enables policy update rescaling toward a target magnitude

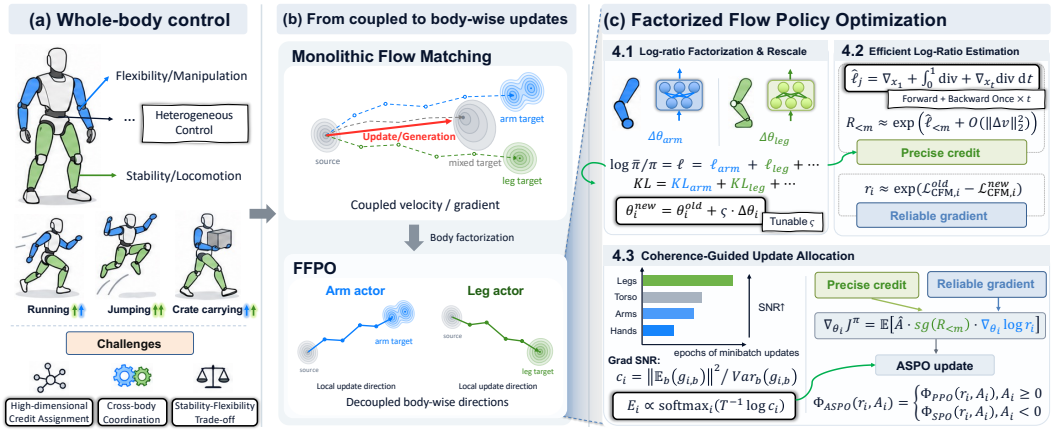


Figure 1: An illustration of FFPO. (a) Humanoid whole-body control faces the challenge of control heterogeneity. (b) Monolithic flow matching methods learn coupled targets. (c) We propose FFPO to decompose the log-ratios, attribute them as credits to different body parts, and adjust the step sizes and update epochs for higher sample efficiency.

(Section 4.1). It then estimates the prefix-ratio and calculates a gradient proxy for the current body (Section 4.2). The prefix-ratio is used for sequential update, while the gradient coherence further determines the allocation of epochs (Section 4.3). Finally, we analyze the effect of cross-body interference on policy improvement (Section 4.4).

4.1 Body-Factorized Policy Update Rescaling

Intuitively, body-wise factorization isolates policy updates from different functional parts, allowing their contributions to the joint update to be controlled explicitly. Given the policy $\pi(a \mid s) = \prod_{i=1}^{M} \pi_i(a_i \mid s)$, the joint log-ratio between candidate and old policies decomposes as

$$\ell = \log \frac{\pi^{\text{cand}}}{\pi^{\text{old}}} = \sum_{i=1}^{M} \log \frac{\pi_i^{\text{cand}}}{\pi_i^{\text{old}}} = \sum_{i=1}^{M} \ell_i. \tag{4}$$

It follows that $D_{\text{KL}}\big(\pi^{\text{old}} \big\| \pi^{\text{cand}}\big) = \sum_{i=1}^{M} D_{\text{KL}}\big(\pi_i^{\text{old}} \big\| \pi_i^{\text{cand}}\big)$, so the joint policy update can be attributed to individual body agents. For candidate updates, FFPO estimates the aggregate update proxy over N rollout samples by

$$\hat{\rho}^2_{\text{cand}} = \frac{1}{N} \sum_{n=1}^{N} \bar{\ell}_n^2, \qquad \bar{\ell}_n = \sum_{i=1}^{M} \text{clip}(\ell_{i,n}, -B_{\tan}, B_{\tan}). \tag{5}$$

Noticing that $D_{\text{KL}}(\pi^{\text{old}} \| \pi^{\text{cand}}) = \frac{1}{2}\mathbb{E}[\ell^2] + O(\mathbb{E}|\ell|^3)$ and $\hat{\rho}^2_{\text{cand}}$ converges in probability to $\mathbb{E}[\bar{\ell}^2] \approx 2D_{\text{KL}}$ when not clipped, $\hat{\rho}_{\text{cand}}$ provides a local square-root KL proxy, with clipping suppressing finite-sample ratio tails. FFPO then controls the target policy-update magnitude $\rho_\star$ through

$$\theta_i^{\text{new}} = \theta_i^{\text{old}} + \varsigma(\theta_i^{\text{cand}} - \theta_i^{\text{old}}), \qquad \varsigma = \text{clip}\left(\frac{\rho_\star}{\hat{\rho}_{\text{cand}} + \varepsilon}, \varsigma_{\min}, \varsigma_{\max}\right). \tag{6}$$

Because local log-ratios are first-order linear in parameter displacement, the same factor gives $\ell_i^{\text{new}} \approx \varsigma \ell_i^{\text{cand}}$ and $\hat{\rho}_{\text{new}} \approx \varsigma \hat{\rho}_{\text{cand}}$. It therefore preserves the relative body-wise allocation while consistently scaling both local and joint policy updates.

4.2 Efficient Sequential Flow-Ratio Estimation

The sequential update requires two policy ratios with distinct roles: previously updated body agents form a detached prefix $\widehat{R}_{<m}$ for credit assignment, whereas the current agent requires a differentiable ratio $\widetilde{r}_i$ for gradient propagation.

For a previously updated agent j, define the interpolated velocity field $v_j^\eta = v_j^{\text{old}} + \eta \Delta v_j$, where $\Delta v_j = v_j^{\text{cand}} - v_j^{\text{old}}$ and $\eta \in [0, 1]$. Along the old-policy path, its first-order perturbation satisfies

$$\dot{x}_t = v_j^{\text{old}}(x_t, t, s), \quad \dot{\delta x}_t = \nabla_x v_j^{\text{old}}(x_t, t, s)\delta x_t + \Delta v_j(x_t, t, s), \quad x_0 = a_j, \quad \delta x_0 = 0. \tag{7}$$

Here, t denotes flow time. FFPO estimates the end-point log-ratio via a *tangent estimator* based on continuity equation as

$$\widehat{\ell}_j^{\tan} = \nabla_{x_1} \log p_1(x_1)^\top \delta x_1 + \int_0^1 \left[\operatorname{div} \Delta v_j + \nabla_x \operatorname{div} v_j^{\text{old}\,\top} \delta x_t\right] \mathrm{d}t, \tag{8}$$

where the velocity fields inside the integral are evaluated at (x_t, t, s) and p_1 is the terminal noise density. This gives the first-order approximation $\ell_j = \log(\pi_j^{\text{cand}}/\pi_j^{\text{old}}) = \widehat{\ell}_j^{\tan} + O(\|\Delta v_j\|_{C^2}^2)$.

The estimator introduces no backpropagation through time. At each integration node, a fused forward-mode pass computes v_j^{old}, $J_t = \nabla_x v_j^{\text{old}}$, and $\chi_t = \operatorname{tr}(J_t)$, while one node-local reverse-mode derivative gives $\nabla_x \chi_t$. Evaluating the updated field at the cached path gives Δv_j and $\operatorname{div} \Delta v_j$ by subtraction. The resulting detached estimates are computed once and accumulated into the clipped prefix

$$H_{m-1} = \sum_{j<m} \operatorname{clip}(\widehat{\ell}_{\sigma_j}^{\tan}, -B_{\tan}, B_{\tan}), \qquad \widehat{R}_{<m} = \exp(\operatorname{clip}(H_{m-1}, -B_{\text{pre}}, B_{\text{pre}}))\,. \tag{9}$$

For the current agent $i = \sigma_m$, FFPO instead uses the differentiable FPO surrogate (McAllister et al., 2026) for efficient and reliable gradients:

$$\widetilde{r}_i^{(q)} = \exp\left(\widetilde{\ell}_i^{(q)}\right), \qquad \widetilde{\ell}_i^{(q)} = \operatorname{clip}\left(\mathcal{L}_{\text{CFM}}^{(q)}(\theta_i^{\text{old}}) - \mathcal{L}_{\text{CFM}}^{(q)}(\theta_i), -B_{\text{CFM}}, B_{\text{CFM}}\right). \tag{10}$$

At $\theta_i = \theta_i^{\text{old}}$, this pseudo log-ratio $\widetilde{\ell}_i = 0$, ratio $\widetilde{r}_i = 1$, and $\nabla_{\theta_i} \widetilde{\ell}_i = -\nabla_{\theta_i} \mathcal{L}_{CFM}$. FFPO therefore combines an accurate detached prefix for sequential credit assignment with an efficient differentiable ratio for the current-agent update. Full derivations are provided in Appendix A.1.

4.3 Coherence-Guided Sequential Policy Optimization

Body agents progress at different rates: upper-body agents may require substantial adaptation to complete an interaction, while leg agents that already maintain stable support may benefit from fewer updates. Under a fixed total epoch budget, FFPO allocates more optimization effort to body agents with higher gradient coherence, concentrating more update on effective directions. Concretely, for minibatch gradients $\{g_{i,b}\}_{b=1}^K$, we distribute the total optimization epoch E_{total} by estimating $E_i = E_{\min} + \lfloor (E_{\text{total}} - M E_{\min}) w_i \rfloor$, where $w_i = \operatorname{softmax}_i \left(T^{-1} \log \text{EMA}(c_i)\right), c_i = \|\bar{g}_i\|_2^2 / (K^{-1} \sum_{b=1}^K \|g_{i,b}\|_2^2 - \|\bar{g}_i\|_2^2 + \varepsilon)$. The remaining $E_{\text{total}} - \sum_i E_i$ epochs are uniformly randomly distributed to each body agent. Here, c_i is the signal-to-noise ratio of the body-wise gradient that measures its coherence, and E_i is the optimization epochs on one batch during each iteration. The individual advantage $\tilde{A}_i = \operatorname{stopgrad}(\widehat{R}_{<m})\hat{A}$, and the current agent maximizes the Asymmetric Simple Policy Optimization (ASPO) (Yi et al., 2026) objective for more stable training:

$$J_i = \mathbb{E}\left[\Phi_{\text{ASPO}}(\tilde{r}_i^{(q)}, \tilde{A}_i)\right], \; \Phi_{\text{ASPO}}(r, A) = \begin{cases} \min\left(rA, \operatorname{clip}(r, 1-\epsilon_c, 1+\epsilon_c)A\right), & A \geq 0, \\ rA - \dfrac{|A|}{2\epsilon_c}(r-1)^2, & A < 0. \end{cases} \tag{11}$$

Each iteration computes a centralized advantage, allocates body-wise epochs, sequentially forms candidate updates and cached prefixes, and jointly rescales the candidate displacement using the aggregate log-ratio scale. The critic remains centralized, and all body agents generate actions in parallel during rollout and deployment. More details are presented in Appendix C.

4.4 Theoretical Results: Cross-Body Noise Tax

We analyze how body factorization affects the policy gradient through local Fisher geometry analysis. First, we consider the body-factorized policy $\pi_{\boldsymbol{\theta}}(a \mid s) = \prod_{i=1}^{M} \pi_{\theta_i}(a_i \mid s)$ and a monolithic policy $\pi_\phi(a \mid s)$, whose objectives are

$$J_{\mathrm{B}}(\boldsymbol{\theta}) = \mathbb{E}_{\tau\sim\pi_{\boldsymbol{\theta}}}[R(\tau)], \qquad J_{\mathrm{M}}(\phi) = \mathbb{E}_{\tau\sim\pi_\phi}[R(\tau)], \tag{12}$$

respectively. For $q \in \{\mathrm{B}, \mathrm{M}\}$, define the ordinary policy gradient $g_q = \nabla J_q$, the Fisher information $F_q = \mathbb{E}[\nabla \log \pi_q \nabla \log \pi_q^\top]$, and the natural-gradient direction $d_q = F_q^{-1} g_q$. Locally, the monolithic direction decomposes as $d_{\mathrm{M}} = d_{\mathrm{B}} + d_\times$, where $d_\times$ is Fisher-orthogonal to d_{B} and captures the remaining cross-body policy update (see Lemma 3 in Appendix A.2).

Then we can analyze the interference caused by the cross-body term $d_\times$. For $k \in \{\mathrm{B}, \times\}$, let $\widehat{d}_k = d_k + \zeta_k$ be the minibatch estimate, F_k denotes the Fisher metric restricted in tangent subspace, $I_k = d_k^\top F_k d_k$ be the signal, $N_k = \mathrm{tr}(F_k \mathrm{Cov}(\zeta_k))$ be the noise, $C_k = I_k + N_k$ measure the total update and $\tau_k = N_k / C_k$ is the noise fraction. Then we have the following result:

Theorem 1. *Under the local matched-policy conditions detailed in Appendix A, policy parameter updates rescaled to the same expected local KL δ satisfy*

$$\Delta J_{\mathrm{M}}(\delta) \approx \sqrt{2\delta}\, \frac{(1-\tau_{\mathrm{B}})C_{\mathrm{B}} + (1-\tau_\times)C_\times}{\sqrt{C_{\mathrm{B}} + C_\times}}, \qquad \Delta J_{\mathrm{B}}(\delta) \approx \sqrt{2\delta}\,(1-\tau_{\mathrm{B}})\sqrt{C_{\mathrm{B}}}, \tag{13}$$

and, for fixed cross-body minibatch noise $N_\times$,

$$\left.\frac{\partial \Delta J_{\mathrm{M}}}{\partial \tau_\times}\right|_{N_\times} = -\sqrt{2\delta}\, \frac{N_\times}{\tau_\times^2} \frac{\frac{1}{2}(I_{\mathrm{B}} + I_\times) + N_{\mathrm{B}} + N_\times}{(C_{\mathrm{B}} + C_\times)^{3/2}} < 0, \quad \left.\frac{\partial \Delta J_B}{\partial \tau_\times}\right|_{I_B, N_B} = 0 \tag{14}$$

Thus, noisy cross-body update consumes the KL budget δ without proportional improvement: a *cross-body noise tax*. FFPO removes it through body-wise factorization and concentrates its policy update on the more reliable body gradients. Furthermore, $\Delta J_B > \Delta J_M$ in the first order sense if and only if $I_B^2 N_\times > C_B I_\times^2 + (2C_B I_B - I_B^2) I_\times$, indicating that factorization dominates joint optimization when the cross-body signal $I_\times$ is insufficient against the cross-body noise $N_\times$.

5 Experiments

We evaluate FFPO to answer the following research questions: (1) Can FFPO improve learning efficiency and final performance? (Section 5.2) (2) How does FFPO affect the emergence of coordinated locomotion? (Section 5.3) (3) What are the contributions of each component in FFPO? (Section 5.4) (4) Is FFPO deployable on a real-world humanoid robot? (Section 5.5)

5.1 Experimental Setup

We evaluate FFPO on 16 humanoid whole-body control tasks in MuJoCo-Warp (Google DeepMind & NVIDIA, 2025). The benchmark includes 11 locomotion and balance tasks: Stand, Walk, Run, Step-Up, Turn-Jog, One-Foot Balance, Normal Walk, March, Goose Step, Frog Jump, and Zombie Hop, which cover basic gait learning, static and dynamic balance, terrain traversal, stylized motion tracking, and inter-limb coordination. It further includes 5 loco-manipulation tasks: Bimanual Lift, Door Open, Pick-Carry-Place, Crate Carry, and Box Push, which require coordinated locomotion, balance, object interaction, and sustained force control. Among these tasks, Stand, Walk, Run, Step-Up are reference-free and the others are based on corresponding demonstrations.

We compare FFPO against five online methods. (1) Gaussian PPO uses a diagonal-Gaussian actor with the clipped PPO objective. (2) FPO replaces the Gaussian actor with a conditional flow model and estimates policy ratios via conditional flow-matching loss. (3) FPO++ strengthens FPO estimator with per-sample pseudo likelihood ratios and asymmetric clipping. (4) GenPO recovers policy ratios by inverting the generative process, whereas (5) PolicyFlow directly regularizes velocity-field updates to stabilize online flow-policy learning. Further experiment, evaluation and baseline details are presented in Appendix B.

Table 1: Aggregate normalized training performance with 95% stratified-bootstrap confidence intervals. The reward values are normalized for aggregation across tasks.

Algorithm	AUC mean ↑	AUC IQM ↑	Final Mean ↑	Final IQM ↑	Drawdown Mean ↓
Gaussian PPO	0.437 [0.410, 0.463]	0.480 [0.440, 0.517]	0.161 [0.077, 0.240]	0.158 [0.018, 0.287]	0.517 [0.441, 0.594]
FPO	-0.683 [-0.793, -0.569]	-0.430 [-0.511, -0.369]	-0.962 [-1.107, -0.822]	-0.732 [-0.878, -0.628]	1.595 [1.457, 1.735]
FPO++	0.719 [0.702, 0.736]	0.817 [0.813, 0.819]	0.678 [0.663, 0.693]	**0.979 [0.967, 0.990]**	0.314 [0.302, 0.327]
GenPO	0.742 [0.738, 0.747]	0.800 [0.795, 0.805]	0.851 [0.835, 0.867]	0.897 [0.888, 0.907]	**0.005 [0.002, 0.008]**
PolicyFlow	0.623 [0.611, 0.635]	0.673 [0.661, 0.684]	0.790 [0.780, 0.800]	0.890 [0.876, 0.903]	0.030 [0.027, 0.032]
FFPO	**0.849 [0.841, 0.857]**	**0.875 [0.865, 0.883]**	**0.971 [0.959, 0.982]**	**0.976 [0.968, 0.982]**	**0.014 [0.008, 0.021]**

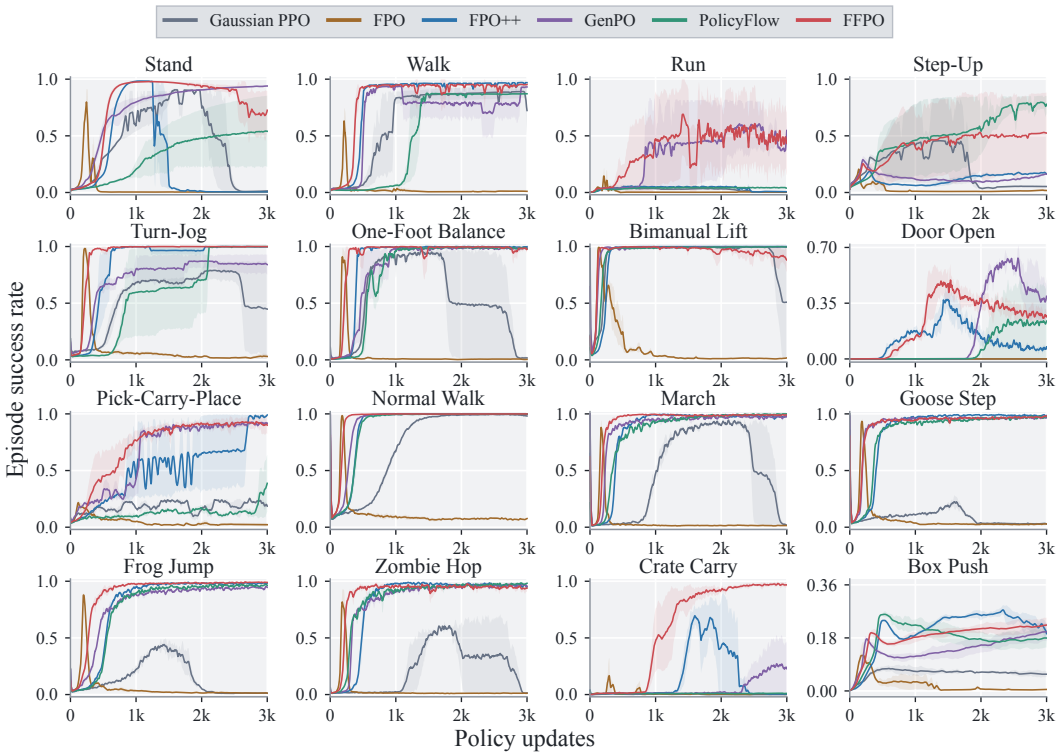
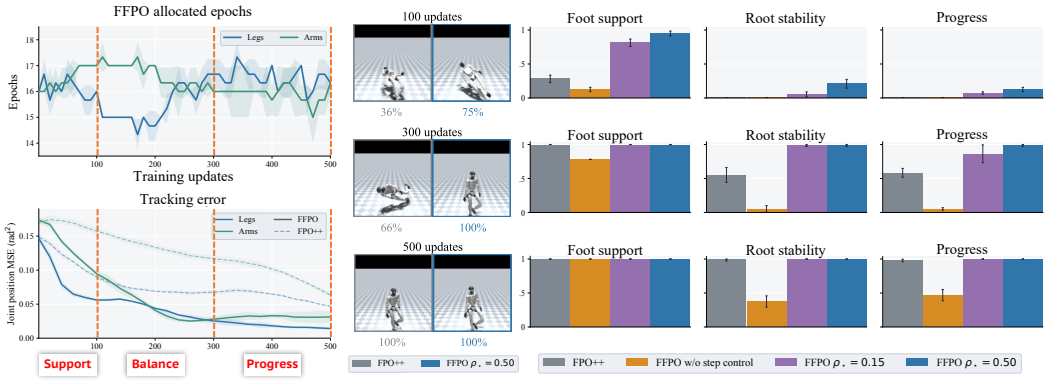


Figure 2: Task success rate curves on all 16 tasks across 5 seeds. Each curve shows the mean and the shaded region denotes one standard deviation.

5.2 Main Results

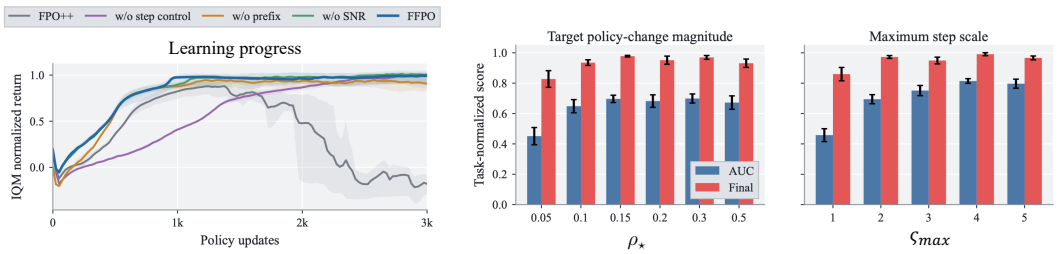
Task success rate in Figure 2 reveals a tension between rapid improvement and reliable convergence. Gaussian PPO has the least expensive actor update, but its learning curves improve more slowly and frequently deteriorate late in training as shown by the lower AUC of 0.437 and higher drawdown of 0.517 in Table 1. FPO often makes rapid initial progress, yet this gain is followed by severe collapse on many tasks, indicated by the highest drawdown. FPO++ substantially stabilizes flow-policy learning and reaches high peak returns, although late degradation remains visible on tasks such as Stand and Crate Carry, leading to 0.979 Final IQM but 0.678 Final Mean. These trajectories show that fast early optimization alone does not ensure that the acquired policy remains within a reliable improvement region.

The more conservative baselines reduce this instability but also introduce lower efficiency. As shown in Table 1 and Figure 2, GenPO has the smallest aggregate drawdown and remains stable throughout training, whereas PolicyFlow exhibits similar stability with slower improvement. However, both converge below FFPO on coordination-intensive tasks such as Turn-Jog, Frog Jump, and Crate Carry. Qualitative rollouts further show that their stable solutions can omit fine motion details, suggesting convergence to locally adequate but less coordinated behaviors. FFPO resolves this trade-off more consistently: its body-wise optimization preserves stable support behavior while allowing task-relevant body agents to continue improving, yielding faster learning across both locomotion and object-interaction tasks as witnessed by the highest normalized AUC mean of 0.849. Performance profile curves in Figure 7 of Appendix D.2 also show that FFPO consistently exhibits faster learning and better final performance.



(a) Allocated epochs and tracking errors (b) Motion completion (c) Fraction of behavioral modes

Figure 3: Mechanism analysis on Normal Walk. (a) shows the allocated epochs of FFPO and the tracking error curves of FPO++ and FFPO. (b) shows the behavior and motion completion rate of policies trained by FPO++ and FFPO. (c) shows the fraction of three types of behaviors at 100, 300 and 500 updates.



(a) Component ablation (b) Sensitivity analysis

Figure 4: Component ablation and sensitivity analysis on Stand, Turn-Jog, and Crate Carry.

5.3 Mechanism Analysis

Figure 3 traces how body-wise optimization translates into walking behavior. At 100 updates, FFPO achieves alternating foot support in approximately 95% of rollouts, while root stability and forward progress remain infrequent. Meanwhile, leg tracking error has already fallen sharply, indicating that a reliable support pattern emerges before sustained locomotion. During the subsequent phase at 100-300 updates, FFPO allocates slightly more epochs to the arms on average, while arm tracking error decreases and root stability improves, suggesting that coordinated upper-body motion contributes to balance regulation. With support and stability established, forward progress rises to nearly all rollouts by 300 updates, when FFPO completes the reference motion and FPO++ reaches only 66% completion. A smaller policy-update target delays this transition, while removing step control leaves stability and progress low over the same interval. By 500 updates, leg and arm allocations approach parity as both parts refine coordinated walking. These stages illustrate how body-wise allocation and controlled joint policy updates help reliable local improvements accumulate into whole-body locomotion. We further investigate the effect of body factorization, gradient coherence and the efficiency of ratio estimator in Appendix D.3-D.5, showcasing that accurate prefix and reliable gradient lead to better efficiency.

5.4 Ablation Studies and Sensitivity Analysis

We dissect the contributions of each component within FFPO and conduct experiments on three representative tasks Stand, Turn-Jog, and Crate Carry. As the aggregated learning curves in Figure 4(a) show, removing policy-step control substantially slows learning, while removing prefix correction reduces both learning speed and terminal performance. SNR-based epoch allocation provides a

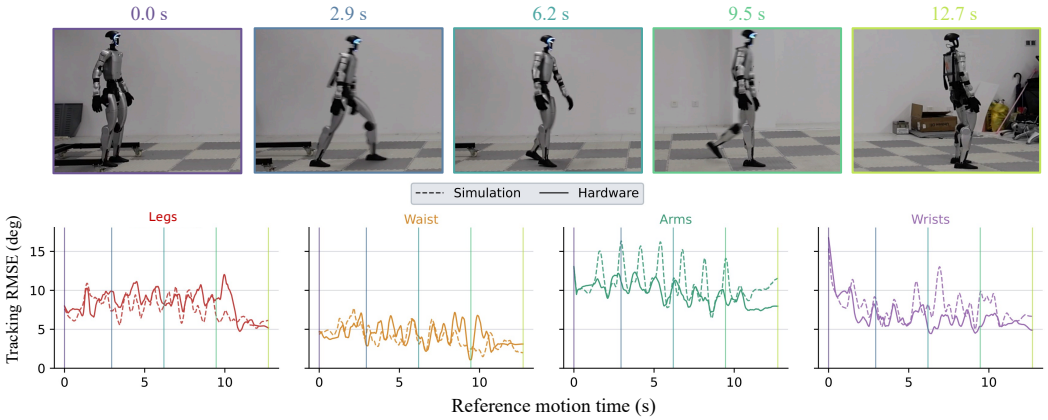


Figure 5: Real-world deployment on Unitree G1 and body-wise tracking errors in simulation and hardware for Normal Walk.

smaller task-dependent efficiency gain while preserving comparable asymptotic performance. In contrast, monolithic FPO++ exhibits a pronounced late-training collapse, highlighting the stability benefit of body-wise optimization. Figure 4(b) shows the effect of two core hyperparameters of FFPO that control the balance between update magnitude and stability: performance remains high for $\rho_\star \in [0.1, 0.3]$ and $\varsigma_{\max} \in [2, 5]$, with $\rho_\star = 0.15$ and $\varsigma_{\max} = 4$ providing the best overall balance between learning efficiency and final performance. We defer more ablation analysis on the sensitivity of the number of body agents, SNR temperature, SNR-EMA decay, minimum epochs and component composition to Appendix D.6.

5.5 Sim2Real Evaluation

We deploy FFPO policies on a Unitree G1 humanoid with 29 DoF using only proprioceptive observations and a 50 Hz control frequency. The deployment suite contains Normal Walk, Goose Step, and March, with Normal Walk presented in the main text and the remaining motions reported in Appendix D.7. We deploy the FFPO checkpoint trained for 3,000 updates and fine-tuned for 500 additional updates in simulation with domain randomization and observation noise. As shown in Figure 5, the robot maintains balance and completes the full 12.7-second reference motion without external perception. Moreover, the legs, waist, arms, and wrists agents exhibit consistently low tracking errors on hardware, with temporal error profiles closely matching those observed in simulation. These results indicate that the coordinated behavior learned by the factorized body agents transfers coherently from simulation to the physical G1.

6 Conclusion and Future Work

In this work, we proposed Factorized Flow Policy Optimization (FFPO) to address unstable and inefficient policy improvement caused by heterogeneous body-wise gradients and uncertain cross-body interactions in monolithic flow policies. FFPO performs prefix-corrected sequential improvement with body-wise epoch allocation guided by gradient coherence, and synchronously rescales candidate parameter updates to control the joint policy-update magnitude. It combines accurate detached prefix estimation with an efficient differentiable surrogate for current-body optimization. Our theoretical analysis shows the cross-body noise tax such that uncertain cross-body updates consume the policy-update budget without yielding proportional improvement. Experiments across 16 humanoid whole-body control tasks demonstrate that FFPO improves learning efficiency and final reward by about 14% over the strongest baseline, while deployment on Unitree G1 validates its practical feasibility. The current study focuses on humanoid morphology with manually specified body partitions and a limited set of physical-interaction tasks. Future work will investigate adaptive body decomposition, more dexterous and long-horizon interactions, and broader evaluation across robot morphologies and real-world operating conditions.

AI USE STATEMENT

In this work, we used generative AI tools to assist with related-work searches, language editing, code development, illustration creating (Figure 1), verification of theoretical derivations, and general manuscript proofreading. Generative AI was not used to produce or alter experimental data. All quantitative results and figures were derived from actual experimental runs, and their analysis and interpretation were performed by the authors. We carefully reviewed all AI-assisted text, independently checked the cited literature and theoretical arguments, and manually tested AI-assisted code for correctness and reliability. We take full responsibility for the final content of this work, including all text, claims, code, results, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work develops a reinforcement-learning method for improving the training efficiency of humanoid robot control policies. It does not involve human subjects, personal or sensitive information, or decisions concerning individuals, and it introduces no discrimination, bias, or fairness concerns. Existing motion data and robotic assets are used in accordance with their applicable licenses and research terms. Hardware experiments are conducted in controlled environments with predefined safety constraints and operator supervision. We identify no direct harmful applications, privacy or security risks, legal compliance concerns, or conflicts of interest arising from this study.

REPRODUCIBILITY STATEMENT

We provide the information required to reproduce FFPO and its evaluation throughout the main paper, appendices, and supplementary materials. The appendices specify the complete algorithmic procedure, network architecture, hyperparameters, training protocol, evaluation protocol, task configurations, and implementation details. They also document the motion data sources and preprocessing procedures, motion-auditing and sampling protocols, and the observation and action interfaces. For the theoretical results, we state the required assumptions and provide complete derivations and additional discussion in the theoretical appendix. An anonymized web page containing the code and configurations for reproducing the reported experiments, and corresponding video demos, is released at `https://ffpo.pages.dev/`.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in neural information processing systems*, volume 34, pp. 29304–29320, 2021.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *International Conference on Learning Representations*, 2020.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in neural information processing systems*, volume 31, 2018.

Xuxin Cheng, Yandong Ji, Junming Chen, Ruihan Yang, Ge Yang, and Xiaolong Wang. Expressive whole-body control for humanoid robots. *arXiv preprint arXiv:2402.16796*, 2024.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Shutong Ding, Ke Hu, Shan Zhong, Haoyang Luo, Weinan Zhang, Jingya Wang, Jun Wang, and Ye Shi. Genpo: Generative diffusion models meet on-policy reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, pp. 130443–130474, 2025.

Zhaohan Feng, Ruiqi Xue, Lei Yuan, Yang Yu, Ning Ding, Meiqin Liu, Bingzhao Gao, Jian Sun, Xinhu Zheng, and Gang Wang. Multi-agent embodied ai: Advances and future directions. *arXiv preprint arXiv:2505.05108*, 2025.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Google DeepMind and NVIDIA. MuJoCo Warp (MJWarp): GPU-Optimized MuJoCo for NVIDIA Hardware, 2025. URL `https://github.com/google-deepmind/mujoco_warp`.

Zhaoyuan Gu, Junheng Li, Wenlan Shen, Wenhao Yu, Zhaoming Xie, Stephen McCrory, Xianyi Cheng, Abdulaziz Shamsah, Robert Griffin, C Karen Liu, et al. Humanoid locomotion and manipulation: Current progress and challenges in control, planning, and learning. *IEEE/ASME Transactions on Mechatronics*, 31(2):2300–2330, 2026.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris M. Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. In *8th Annual Conference on Robot Learning*, 2024a.

Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8944–8951, 2024b.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, et al. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2022.

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

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Ian Mason, Sebastian Starke, and Taku Komura. Real-time style modelling of human locomotion via feature-wise transformations and local motion phases. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 5(1):1–18, 2022.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *International Conference on Learning Representations*, volume 2026, pp. 36352–36372, 2026.

Linghui Meng, Muning Wen, Chenyang Le, Xiyun Li, Dengpeng Xing, Weinan Zhang, Ying Wen, Haifeng Zhang, Jun Wang, Yaodong Yang, et al. Offline pre-trained multi-agent decision transformer. *Machine Intelligence Research*, 20(2):233–248, 2023.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Munoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

Yuxuan Mu, Ziyu Zhang, Yi Shi, Dun Yang, Minami Matsumoto, Kotaro Imamura, Guy Tevet, Chuan Guo, Michael Taylor, Chang Shu, Pengcheng Xi, and Xue Bin Peng. Smp: Reusable score-matching motion priors for physics-based character control. *ACM Trans. Graph.*, 45(4), 2026.

NVIDIA. Isaac Sim, 2025. URL `https://github.com/isaac-sim/IsaacSim`.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4):1–20, 2021.

Ilija Radosavovic, Bike Zhang, Baifeng Shi, Jathushan Rajasegaran, Sarthak Kamat, Trevor Darrell, Koushil Sreenath, and Jitendra Malik. Humanoid locomotion as next token prediction. In *Advances in neural information processing systems*, volume 37, pp. 79307–79324, 2024.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304, 2018.

Allen Ren, Justin Lidard, Lars Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, volume 2025, pp. 77288–77329, 2025.

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.

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

Richard S Sutton and Andrew Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

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

Julen Urain, Ajay Mandlekar, Yilun Du, Mahi Shafiullah, Danfei Xu, Katerina Fragkiadaki, Georgia Chalvatzaki, and Jan Peters. Deep generative models in robotics: A survey on learning from multimodal demonstrations. *arXiv preprint arXiv:2408.04380*, 2024.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021a.

Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. DOP: Off-policy multi-agent decomposed policy gradients. In *International Conference on Learning Representations*, 2021b.

Yinhuai Wang, Qihan Zhao, Runyi Yu, Hok Wai Tsui, Ailing Zeng, Jing Lin, Zhengyi Luo, Jiwen Yu, Xiu Li, Qifeng Chen, et al. Skillmimic: Learning basketball interaction skills from demonstrations. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17540–17549, 2025.

Yinhuai Wang, Qihan Zhao, Yuen Fui Lau, Runyi Yu, Hok Wai Tsui, Qifeng Chen, Jingbo Wang, Jiangmiao Pang, and Ping Tan. Humanx: Toward agile and generalizable humanoid interaction skills from human videos. *arXiv preprint arXiv:2602.02473*, 2026.

Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16509–16521, 2022.

Shunpeng Yang, Ben Liu, and Hua Chen. Policyflow: Policy optimization with continuous normalizing flow in reinforcement learning. In *International Conference on Learning Representations*, 2026.

Brent Yi, Hongsuk Choi, Himanshu Gaurav Singh, Xiaoyu Huang, Takara E Truong, Carmelo Sferrazza, Yi Ma, Rocky Duan, Pieter Abbeel, Guanya Shi, et al. Flow policy gradients for robot control. *arXiv preprint arXiv:2602.02481*, 2026.

Shaofeng Yin, Yanjie Ze, Hong-Xing Yu, C Karen Liu, and Jiajun Wu. Visualmimic: Visual humanoid loco-manipulation via motion tracking and generation. *arXiv preprint arXiv:2509.20322*, 2025.

Lei Yuan, Ziqian Zhang, Lihe Li, Cong Guan, and Yang Yu. A survey of progress on cooperative multi-agent reinforcement learning in open environment. *arXiv preprint arXiv:2312.01058*, 2023.

Weishuai Zeng, Kangning Yin, Xiaojie Niu, Shunlin Lu, Weixiang Zhong, Jiahe Chen, Feiyu Jia, Xiao Chen, Zirui Wang, Furui Xu, et al. Scaling behavior foundation model for humanoid robots. *arXiv preprint arXiv:2607.15163*, 2026.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 14754–14762, 2025a.

Tonghe Zhang, Chao Yu, Sichang Su, and Yu Wang. Reinflow: Fine-tuning flow matching policy with online reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 106282–106319, 2025b.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

A Derivations

A.1 First-Order Flow-Ratio Estimation

This section derives the detached prefix-ratio estimator used in Section 4.2. For a previously updated body j, define the interpolated velocity field

$$v_j^\eta = v_j^{\mathrm{old}} + \eta \Delta v_j, \qquad \Delta v_j = v_j^{\mathrm{cand}} - v_j^{\mathrm{old}}, \qquad \eta \in [0,1]. \tag{15}$$

Let x_t^η solve $\dot{x}_t^\eta = v_j^\eta(x_t^\eta, t, s)$ from the fixed action $x_0^\eta = a_j$ to the terminal noise state x_1^η. The density transported by this flow satisfies the continuity equation (Chen et al., 2018)

$$\partial_t p_t^\eta(x) + \nabla_x \cdot \big(p_t^\eta(x) v_j^\eta(x,t,s)\big) = 0, \tag{16}$$

and therefore, along its characteristic path,

$$\frac{\mathrm{d}}{\mathrm{d}t} \log p_t^\eta(x_t^\eta) = -\operatorname{div} v_j^\eta(x_t^\eta, t, s). \tag{17}$$

Because FFPO follows the action-at-$t=0$ to noise-at-$t=1$ convention, integration gives

$$\log \pi_j^\eta(a_j \mid s) = \log p_1(x_1^\eta) + \int_0^1 \operatorname{div} v_j^\eta(x_t^\eta, t, s)\, \mathrm{d}t. \tag{18}$$

Define the path perturbation $\delta x_t = \partial_\eta x_t^\eta|_{\eta=0}$. Differentiating the flow equation at $\eta = 0$ yields the forward sensitivity equation

$$\dot{\delta x}_t = \nabla_x v_j^{\mathrm{old}}(x_t, t, s)\delta x_t + \Delta v_j(x_t, t, s), \qquad \delta x_0 = 0. \tag{19}$$

Differentiating the log-density identity along the same path gives the tangent estimator:

$$\begin{aligned} \widehat{\ell}_j^{\tan} &:= \frac{\partial}{\partial \eta} \log \pi_j^\eta(a_j \mid s)\bigg|_{\eta=0} \\ &= \nabla_{x_1} \log p_1(x_1)^\top \delta x_1 + \int_0^1 \left[\operatorname{div} \Delta v_j(x_t,t,s) + \nabla_x \operatorname{div} v_j^{\mathrm{old}}(x_t,t,s)^\top \delta x_t\right] \mathrm{d}t. \end{aligned} \tag{20}$$

Proposition 2 (First-order accuracy of the tangent estimator)**.** *Fix (a_j, s) and let $v_j^\eta = v_j^{\mathrm{old}} + \eta \Delta v_j$ for $\eta \in [0,1]$. Assume that the corresponding flow paths remain in a compact set, that v_j^η has uniformly bounded spatial derivatives through order three in a neighborhood of these paths, and that $\log p_1$ has bounded derivatives through order two at their endpoints. Then, for sufficiently small $\|\Delta v_j\|_{C^2}$,*

$$\ell_j := \log \pi_j^{\mathrm{cand}}(a_j \mid s) - \log \pi_j^{\mathrm{old}}(a_j \mid s) = \widehat{\ell}_j^{\tan} + O\big(\|\Delta v_j\|_{C^2}^2\big). \tag{21}$$

Proof. Let $f(\eta) = \log \pi_j^\eta(a_j \mid s)$. Differentiating the flow path with respect to η at zero gives the tangent dynamics in Equation 19. The continuity equation then yields $f'(0) = \widehat{\ell}_j^{\tan}$ as in Equation 20. Under the stated bounds, the second-order flow variations and density derivatives satisfy $\sup_{\eta \in [0,1]} |f''(\eta)| \le C\|\Delta v_j\|_{C^2}^2$. Taylor's theorem gives the result. □

The estimator can be evaluated without backpropagation through the ODE solver. At every integration node, a fused forward-mode pass through the old velocity network produces v_j^{old}, its state Jacobian $J_t = \nabla_x v_j^{\mathrm{old}}$, and its exact divergence $\chi_t = \mathrm{tr}(J_t)$. A node-local reverse-mode derivative of the scalar χ_t produces $\nabla_x \chi_t$, while evaluating the updated network at the same cached states gives Δv_j and $\mathrm{div}\, \Delta v_j$ by subtraction. The recurrences $J_t \delta x_t + \Delta v_j$ and $\mathrm{div}\, \Delta v_j + \nabla_x \chi_t^\top \delta x_t$ are then integrated forward together. All old-policy nodes are evaluated as a batched network call and cached once for the rollout; subsequent body updates reuse this detached geometry. Consequently, the estimator uses only node-local Jacobian operations and a forward tangent recurrence, with no backpropagation through time. See Algorithm 2 for details.

For the current body, the differentiable FPO surrogate has the complementary local expansion

$$\widetilde{\ell}_i^{(q)}(\theta_i^{\mathrm{old}} + \Delta\theta_i) = -\left[\nabla_{\theta_i} \mathcal{L}_{\mathrm{CFM},i}^{(q)}(\theta_i)\Big|_{\theta_i=\theta_i^{\mathrm{old}}}\right]^\top \Delta\theta_i + O(\|\Delta\theta_i\|^2). \tag{22}$$

because $\widetilde{\ell}_i^{(q)} = \mathcal{L}_{\mathrm{CFM},i}^{(q)}(\theta_i^{\mathrm{old}}) - \mathcal{L}_{\mathrm{CFM},i}^{(q)}(\theta_i)$. Thus, the tangent estimator supplies an accurate detached first-order prefix, while the CFM surrogate preserves the first-order gradient path for the current body.

A.2 Proof of Theorem 1

We first state the local conditions and two intermediate results, and then derive the KL-normalized improvements in Theorem 1.

Assumption 1. *The body-factorized and monolithic policy classes represent the same local policy distribution, use the same objective and advantage, and are compared under the exact Fisher metric. The tangent directions expressible by the body-factorized policy form a local subspace $\mathcal{T}_{\mathrm{B}}$ of the monolithic policy tangent space.*

Lemma 3 (body-wise Fisher projection)**.** *Let P_{B}^F denote the Fisher-orthogonal projection onto $\mathcal{T}_{\mathrm{B}}$. Under Assumption 1,*

$$d_{\mathrm{B}} = P_{\mathrm{B}}^F d_{\mathrm{M}}, \qquad d_\times = (I - P_{\mathrm{B}}^F) d_{\mathrm{M}}, \tag{23}$$

such that

$$d_{\mathrm{M}} = d_{\mathrm{B}} + d_\times, \qquad d_{\mathrm{B}}^\top F_{\mathrm{M}} d_\times = 0. \tag{24}$$

Proof. The monolithic natural gradient is the Riesz representation of the objective differential under the Fisher inner product: $g_{\mathrm{M}}^\top u = \langle d_{\mathrm{M}}, u \rangle_F$. Restricting this identity to $u \in \mathcal{T}_{\mathrm{B}}$ gives $P_{\mathrm{B}}^F d_{\mathrm{M}}$ as the natural gradient of the body-factorized objective. The second identity follows from the defining orthogonality of the projection. □

For the stochastic comparison, apply the same fixed projection to every monolithic direction, so $\widehat{d}_{\mathrm{M}} = \widehat{d}_{\mathrm{B}} + \widehat{d}_\times$ and the mixed Fisher term vanishes samplewise. Write $\widehat{d}_k = d_k + \zeta_k$, $\mathbb{E}[\zeta_k] = 0$, and $\Sigma_k = \mathrm{Cov}(\zeta_k)$ for $k \in \{\mathrm{B}, \times\}$.

Lemma 4 (local policy update and expected improvement)**.** *Let*

$$I_k = d_k^\top F_k d_k, \qquad N_k = \mathrm{tr}(F_k \Sigma_k), \qquad C_k = I_k + N_k. \tag{25}$$

For a step of size α, the expected local KL changes and first-order improvements satisfy

$$\mathbb{E}[D_{\mathrm{KL}}^{\mathrm{M}}] = \frac{\alpha^2}{2}(C_{\mathrm{B}} + C_\times) + O(\alpha^3), \qquad \mathbb{E}[\Delta J_{\mathrm{M}}] = \alpha(I_{\mathrm{B}} + I_\times) + O(\alpha^2), \tag{26}$$

$$\mathbb{E}[D_{\mathrm{KL}}^{\mathrm{B}}] = \frac{\alpha^2}{2} C_{\mathrm{B}} + O(\alpha^3), \qquad \mathbb{E}[\Delta J_{\mathrm{B}}] = \alpha I_{\mathrm{B}} + O(\alpha^2). \tag{27}$$

Proof. The local KL expansion is $D_{\mathrm{KL}} = \frac{1}{2}\alpha^2 \widehat{d}^\top F \widehat{d} + O(\alpha^3)$. Fisher orthogonality removes the mixed term, and

$$\mathbb{E}[\widehat{d}_k^\top F_k \widehat{d}_k] = d_k^\top F_k d_k + \mathrm{tr}(F_k \Sigma_k) = C_k. \tag{28}$$

The first-order objective change is $g^\top(\alpha \widehat{d})$. Unbiasedness removes ζ_k, while Lemma 3 gives $g_{\mathrm{M}}^\top d_{\mathrm{M}} = I_{\mathrm{B}} + I_\times$ and $g_{\mathrm{B}}^\top d_{\mathrm{B}} = I_{\mathrm{B}}$. □

We now prove Theorem 1. Matching the expected local KL in Lemma 4 to δ gives

$$\alpha_{\mathrm{M}} = \sqrt{\frac{2\delta}{C_{\mathrm{B}} + C_\times}}, \qquad \alpha_{\mathrm{B}} = \sqrt{\frac{2\delta}{C_{\mathrm{B}}}}. \tag{29}$$

Substituting these step sizes into the expected improvements and using $I_k = (1 - \tau_k)C_k$ yields

$$\Delta J_{\mathrm{M}}(\delta) \approx \sqrt{2\delta}\, \frac{(1-\tau_{\mathrm{B}})C_{\mathrm{B}} + (1-\tau_\times)C_\times}{\sqrt{C_{\mathrm{B}} + C_\times}}, \qquad \Delta J_{\mathrm{B}}(\delta) \approx \sqrt{2\delta}\,(1-\tau_{\mathrm{B}})\sqrt{C_{\mathrm{B}}}. \tag{30}$$

For fixed $N_\times$, the identities $C_\times = N_\times/\tau_\times$ and $I_\times = N_\times(1-\tau_\times)/\tau_\times$ give

$$\left.\frac{\partial \Delta J_{\mathrm{M}}}{\partial \tau_\times}\right|_{N_\times} = -\sqrt{2\delta}\, \frac{N_\times}{\tau_\times^2}\, \frac{\frac{1}{2}(I_{\mathrm{B}} + I_\times) + N_{\mathrm{B}} + N_\times}{(C_{\mathrm{B}} + C_\times)^{3/2}} < 0, \tag{31}$$

which completes the proof. The analysis uses exact local natural gradients to expose the policy-space geometry. FFPO approximates this construction with finite-batch CFM gradients, sequential ratio estimates, and optimizer-generated candidate steps; the empirical diagnostics evaluate the corresponding approximation quality.

Table 2: Overview of the humanoid whole-body control tasks.

Task family	Tasks	Capability tested
Reference-free locomotion	Stand; Walk; Run; Step-Up	Basic gait learning, command following, static and dynamic balance, and terrain traversal without reference-motion supervision.
Reference-based locomotion	Turn-Jog; One-Foot Balance; Normal Walk; March; Goose Step; Frog Jump; Zombie Hop	Natural and stylized motion tracking, balance transitions, unilateral support, dynamic jumping, and coordinated inter-limb control.
Loco-manipulation	Bimanual Lift; Door Open; Pick-Carry-Place; Crate Carry; Box Push	Balance-preserving whole-body interaction, articulated-object operation, bimanual object handling, object transport, and forceful manipulation during locomotion.

B Experimental Details

B.1 Benchmark Overview

We evaluate all methods on the same 16-task humanoid whole-body control benchmark as listed in Table 2, covering reference-free locomotion, reference-based locomotion, and loco-manipulation. Across algorithms, each task uses matched observations, action spaces, rewards, rollout budgets,

Table 3: Common policy inputs and outputs used by all evaluated algorithms.

Tasks	Actor and critic observation	Obs. dim.	Action dim.
Stand, Walk, Run, Step-Up	Root pose, root linear and angular velocity, and joint positions and velocities.	87	37
Turn-Jog, One-Foot Balance	Robot state; reference joint-position error and joint velocity; relative root displacement and orientation; reference root velocity, foot positions, and foot contacts.	182	37
Bimanual Lift, Pick-Carry-Place	Reference-tracking observation augmented with object pose and velocity, hand-target displacements, contact targets, and normalized motion stage.	217	37
Door Open	Reference-tracking and object-interaction observation, including the door configuration and hand–handle relation.	206	37
Normal Walk, March, Goose Step, Frog Jump, Zombie Hop	Reference joint positions and velocities, relative torso orientation, base angular velocity, joint-position offsets, joint velocities, and the previous action.	154	29
Crate Carry, Box Push	The 154-dimensional tracking observation augmented with root-frame object position, target displacement, velocity error, hand–object displacements, object-up direction, and contact targets (20 dims).	174	29

and environment configurations as detailed in Table 3. All algorithms use a shared actor-critic observation contract within each task. The actor and critic receive the same normalized observation, and the critic produces a scalar state-value estimate. No additional critic-only observation is used.

The 37-DoF tracking observation **input** contains the 44-dimensional generalized position, 43-dimensional generalized velocity, 37-dimensional reference joint-position error, 37-dimensional reference joint velocity, 3D relative root displacement, 4D relative root orientation, 6D reference root velocity, 6D reference foot positions, and two foot-contact indicators, totaling 182 dimensions for non-interaction tasks. Interaction tasks additionally include the object states and 22 object-, hand-, contact-, and stage-related features. The 29-DoF tracking observation **input** contains 29 reference joint positions, 29 reference joint velocities, a 6D relative torso orientation, 3D base angular velocity, 29 joint-position offsets, 29 joint velocities, and the previous 29D action, totaling 154 dimensions. The object-interaction tasks append 20 object and contact features.

Each policy **outputs** a normalized 37D or 29D joint-action vector, which is converted into joint-position targets for the PD controller. The 37-DoF simulation model actuates 12 leg joints, one waist joint, 10 arm/wrist joints, and 14 hand joints. The deployable 29-DoF model actuates 12 leg joints, three waist joints, and 14 arm/wrist joints, while leaving the hands unactuated.

The reference motions are drawn from BONES-SEED[1], the 100STYLE locomotion dataset Mason et al. (2022) (March, Zombie Hop, Box Push), and the ScaleBFM test set Zeng et al. (2026) (Goose Step). We use the released Unitree G1 trajectories from BONES-SEED and the G1-retargeted 100STYLE motions distributed through Any4HDMI[2].

B.2 Locomotion Tasks and Reward Functions

All locomotion tasks use the following bounded per-step reward template:

$$r_t = w_{\text{task}} r_t^{\text{task}} + w_{\text{track}} r_t^{\text{track}} + w_{\text{stable}} r_t^{\text{stable}} - \sum_k \lambda_k c_{k,t}. \tag{32}$$

The implementation logs each component separately before weighting. Regularizers penalize instantaneous violations to prevent their magnitude from growing with elapsed episode time. Large

[1] https://huggingface.co/datasets/bones-studio/seed

[2] https://huggingface.co/datasets/elijahgalahad/any4hdmi-g1-100style

penalties and termination are reserved for clear failures such as falling, unrecoverable states, invalid contacts, or extreme actuation.

- **Command-conditioned balance and locomotion.** Stand, Walk, and Run use root-state or velocity objectives together with support, uprightness, bounded actuation regularization, task-specific tolerances, and success-hold durations.
- **Transition, asymmetric-support, and terrain tasks.** Turn-Jog and One-Foot Balance combine task progress with motion or pose tracking. Step-Up uses forward velocity, vertical progress, swing-foot clearance, valid stair contact, stable support, and a 3 cm stair geometry.
- **Motion-conditioned locomotion.** Normal Walk, March, Goose Step, Frog Jump, and Zombie Hop share a reference-tracking formulation based on local-coordinate root tracking, joint and link tracking, reference-state initialization, phase sampling, reference frame rate, and bounded stability regularization.

B.3 Manipulation and Loco-Manipulation Rewards

All loco-manipulation tasks use the decomposition

$$r_t = w_{\text{motion}} r_t^{\text{motion}} + w_{\text{interaction}} r_t^{\text{interaction}} + w_{\text{success}} r_t^{\text{success}} + w_{\text{stable}} r_t^{\text{stable}} - \sum_k \lambda_k c_{k,t}. \tag{33}$$

Motion tracking supplies a reachable whole-body prior, while interaction terms become informative through physical contact, object motion, or articulated-object progress. The completion term corresponds to the physical task outcome and carries the largest task-level value.

- **Bimanual Lift.** The task specifies object geometry and mass, the initial hand-object layout, valid bilateral interaction, lift-height and hold-time criteria, non-hand contact handling, and the success condition.
- **Door Open.** Door articulation precedes a motion-free locomotion segment after the doorway becomes traversable. The task specifies hinge limits and damping, the opening threshold, target region, stable hold requirement, and fall termination.
- **Pick-Carry-Place.** The task combines a motion-supported pick segment, a lift-and-hold condition, carry displacement, a release or placement condition, valid support contacts, and physical-state completion predicates.
- **Crate Carry.** The task uses object-relative observations, hindsight-adjusted hand targets, a bilateral retention condition, a locomotion target, object-ground clearance, and a physical success criterion.
- **Box Push.** The task uses projected hand targets, valid hand contact, a non-hand contact penalty, continuous object displacement or goal progress, and a physical success criterion. The box moves only through simulated physical interaction.

We further introduce motion auditing that verifies collision-free placement, reachable interaction points, sustained contact corridors, physically driven object motion, and phase-wise RSI validity.

B.4 Baseline Details

- **Gaussian PPO.** (Schulman et al., 2017) A diagonal-Gaussian MLP policy trained with the standard clipped PPO objective; it provides a computationally lightweight unimodal reference.
- **FPO.** (McAllister et al., 2026) A flow policy optimized online through a conditional flow-matching surrogate for the policy gradient.
- **FPO++.** (Yi et al., 2026) The strengthened FPO implementation using per-sample ratios and asymmetric clipping to improve online flow-policy stability.

- **GenPO.** (Ding et al., 2025) A generative-policy optimizer that reconstructs likelihood ratios through inversion of the generative process.
- **PolicyFlow.** (Yang et al., 2026) A flow-policy optimizer that regularizes velocity-field variation and mode collapse during online policy improvement.

All methods use matched observation and action interfaces, environment transitions, rollout batches, critic-update protocols, initialization, rewards, and checkpoint schedules. Each actor follows the objective and update schedule defined by its algorithm.

B.5 Reference Processing, Motion Auditing, and Sampling

Before training, each reference motion is converted into a fixed trajectory containing joint and root states, body and foot positions, contact labels, and semantic motion phases. For interaction tasks, event boundaries and contact intervals are inferred from the complete demonstration and used to construct hindsight-adjusted hand and object targets consistent with the scene geometry. This offline processing may access future reference frames, while its outputs remain fixed throughout training and evaluation; no future simulated state or policy outcome is exposed to the agent. All algorithms use the same processed references and target trajectories.

Candidate reset states are then checked for finite values, valid joint configurations, trajectory continuity, and contact compatibility. Tasks requiring dynamic auditing additionally perform a short closed-loop replay from each candidate phase to exclude physically infeasible or unstable initial states. Reference State Initialization (RSI) samples only from frames that pass these checks. Unless specified otherwise, an episode starts from the first reference frame with probability 0.6; otherwise, a semantic phase is sampled uniformly, followed by uniform sampling among its valid frames. This phase-stratified distribution prevents long motion segments from dominating the reset distribution. Box Push always starts from the first frame, while reference-free tasks use their task-specific initial-state distributions.

B.6 Statistical Protocol and Metrics

The benchmark contains 16 tasks and every run uses 2,048 parallel environments, a 24-step rollout, and 3,000 policy updates. All algorithms use five seeds (40–44) per task. Learning-curve statistics use complete 3,000-update trajectories, including runs that exhibit late collapse.

For task t, algorithm alg, seed seed, and update u, the normalized training reward is

$$Z_{t,\text{alg,seed}}(u) = \frac{R_{t,\text{alg,seed}}(u) - L_t}{U_t - L_t}. \tag{34}$$

The lower anchor L_t is the equal-algorithm average of the five baseline seed means over the first 100 updates. The upper anchor U_t is the largest 200-update moving average attained by any baseline seed mean between updates 200 and 3,000. FFPO is excluded from both anchors, and normalized scores are not clipped. Normalized AUC is the trapezoidal integral of Z over updates 1–3,000 divided by 2,999. Final performance is the mean over the last 200 updates, peak performance is the best 200-update moving average, and drawdown is peak minus final performance. Task-balanced means assign equal weight to every task and equal weight to seeds within each task. The interquartile mean (IQM) is the fractionally trimmed mean of the central 50% of this weighted task-seed distribution. Performance profiles measure the fraction of task-seed scores strictly exceeding thresholds from -1.0 to 1.4 in increments of 0.02.

Uncertainty is estimated with 2,000 stratified bootstrap replicates. Complete seed trajectories are resampled within each task-algorithm stratum while the task set and normalization anchors remain fixed; percentile 95% confidence intervals are then computed from the bootstrap distribution. AUC uses the complete trajectory, and final-performance comparisons use the last 200 updates. Reward-best checkpoints are used only for explicitly labelled diagnostic evaluations or video export.

Table 4: Hyperparameter and optimization settings.

Component	Value
Parallel environments	2,048
Rollout length	24 steps
Flow matching model	8 steps for rollout and training; 4 for deployment
Prefix estimator	First-order flow estimator; 4-step RK4
Optimizer	AdamW, learning rate 10^{-4}, weight decay 10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$
ASPO clipping	$\epsilon_c = 0.05$
Actor update allocation	64 total body epochs, 4 minimum per body, 4 minibatches per epoch
Epoch allocation	EMA decay 0.9; temperature $T = 2$
Critic optimization	32 epochs, 4 minibatches per epoch
Return estimation	$\gamma = 0.99$, GAE $\lambda = 0.95$
Step control	Target RMS log-ratio $\rho_\star$; $\varsigma_{\min} = 0$; task-selected $\varsigma_{\max}$, see Table 5
Gradient Norm Clip	1
Other clip range	$B_{\tan} = 20, B_{\text{pre}} = 5, B_{\text{CFM}} = 10$
Actor network	MLP$[d_{obs} + 8 + d_i, 148, 148, d_i]$
Centralized Critic network	MLP$[d_{obs}, 768, 768, 768, 1]$

Table 5: $(\rho_\star, \varsigma_{\max})$ choices for different tasks.

Task	$\rho_\star$	$\varsigma_{\max}$
Stand	0.15	4
Walk	0.15	4
Run	0.15	4
Step-Up	0.5	4
Turn-Jog	0.3	5
One-Foot Balance	0.3	5
Bimanual Lift	0.3	5
Door Open	0.15	3
Pick-Carry-Place	0.15	3
Normal Walk	0.5	5
March	0.3	5
Goose Step	0.4	5
Frog Jump	0.3	5
Zombie Hop	0.5	5
Crate Carry	0.2	4
Box Push	0.2	4

C FFPO Implementation Details

C.1 Network Architecture and Hyperparameters

FFPO uses one conditionally independent flow actor for each of four functional body groups and one centralized value function. Each actor receives the normalized policy observation, an 8-dimensional flow-time embedding, and its body-local noisy action. Its exact-divergence velocity field is an MLP with two 148-unit ELU hidden layers and a body-local output. The centralized critic uses three 768-unit ELU hidden layers. These widths and body partitions are architectural choices and are independent of the FFPO objective. Concrete hyperparameter and optimization settings are listed in Table 4. The task-specific $(\rho_\star, \varsigma_{\max})$ values are listed in Table 5.

C.2 Overall Training Algorithm

The procedure of FFPO is listed in Algorithm 1, calling the tangent estimator in Algorithm 2. Four actors are updated for 64 epochs in total, which are distributed based on the SNR and the minimum body epochs calculated via $E_i = E_{\min} + \lfloor (E_{\text{total}} - M \cdot E_{\min}) w_i \rfloor$. The remaining $E_{\text{total}} - \sum_i E_i$ epochs are uniformly randomly distributed to each body agent. During optimization, the current body uses an ordinary differentiable CFM pass, whereas each candidate preceding body contributes

Algorithm 1 FFPO training

Require: Body actors $\{\pi_{\theta_i}\}_{i=1}^{M}$, centralized critic V_ψ, total actor epochs E_{total}, stored SNR statistics $\{\bar{c}_i\}$, target RMS log-ratio $\rho_\star$, and scale bounds $[\varsigma_{\min}, \varsigma_{\max}]$

for policy update $k = 1, \dots, K$ **do**
  Collect a rollout with all body actors evaluated in parallel; store actions, rewards, values, and old-policy flow quantities
  Compute returns and normalized generalized advantages $\widehat{A}$
  Set $w_i \propto \exp(\log(\bar{c}_i + \varepsilon)/T)$ and integer-allocate $\{E_i\}_{i=1}^{M}$ subject to $\sum_i E_i = E_{\text{total}}$
  Sample a body-update permutation σ; save actor anchors $\theta_i^{\text{old}} \leftarrow \theta_i$
  Initialize detached local-loss and log-ratio caches
  for $m = 1, \dots, M$ **do**
    $i \leftarrow \sigma_m$ and $H_{m-1} \leftarrow \sum_{r<m} \text{clip}(\text{stopgrad}(\widehat{\ell}^{\tan}_{\sigma_r}), -B_{\tan}, B_{\tan})$ `# computed through Algorithm 2`
    for $e = 1, \dots, E_i$ and each actor minibatch $\mathcal{B}$ **do**
      $\tilde{\ell}_i^{(q)} \leftarrow \text{clip}\left(\mathcal{L}^{\text{old}}_{\text{CFM},i} - \mathcal{L}_{\text{CFM},i}(\theta_i), -B_{\text{CFM}}, B_{\text{CFM}}\right)$
      $\hat{R}_{<m} \leftarrow \exp(\text{clip}(H_{m-1}, -B_{\text{pre}}, B_{\text{pre}})), \quad \tilde{r}_i^{(q)} \leftarrow \exp(\tilde{\ell}_i^{(q)})$
      $\mathcal{L}_i \leftarrow -\mathbb{E}_{\mathcal{B}}\left[\Phi_{\text{ASPO}}(\tilde{r}_i^{(q)}, \tilde{A}_i)\right]$
      Apply one finite, gradient-clipped AdamW candidate update to θ_i^{cand}
      Accumulate the first four minibatch gradients for the body-wise SNR estimate c_i
    end for
    Update $\bar{c}_i \leftarrow \beta \bar{c}_i + (1-\beta) c_i$
    Current body i: compute $\widehat{\ell}_i^{\tan}$ from the old-policy flow and the current velocity-field perturbation; update both caches
  end for
  Update V_ψ on the stored returns with finite, gradient-clipped AdamW steps
  Form candidate displacements $\Delta\theta_i \leftarrow \theta_i^{\text{cand}} - \theta_i^{\text{old}}$
  $\widehat{\rho}_{\text{cand}} \leftarrow \sqrt{\mathbb{E}[(\sum_i \widehat{\ell}_i)^2]}, \quad \varsigma \leftarrow \text{clip}\left(\rho_\star/(\widehat{\rho}_{\text{cand}} + \varepsilon), \varsigma_{\min}, \varsigma_{\max}\right)$
  Commit $\theta_i^{\text{new}} \leftarrow \theta_i^{\text{old}} + \varsigma \Delta\theta_i$ for every body i
end for

Algorithm 2 Batched Tangent Ratio Estimation

Require: Action a, condition s, fields $v^{\text{old}}, v^{\text{cand}}$, grid $\{t_k\}_{k=0}^{K}$

$\mathcal{C} \leftarrow$ `cache_old_path`$(a, s, v^{\text{old}}, \{t_k\})$`.detach()`
$v_n^{\text{cand}} \leftarrow$ `batched_field`$(v^{\text{cand}}, \mathcal{C}.x, \mathcal{C}.t, s)$
$d_n^{\text{cand}} \leftarrow$ `batched_divergence`$(v^{\text{cand}}, \mathcal{C}.x, \mathcal{C}.t, s)$
$\Delta v_n \leftarrow v_n^{\text{cand}} - \mathcal{C}.v, \qquad \Delta d_n \leftarrow d_n^{\text{cand}} - \mathcal{C}.d$
$u \leftarrow 0, \quad I \leftarrow 0$
for $k = 0, \dots, K-1$ **do**
  $(u, I) \leftarrow$ `RK4TangentStep`$(u, I; \mathcal{C}.J_k, \mathcal{C}.g_k, \Delta v_k, \Delta d_k)$
end for
$\widehat{\ell}^{\tan} \leftarrow \nabla_{x_K} \log p_1(\mathcal{C}.x_K)^\top u + I$
return $\text{clip}\left(\widehat{\ell}^{\tan}, -B_{\tan}, B_{\tan}\right)$ {batched node evaluation; no trajectory BPTT}

a detached first-order flow ratio that is cached and reused by later updates. The estimator uses forward sensitivity and local automatic differentiation and does not backpropagate through the integration trajectory. FFPO stores separate actor parameters and the local and endpoint ratio caches. In Algorithm 2, `RK4TangentStep` applies one standard fourth-order Runge–Kutta step to the coupled dynamics $\dot{u}_t = J_t^{\text{old}} u_t + \Delta v_t$ and $\dot{I}_t = \Delta d_t + (\nabla_x \text{div}\, v_t^{\text{old}})^\top u_t$.

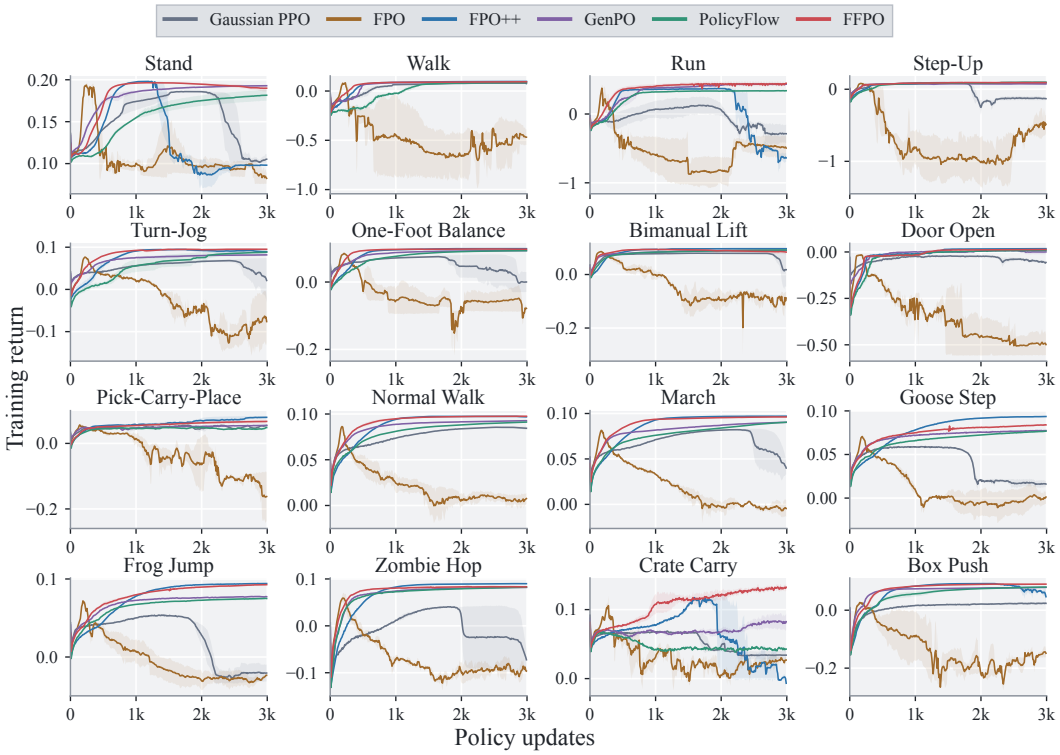


Figure 6: Training reward curves on all 16 tasks across 5 seeds. Each curve shows the mean and the shaded region denotes one standard deviation.

C.3 Training Details

Each rollout is flattened and divided into four shuffled minibatches. The body actors are optimized sequentially in a newly sampled order, with the integer epoch counts assigned before the first body update. The pre-clipping ASPO minibatch gradients are collected during the current body update. Every candidate body update refreshes the detached prefix cache used by later bodies. The SNR-EMA starts from the first valid body agent SNR, and is subsequently updated with an EMA decay of 0.9. The centralized critic is then optimized for 32 epochs over four minibatches, after which all actor parameters are committed by scaling their displacements from the pre-update anchors. Actor and critic gradients are required to be finite and are clipped to norm 1. The implementation uses no clipped value loss and non-finite minibatch is skipped and counted. We select the task-specific FFPO step-control parameters with a frozen two-stage protocol. First, $\rho_\star \in \{0.15, 0.20, 0.30, 0.40, 0.50\}$ is scanned with $\varsigma_{\max} = 4$. Second, $\varsigma_{\max} \in \{1, 2, 3, 4, 5\}$ is scanned at the selected $\rho_\star$. We select the configuration with the largest reward AUC.

C.4 Evaluation Protocol

Evaluation restores the actor, centralized observation normalizer, and task configuration from the specified checkpoint and disables optimizer updates. Unless stated otherwise, evaluation uses 128 parallel environments, 512 completed episodes, and at most 20,000 simulation steps. Actions follow the deterministic flow-policy inference path. Stochastic evaluation uses a separately seeded action generator and is explicitly labelled. Reference-state initialization follows the training distribution by default, while frame-zero initialization forms a separate condition. Converged-performance and stability analyses use the final checkpoint, while best-versus-final diagnostics include both checkpoints. Videos use the same policy, reset condition, and fixed evaluation seeds as the corresponding numerical evaluation.

Table 6: Normalized reward AUC on Stand–Door Open (part 1).

Algorithm	Stand	Walk	Run	Step-Up	Turn-Jog	One-Foot	Lift	Door Open
Gaussian PPO	0.486 ± 0.041	0.781 ± 0.020	0.169 ± 0.238	0.346 ± 0.003	0.550 ± 0.093	0.485 ± 0.269	0.700 ± 0.049	0.729 ± 0.005
FPO	-0.051 ± 0.129	-1.224 ± 1.182	-0.642 ± 0.197	-3.831 ± 1.191	-0.315 ± 0.124	-0.579 ± 0.352	-0.737 ± 0.299	-0.789 ± 0.421
FPO++	0.212 ± 0.045	0.875 ± 0.007	0.444 ± 0.061	0.836 ± 0.000	0.811 ± 0.002	0.860 ± 0.005	**0.887 ± 0.006**	**0.864 ± 0.010**
GenPO	0.801 ± 0.012	0.852 ± 0.012	0.821 ± 0.005	0.814 ± 0.000	0.750 ± 0.032	0.827 ± 0.009	0.834 ± 0.014	0.862 ± 0.021
PolicyFlow	0.523 ± 0.061	0.625 ± 0.045	0.702 ± 0.003	0.815 ± 0.093	0.575 ± 0.147	0.717 ± 0.010	0.803 ± 0.052	0.801 ± 0.027
FFPO	**0.825 ± 0.012**	**0.897 ± 0.003**	**0.886 ± 0.048**	**0.881 ± 0.082**	**0.895 ± 0.007**	**0.910 ± 0.002**	0.876 ± 0.041	0.854 ± 0.006

Normal Walk case study. For the analysis in Section 5.3, we train FPO++ and three FFPO variants (without step control, and with $\rho_\star = 0.15$ or 0.50) for 500 policy updates using five training seeds. At updates 100, 300, and 500, each checkpoint is evaluated from the first reference frame with 32 stochastic rollouts per seed. Figure 3(a) shows the allocated epochs and joint-position tracking errors for the legs and arms, while Figure 3(b) shows representative rollouts. For Figure 3(c), each behavior is recorded as present or absent in each rollout. *Foot support* requires at least two switches between left-only and right-only support, with each support bout lasting at least four control steps. *Root stability* requires the rollout to reach 100 control steps, with root-orientation error below 0.30 rad in at least 80 of those steps. *Progress* requires the reference phase to reach 50%, forward displacement to reach at least 50% of the reference displacement, and the running forward-velocity RMSE to remain at or below 0.35 m/s. The bars show the mean fraction of rollouts meeting each criterion across training seeds; error bars show one standard deviation.

C.5 Sim2Sim and Sim2Real Hardware Configuration

Cross-simulator validation. Before hardware deployment, each policy is validated through a Mujoco-Warp (Google DeepMind & NVIDIA, 2025) → Isaac Sim(NVIDIA, 2025) → MuJoCo(Todorov et al., 2012) chain. The same checkpoint, reference motion, observation normalization, 29-DoF joint ordering, action transformation, PD controller, and 50-Hz policy frequency are used across all backends. The actor observation contains the reference joint state and orientation, proprioceptive joint and IMU measurements, and the previous action. Reference phase is advanced by the policy-step clock and reset together with the previous action and initial yaw alignment. At matched states, we verify element-wise consistency of the packed observations and policy outputs; the maximum observation and PyTorch-to-ONNX action discrepancies are 1.21×10^{-7} and 4×10^{-6}, respectively.

Hardware configuration and safety. Hardware experiments use the 29-DoF Unitree G1 with a robot-local C++ controller. Policy inference runs at 50 Hz, while a 1-kHz low-level loop publishes joint-position targets through the Unitree `LowState`/`LowCmd` interface. The policy relies exclusively on onboard joint encoders, the IMU, a locally stored reference motion, and internal phase and action states; no external localization, motion capture, or camera input is used. The measured 99th-percentile policy period is 20.07 ms, and the 99th-percentile LowState age is 1.31 ms. Safety checks cover command validity, joint and action limits, target-rate and torque monitoring, torso tilt, root height, and communication timeouts. An invalid or stale command, excessive tilt, LowState timeout, or operator emergency stop transfers the controller to damping mode.

Hardware evaluation protocol. Each trial begins from a stabilized standing state and is initiated only after the robot posture, communication state, and test area have been checked. An operator remains ready to trigger hardware emergency damping throughout execution. We approve Normal Walk, March, and Goose Step for hardware evaluation and conduct four trials each; the remaining 13 benchmark tasks are evaluated in simulation. All hardware trials are recorded using a fixed lateral camera placed perpendicular to the nominal direction of motion. The camera is used only for evaluation and visualization.

Table 7: Normalized reward AUC on Pick Carry–Box Push (part 2).

Algorithm	Pick-Carry	Normal Walk	March	Goose Step	Frog Jump	Zombie Hop	Crate Carry	Box Push
Gaussian PPO	0.486 ± 0.037	0.634 ± 0.013	0.531 ± 0.089	0.142 ± 0.030	0.058 ± 0.087	0.403 ± 0.175	-0.052 ± 0.061	0.547 ± 0.017
FPO	-0.831 ± 0.069	-0.276 ± 0.174	-0.278 ± 0.026	-0.351 ± 0.140	-0.347 ± 0.181	0.022 ± 0.060	-0.442 ± 0.149	-0.254 ± 0.253
FPO++	**0.698 ± 0.093**	0.836 ± 0.003	0.830 ± 0.002	**0.755 ± 0.016**	0.754 ± 0.006	0.871 ± 0.004	0.132 ± 0.275	0.844 ± 0.019
GenPO	0.508 ± 0.012	0.806 ± 0.012	0.747 ± 0.006	0.630 ± 0.007	0.663 ± 0.005	0.866 ± 0.002	0.235 ± 0.054	0.858 ± 0.002
PolicyFlow	0.418 ± 0.010	0.703 ± 0.006	0.663 ± 0.002	0.547 ± 0.002	0.598 ± 0.009	0.856 ± 0.001	-0.162 ± 0.008	0.780 ± 0.030
FFPO	0.624 ± 0.048	**0.892 ± 0.012**	**0.875 ± 0.007**	0.692 ± 0.009	**0.794 ± 0.015**	**0.901 ± 0.011**	**0.889 ± 0.096**	**0.896 ± 0.013**

Table 8: Final performance on Stand–Door Open (part 1).

Algorithm	Stand	Walk	Run	Step-Up	Turn-Jog	One-Foot	Lift	Door Open
Gaussian PPO	-0.097 ± 0.072	0.937 ± 0.006	-0.247 ± 0.333	-0.345 ± 0.221	0.327 ± 0.626	-0.083 ± 0.548	0.323 ± 0.682	0.655 ± 0.023
FPO	-0.297 ± 0.166	-1.189 ± 0.667	-0.577 ± 0.396	-2.563 ± 1.342	-0.856 ± 0.979	-0.889 ± 0.339	-1.493 ± 0.402	-1.511 ± 0.435
FPO++	-0.158 ± 0.010	**1.000 ± 0.008**	-0.799 ± 0.111	0.941 ± 0.002	0.945 ± 0.105	**1.000 ± 0.001**	**1.000 ± 0.003**	**1.000 ± 0.001**
GenPO	**0.940 ± 0.013**	0.978 ± 0.021	0.997 ± 0.003	0.854 ± 0.002	0.855 ± 0.047	0.933 ± 0.003	0.917 ± 0.002	0.913 ± 0.085
PolicyFlow	0.805 ± 0.106	0.956 ± 0.001	0.835 ± 0.000	**1.000 ± 0.052**	0.924 ± 0.059	0.931 ± 0.003	0.937 ± 0.051	0.954 ± 0.023
FFPO	0.909 ± 0.043	0.986 ± 0.004	**1.006 ± 0.057**	0.962 ± 0.113	**1.004 ± 0.008**	0.982 ± 0.016	0.856 ± 0.120	0.977 ± 0.013

D Additional Results

D.1 Complete Per-Task Results

Figure 6 shows the complete reward curves during the learning of different tasks and methods. The results show the same conclusion in the main results, that Gaussian PPO, FPO and FPO++ struggle to balance between rapid policy improvement and reliable convergence, as witnessed by the degradation of reward on Stand, Run and Crate Carry. Meanwhile GenPO and PolicyFlow exhibit similar stability and relatively slower improvement compared to FFPO.

Tables 6–9 report the complete per-task training results. Scores are normalized with task-specific anchors constructed from the baseline runs, excluding FFPO; values outside $[0, 1]$ are retained so that collapsed runs remain visible. Each cell is the mean $\pm$ sample standard deviation across seeds. AUC integrates normalized reward over all 3,000 updates, whereas final performance averages the last 200 updates.

FFPO attains the highest normalized AUC on 12 of the 16 tasks. Its gains are concentrated in coordination-intensive locomotion and interaction tasks: it leads all baselines on Stand, Run, Turn-Jog, One-Foot Balance, Frog Jump, Zombie Hop, Crate Carry, and Box Push. FPO++ remains strongest on Bimanual Lift, Door Open, Pick-Carry-Place, and Goose Step.

The final-performance tables show a different aspect of the comparison. FPO++ can reach high terminal reward on several tasks despite lower AUC and severe late instability on others, while FFPO combines high final reward with consistently stronger learning curves. In particular, FFPO reaches a normalized final score of 1.316 ± 0.069 on Crate Carry, where FPO++ collapses to -0.911 ± 0.160. Conversely, FPO++ remains stronger on the three manipulation tasks and Goose Step.

D.2 Full Statistical Profiles

Figure 7 summarizes the task-balanced statistical profiles (Agarwal et al., 2021) over all runs. Normalized AUC is the integral of normalized training reward over 3,000 updates; final performance is the last-200-update mean; drawdown is the difference between the best 200-update rolling mean and the final score. Means and interquartile means (IQMs) give every task equal weight, and 95% confidence intervals use 2,000 stratified seed bootstraps with fixed tasks and normalization anchors.

Table 9: Final performance on Pick Carry–Box Push (part 2).

Algorithm	Pick-Carry	Normal Walk	March	Goose Step	Frog Jump	Zombie Hop	Crate Carry	Box Push
Gaussian PPO	0.544 ± 0.094	0.784 ± 0.032	0.206 ± 0.418	-0.274 ± 0.092	-0.562 ± 0.222	0.181 ± 0.450	-0.398 ± 0.152	0.619 ± 0.036
FPO	-2.519 ± 1.120	-0.553 ± 0.152	-0.679 ± 0.181	-0.536 ± 0.198	-0.659 ± 0.042	-0.157 ± 0.093	-0.536 ± 0.076	-0.386 ± 0.046
FPO++	**1.000 ± 0.094**	0.990 ± 0.015	**0.999 ± 0.002**	**1.000 ± 0.015**	**1.000 ± 0.002**	**1.000 ± 0.004**	-0.911 ± 0.160	0.842 ± 0.079
GenPO	0.598 ± 0.019	0.912 ± 0.010	0.885 ± 0.005	0.735 ± 0.013	0.772 ± 0.012	0.952 ± 0.003	0.450 ± 0.261	0.926 ± 0.001
PolicyFlow	0.441 ± 0.060	0.885 ± 0.009	0.883 ± 0.004	0.718 ± 0.010	0.738 ± 0.026	0.951 ± 0.000	-0.242 ± 0.024	0.931 ± 0.009
FFPO	0.795 ± 0.105	**0.997 ± 0.002**	0.986 ± 0.012	0.840 ± 0.017	0.979 ± 0.015	0.959 ± 0.015	**1.316 ± 0.069**	**0.988 ± 0.018**

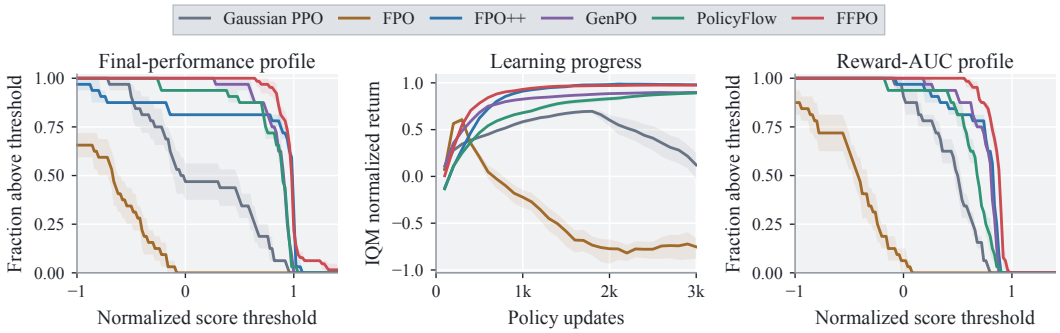


Figure 7: Task-balanced aggregate results. Panels report final performance profile, aggregated learning progress curve, reward-AUC profile; shaded regions denote 95% stratified-bootstrap confidence intervals.

Table 10: Gradient diagnostics aggregated on Stand, Turn-Jog, and Crate Carry.

Variant	Normalized AUC	Post-1000 drawdown	Gradient-SNR fraction < 0.1	Zero-SNR fraction
FPO++	0.348	1.219	24.80%	13.50%
Factorized actors	0.567	0.072	6.7%	0.00%
Full FFPO	0.831	0.132	15.80%	0.00%

D.3 The Effect of Body Factorization

We show how body decomposition affects the sample efficiency, gradient coherence and optimization instability via the AUC, gradient SNR and performance drop over three tasks Stand, Turn-Jog, and Crate Carry. As shown in Figure 8, factorized policies raise the AUC from 0.348 to 0.567 on average compared to FPO++, meanwhile the fraction of low-SNR updates and late instability are substantially reduced, consistent with Theorem 1, showcasing that factorization can directly lead to stabler optimization and higher sample efficiency. Noticeably, FFPO raises the fraction of low-SNR updates to a mildly higher level than Factorized under the control of both step rescaling and the SNR-based epoch allocation, which is hypothesized to increase early exploration and leads to the higher sample efficiency (0.831 AUC on average).

D.4 Gradient Coherence and Policy-Update Utilization

The gradient diagnostics in Table 10 echo the same mechanism in Figure 8, showing that replacing the monolithic policy with factorized body actors raises AUC and effectively reduces drawdown. Concretely, Factorized actors reduce the fraction of low-SNR updates from 24.8% to 6.7% and

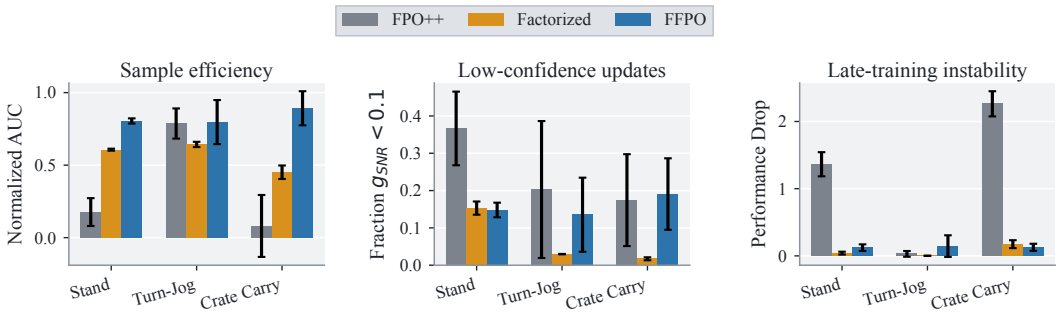
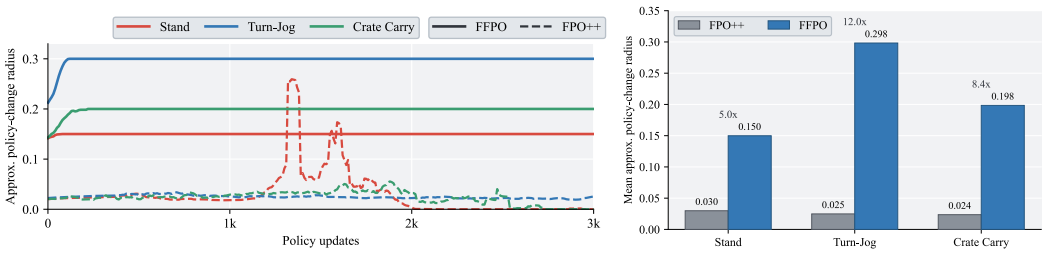


Figure 8: Evidence for update coherence under body factorization. Bars report five-seed means and one standard deviation.



(a) The curve of step sizes during training

(b) The average step sizes

Figure 9: The approximated committed policy update of FFPO and FPO++ on three tasks.

eliminate the 13.5% exact-zero-SNR events observed for FPO++. Full FFPO has 15.8% low-SNR updates and no exact-zero-SNR events while obtaining substantially higher normalized AUC than the factorized-only variant.

We further visualize the committed policy update of FFPO and FPO++ in Figure 9 to show the effectiveness of body factorization and step control within FFPO. The updates are calculated as:

$$\hat{\rho}_{\text{FPO++}} = \sqrt{\text{Var}(\tilde{r}) + (\mathbb{E}[\tilde{r}] - 1)^2}, \tilde{r} = \exp(\mathcal{L}_{\text{CFM}}^{\text{old}} - L_{\text{CFM}}^{\text{new}}) \tag{35}$$

$$\widehat{\rho}_{\text{FFPO}} = \sqrt{\frac{1}{N} \sum_{n} \left(\sum_{i} \varsigma \widehat{\ell}_{i,n} \right)^2}. \tag{36}$$

The results show that FFPO sustains larger and more consistent policy-space updates throughout training, ranging from 0.15 to 0.30 as selected for each task. In contrast, FPO++ generally operates at a smaller update radius of about 0.02 to 0.03, while exhibiting occasional abrupt changes followed by performance degradation, particularly on Stand. This behavior suggests that body factorization reduces cross-body interference and enables more aggressive yet controlled policy improvement. Noticeably, during the early updates of FFPO, the actualized step sizes are relatively lower since the rescaling coefficient ς is clipped by $\varsigma_{\max}$ as $\widehat{\rho}_{\text{FFPO}} = \varsigma_{\max} \widehat{\rho}_{\text{cand}} < \rho_\star$, which acts as a warm-start mechanism toward more stable training.

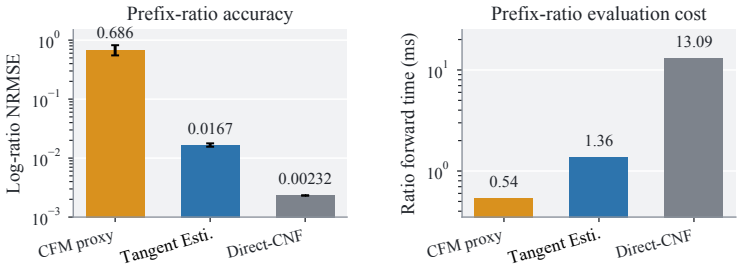


Figure 10: Per-agent prefix-ratio accuracy and forward evaluation cost. Each method estimates one candidate-to-old body ratio using cached old-policy quantities. The timing excludes parameter-gradient computation.

D.5 Ratio Estimation Accuracy and Cost

Sequential body updates impose asymmetric requirements on policy-ratio estimation. Previously updated bodies contribute a detached prefix ratio, whose numerical accuracy directly determines the credit propagated to the current body. In contrast, the current body requires an inexpensive differentiable surrogate that supplies a reliable first-order gradient. We compare the CFM proxy, Tangent-Estimator, and Direct-CNF against a high-accuracy endpoint-density reference in Figure 10. On the G1 calibration, their log-ratio NRMSE values are 0.686 ± 0.135, 0.0167 ± 0.0011, and 0.00232, respectively. Direct-CNF provides the highest numerical accuracy, but its actor computation takes

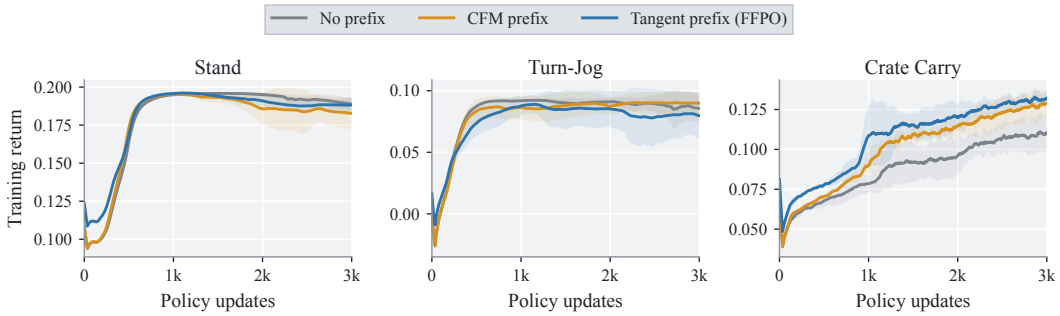


Figure 11: Effect of the detached prefix estimator on Stand, Turn-Jog, and Crate Carry. All variants use identical body decomposition, SNR allocation, policy-step control, and endpoint caching. Curves show the mean over 5 seeds with one-standard-deviation shading.

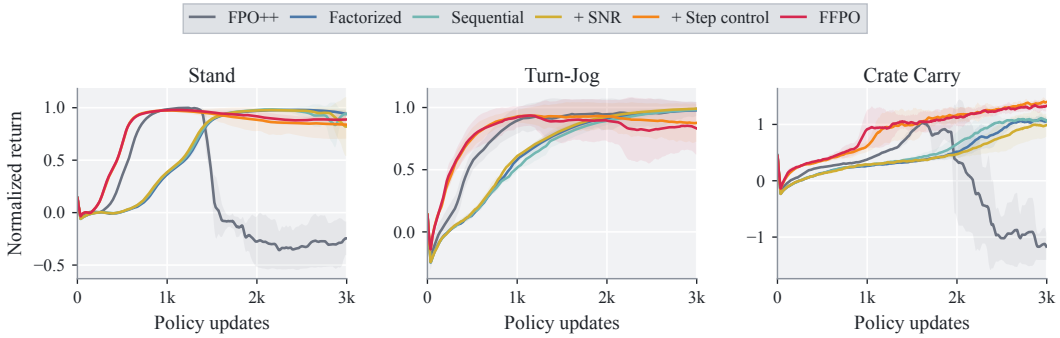


Figure 12: Component ladder on Stand, Turn-Jog, and Crate Carry. Curves show 5-seed means with one-standard-deviation shading. Sequential + SNR and Sequential + Step Control are parallel controls built on the sequential variant.

13.09 ms on the controlled benchmark, compared with 1.36 ms for Tangent-Estimator and 0.54 ms for the CFM proxy. Tangent-Estimator therefore retains a sufficiently accurate likelihood ratio at approximately 10% of the Direct-CNF cost.

Figure 11 examines whether this numerical difference affects sequential policy optimization. We retain the same body decomposition, SNR allocation, policy-step controller, and endpoint cache, and vary only the prefix supplied to subsequent body updates. Across the three tasks, the task-balanced normalized AUC increases from 0.736 without prefix correction to 0.791 with the CFM prefix and 0.831 with the Tangent-Estimator prefix. The paired Tangent-over-CFM difference is 0.040, with a 95% bootstrap interval of $[-0.017, 0.095]$, indicating a task-dependent aggregate effect. The improvement is most pronounced on Crate Carry, where Tangent-Estimator raises normalized AUC by 0.126 over the CFM prefix, and it recovers 0.039 on Stand, while Turn-Jog shows no measurable additional gain. Prefix accuracy is therefore most consequential when sustained coordination causes preceding body updates to contribute materially to the joint policy ratio.

These results motivate the asymmetric estimator design in FFPO. Tangent-Estimator is evaluated once after each candidate body update and cached as the detached prefix, providing accurate credit propagation without repeated likelihood differentiation. The current body instead uses the CFM surrogate, for which the first-order gradient direction is the relevant quantity and the substantially lower differentiation cost is advantageous. Using CFM for the detached prefix propagates its ratio error into the effective advantage and degrades performance on coordination-intensive tasks, whereas differentiating Direct-CNF throughout the sequential update incurs considerably higher computation. FFPO therefore uses the tangent estimator for accurate detached prefix correction and the differentiable CFM surrogate for current-body optimization.

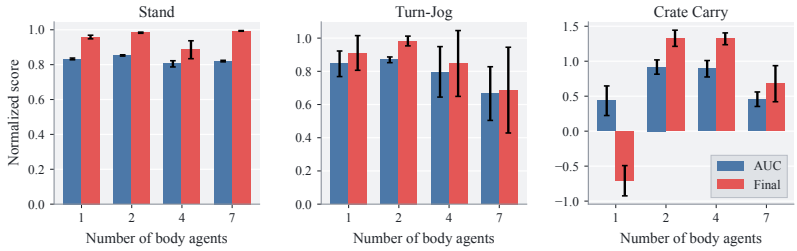


Figure 13: Sensitivity to the number of body agents on Stand, Turn-Jog, and Crate Carry. Bars report normalized AUC and final performance from five seeds per task.

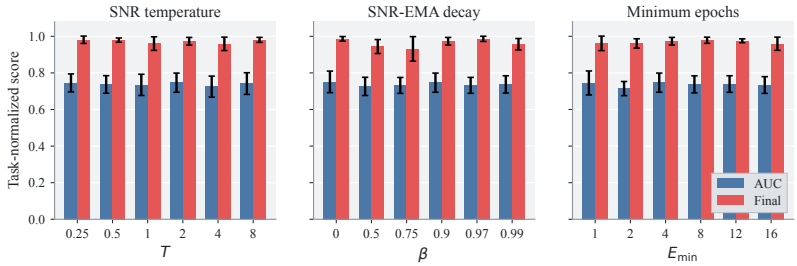


Figure 14: Sensitivity analysis on Temperature, SNR-EMA decay, minimum epochs.

D.6 Additional Ablations and Sensitivity

Figure 12 shows the complete component ladder on Stand, Turn-Jog, and Crate Carry which starts from monolithic FPO++, introduces factorized actors (Factorized), then sequential prefix correction (Sequential), and finally adds SNR-based epoch allocation (+SNR) and policy-step control (+Step control) before forming full FFPO.

Averaged over the three tasks, normalized AUC increases from 0.348 for FPO++ to 0.567 with factorized actors and 0.578 after sequential prefix correction. Adding policy-step control raises AUC to 0.830, while adding SNR allocation alone yields 0.561; the full method reaches 0.831. Thus, body factorization supplies the first coherence gain and policy-step control supplies the largest additional sample-efficiency gain on this subset. The near-zero difference between full FFPO and its no-SNR counterpart shows that SNR allocation is not a standalone aggregate gain here, although it remains useful for redistributing optimization effort across heterogeneous bodies.

Figure 13 varies the decomposition granularity while retaining the same task-balanced evaluation. Two body agents attain the highest mean AUC (0.880), followed by four agents (0.831), one agent (0.704), and seven agents (0.648). The result exhibits a non-monotonic granularity trade-off: two and four functional groups outperform the monolithic and seven-group variants on this three-task benchmark. We use the fixed four-group decomposition throughout the main study to retain functional body specialization and a consistent deployment interface.

Figure 14 reports the sensitivity of Temperature T, SNR-EMA decay β and minimum epochs $E_{\min}$, showing that the mean reward AUC and final reward mean are stable to these hyperparameters.

D.7 Full Deployment Results

We deploy FFPO policies on a Unitree G1 humanoid with 29 DoF using only proprioceptive observations and a 50 Hz control frequency, by first training for 3000 updates in simulation and finetuning for 500 updates in simulation with domain randomization and noisy observations. Then we deploy the final checkpoint on G1 hardware. We select three representative motions including Normal Walk, March and Goose Step as shown in Figure 5, 15 and 16. FFPO maintains low tracking errors and stably completes the three motions.

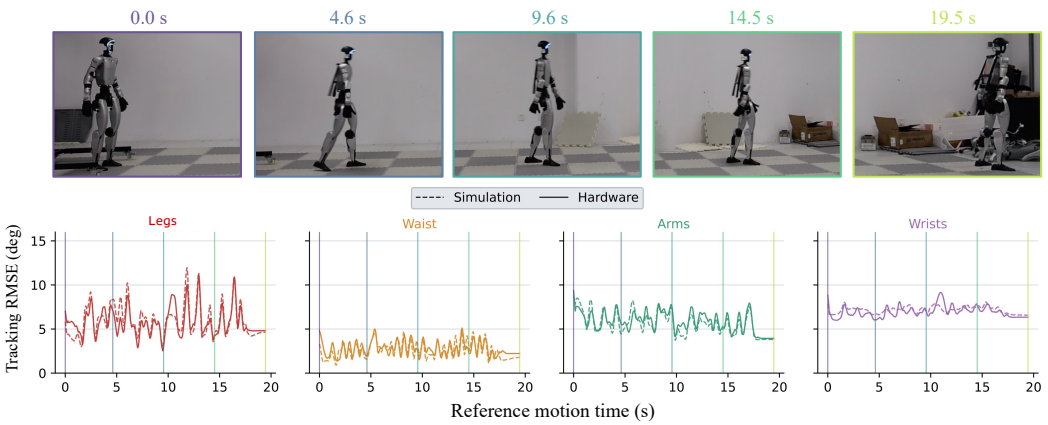


Figure 15: Real deployment on Unitree G1 hardware. Motion: March.

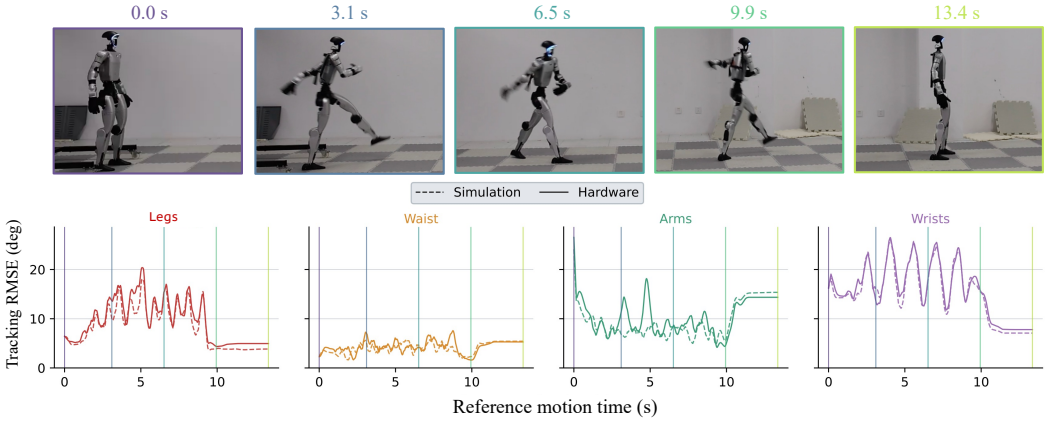


Figure 16: Real deployment on Unitree G1 hardware. Motion: Goose Step.

D.8 Limitations and Future Work

FFPO is designed around a functional decomposition of a single humanoid body. The present study evaluates one primary robot morphology and a small set of manually specified partitions. Broader claims require robots with different kinematic trees, actuation limits, hands, and morphology-dependent couplings. Future work should learn or adapt the decomposition from morphology and task context while preserving interpretable credit assignment.

The interaction suite covers lifting, carrying, pushing, door articulation, and placement, but does not yet represent long-horizon dexterous manipulation, deformable objects, rich tool use, or sustained human-robot contact. These settings introduce contact discontinuities and semantic transitions that may challenge reference-state initialization and local flow-ratio approximations. Physically audited interaction trajectories and dexterous hands are important next steps.

FFPO improves low-level online policy optimization under a supplied task observation, command, or reference motion. It does not provide open-ended task decomposition, semantic goal interpretation, or long-horizon planning. A promising direction is to combine FFPO with a language-conditioned planner or LLM/VLM policy that proposes subgoals, reference segments, or interaction constraints, while FFPO learns coordinated execution and supplies feasibility feedback.