BFM-Zero: A Promptable Behavioral Foundation Model for Humanoid Control Using Unsupervised Reinforcement Learning

Yitang Li[1, a, *] Zhengyi Luo[1, *] Tonghe Zhang[1, $] Cunxi Dai[1, $] Andrea Tirinzoni[2] Anssi Kanervisto[2]
Haoyang Weng[1, a] Kris Kitani[1] Mateusz Guzek[2] Ahmed Touati[2] Alessandro Lazaric[2]
Matteo Pirotta[2, †] Guanya Shi[1, †]

[*]Equal Contribution. [$]Equal Contribution. [†]Equal advising.
[1]Carnegie Mellon University [2]Meta

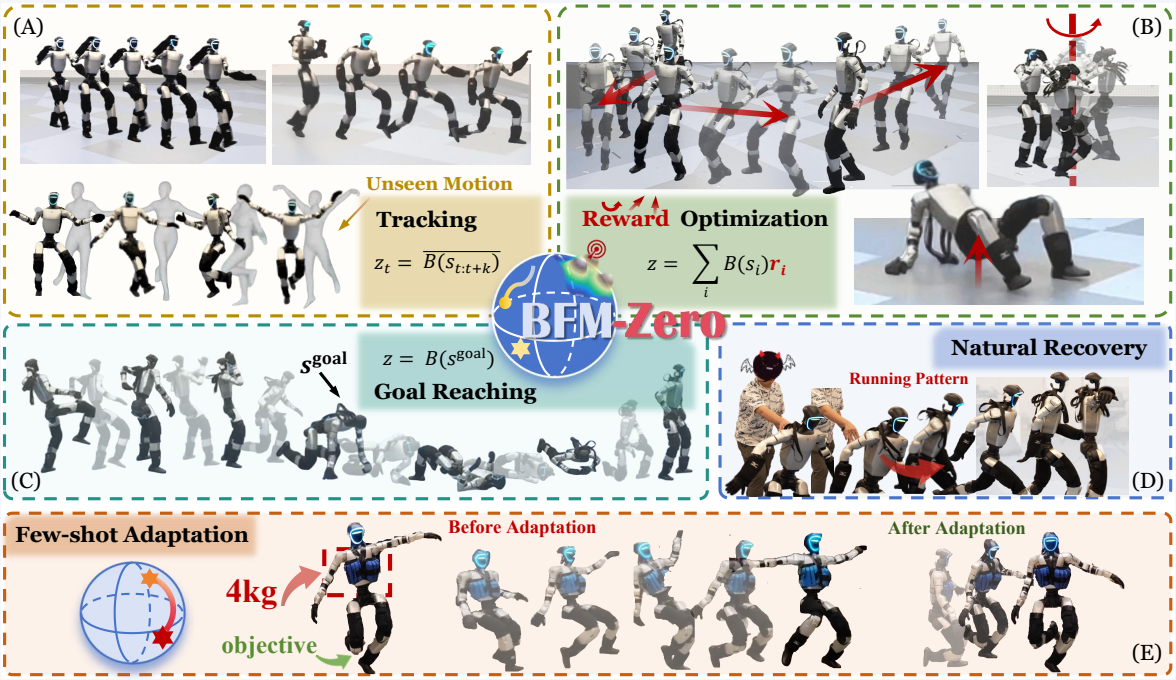


Figure 1: BFM-Zero enables versatile and robust whole-body skills. (A-C) Diverse zero-shot inference methods. (D) Natural recovery from large perturbation. (E) Few-shot adaptation.

[a]Work done during internships at Carnegie Mellon University. The authors are now with Tsinghua University.

Website: https://lecar-lab.github.io/BFM-Zero/

Abstract

Building Behavioral Foundation Models (BFMs) for humanoid robots has the potential to unify diverse control tasks under a single, promptable generalist policy. However, existing approaches are either exclusively deployed on simulated humanoid characters, or specialized to specific tasks such

as tracking. We propose BFM-Zero, a framework that learns an effective shared latent representation that embeds motions, goals, and rewards into a common space, enabling a single policy to be prompted for multiple downstream tasks without retraining. This well-structured latent space in BFM-Zero enables versatile and robust whole-body skills on a Unitree G1 humanoid in the real world, via diverse inference methods, including zero-shot motion tracking, goal reaching, and reward inference, and few-shot optimization-based adaptation. Unlike prior on-policy reinforcement learning (RL) frameworks, BFM-Zero builds upon recent advancements in unsupervised RL and Forward-Backward (FB) models, which offer an objective-centric, explainable, and smooth latent representation of whole-body motions. We further extend BFM-Zero with critical reward shaping, domain randomization, and history-dependent asymmetric learning to bridge the sim-to-real gap. Those key design choices are quantitatively ablated in simulation. A first-of-its-kind model, BFM-Zero establishes a step toward scalable, promptable behavioral foundation models for whole-body humanoid control.

1 Introduction

Humanoid robots have the potential to transform numerous aspects of our daily lives, from manufacturing and logistics to healthcare and personal assistance. However, realizing this potential requires robots to perform a wide range of tasks in dynamic and unstructured environments. Humanoid whole-body control is a fundamental and challenging problem in robotics, serving as the first step to enable the humanoids to work safely in human environments [Gu et al., 2025].

In robotics, foundation models have the potential to unify diverse control objectives under a single policy, allowing robots to adapt to new tasks in a zero-shot[1]way or with efficient post-training. The closest approaches to such paradigms are Vision-Language-Action (VLA) models for robotic manipulations [e.g., Ghosh et al., 2024, Intelligence et al., 2025, Kim et al., 2024, Zhong et al., 2025, Team et al., 2025, Bjorck et al., 2025] that learn from human demonstrations (i.e., behavior cloning). However, for humanoid whole-body control, there is a fundamental mismatch that limits direct behavior cloning: unlike manipulation tasks, there are no readily available actuator-level action labels or large-scale teleoperation datasets.

For whole-body humanoid control, most recent advancements follow the sim-to-real pipeline and rely on reinforcement learning (RL) to train policies in simulation before transferring them to hardware [Gu et al., 2025]. Following the success of RL-based motion tracking in physics-based character animation [e.g., Luo et al., 2024, Tessler et al., 2024, Tirinzoni et al., 2025], recent works [e.g., Zakka et al., 2025, Seo et al., 2025, Chen et al., 2025, Liao et al., 2025, He et al., 2025a, Cheng et al., 2024, He et al., 2025b] have shown remarkable results in transferring policies trained in simulation to real robots. However, most of these approaches rely on *on-policy policy gradient* methods (e.g., PPO [Schulman et al., 2017]) with *explicit tracking-based rewards* and suffer from major limitations. First, they remain task-specific: most policies are trained to explicitly imitate motion capture clips or solve a single task. Second, they are non-adaptive: once trained, policies cannot be easily fine-tuned or composed for new tasks.

[1] *Zero-shot* means that, after pre-training, the policy can be directly deployed in the real world without further interacting with either simulated or real environments. In contrast, *few-shot* means the policy needs to interact with the environment to collect new data in few episodes to improve on certain tasks.

Third, they lack a unified and explainable interface for goal specification and behavior composition, making it difficult for human operators to direct the robot or combine learned skills into new behaviors.

In this work, we investigate whether *off-policy unsupervised* RL can be a suitable approach to train so-called Behavioral Foundation Models (BFMs) for whole-body control of a humanoid robot, enabling it to solve a wide range of downstream tasks specified by rewards, goals, or demonstrations without retraining. For tasks that require retraining, the BFM should enable efficient post-training. This conjecture is far from trivial. First, most existing methods with real-world deployment rely on on-policy training (primarily PPO), and there is little evidence that off-policy learning—commonly used in unsupervised RL for training multi-task policies—is well suited to this context. Second, no evidence exists that unsupervised RL algorithms can handle the sim-to-real gap and dynamic disturbances robustly, either during simulation policy training or at real-world inference.

We develop BFM-Zero[2], an online off-policy unsupervised RL algorithm that leverages motion capture data to regularize the process of learning generalist whole-body control policies towards *human behaviors*. We introduce domain randomization to address the sim-to-real gap and train robust policies via asymmetric history-dependent training, leveraging the privileged information available in simulation. Additionally, we incorporate auxiliary rewards to ensure that the learned behaviors adhere to the safety and operational constraints of the physical robot. To the best of our knowledge, the resulting algorithm allows us to train the *first behavioral foundation model* for real humanoids that can be prompted for different tasks (e.g., reward optimization, pose reaching, and motion tracking) without retraining (i.e., in zero-shot). Such a flexible and ready-to-use model, paves the way to fast adaptation, fine-tuning or even high-level planning. We validate our approach in both simulated environments and on a real Unitree G1 humanoid (Fig. 1 for examples), demonstrating robust generalization across tasks and conditions, and showing that even when the zero-shot policy is not satisfactory, we can effectively improve it within a few episodes of environment interaction. The discussion of related work is available in Section A.

2 BFM-Zero for Humanoid Whole-body Control

In this section, we outline the pipeline for training BFM-Zero in simulation and transferring it to real humanoids. Unlike for virtual characters [e.g., Peng et al., 2022, Tessler et al., 2023, Tirinzoni et al., 2025], applying unsupervised RL to real humanoids has not yet been attempted. Our BFM-Zero framework consists of an unsupervised pre-training stage, a zero-shot inference procedure, and possibly a fast-adaptation post-training stage (as shown in Fig. 2). Section 2.1 provides an overview of unsupervised RL using the forward-backward representation framework adopted by BFM-Zero. Section 2.2 details BFM-Zero pre-training, whose objective is to learn *a unified latent representation* that embeds tasks (e.g., target motions, rewards, goals) into a shared space $Z \subseteq \mathbb{R}^d$ and *a promptable policy* that conditions on this representation to perform diverse behaviors without task-specific retraining. Then, for downstream tasks during inference (Section 2.2), we embed the task into the latent space and use the policy to execute the task in a zero-shot manner. We also show that we can efficiently adapt the zero-shot policy in the latent space Z to improve performance on unseen tasks that are not easily covered by zero-shot inference via sampling-based optimization.

[2] Zero comes from its zero-shot inference capability via unsupervised RL and it is a first-of-its-kind model.

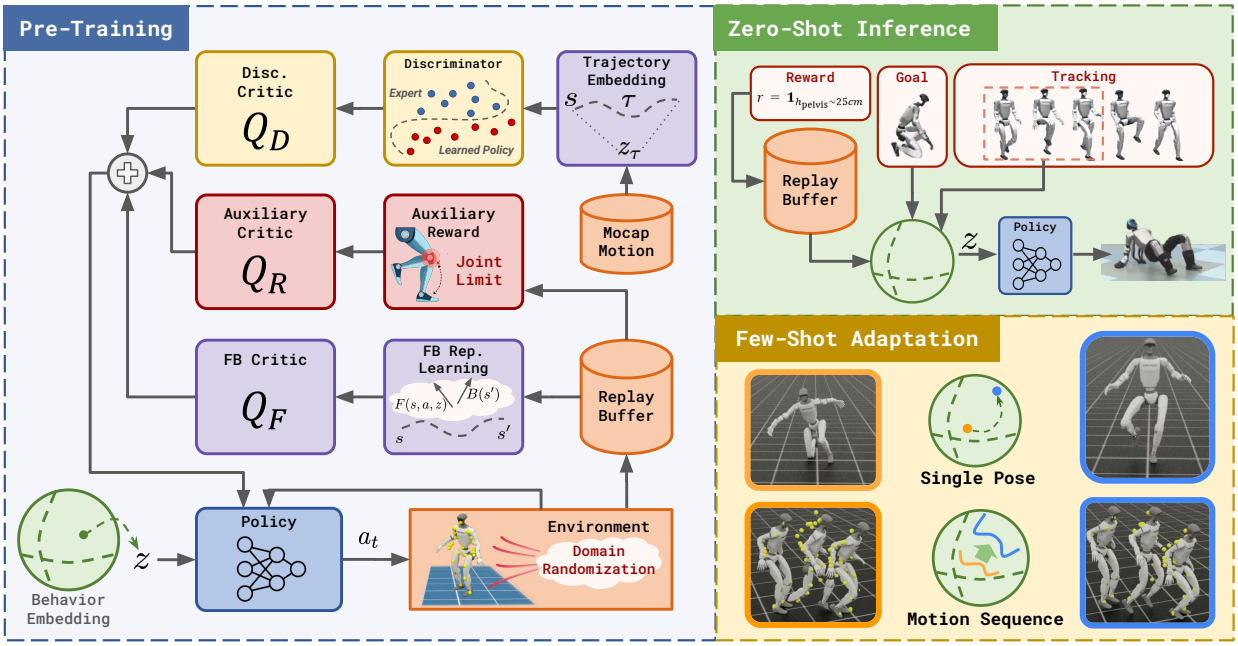


Figure 2: An overview of the BFM-Zero framework. After the pre-training stage, BFM-Zero forms a latent space that can be used for zero-shot reward optimization, single-frame goal reaching, and tracking. It can also be adapted in a few-shot fashion to reach more challenging poses.

Problem formulation. We formulate real-world humanoid control as a partially observable Markov decision process (POMDP) defined by the tuple (S, O, A, P, γ), where S is the full state space, O is the observation space, A is the action space, $P(s_{t+1}|s_t, a_t)$ is the transition dynamics, and $\gamma \in (0,1)$ is the discount factor. For the 29-degree-of-freedom (DoF) humanoid, the action $a \in A \subset \mathbb{R}^{29}$ contains the proportional derivative (PD) controller targets for all DoFs. The privileged information ($s \in \mathbb{R}^{463}$) consists of root height, body pose, body rotation, and linear and angular velocities. The observable state $o_t = \{q_t - \bar{q}, \dot{q}_t, \omega_t^{\text{root}}/4, g_t\} \in \mathbb{R}^{64}$ is defined as joint position $q_t \in \mathbb{R}^{29}$ normalized w.r.t. the nominal position $\bar{q}$, joint velocity $\dot{q}_t \in \mathbb{R}^{29}$, root angular velocity $\omega_t^{\text{root}} \in \mathbb{R}^3$ and root projected gravity $g_t \in \mathbb{R}^3$. We denote by $o_{t,H} = \{o_{t-H}, a_{t-H}, \ldots, o_t\} \in \mathbb{R}^{93 \cdot H + 64}$ the observable history composed by proprioceptive state and action. All the components of the states (except root height) are normalized w.r.t. the current facing direction and root position. At pre-trainig, we assume that the agent has access to a dataset of unlabeled motions $\mathcal{M} = \{\tau\}$, which contains observation and privileged states trajectories *i.e* $\tau = (o_1, s_1, \ldots, o_{l(\tau)}, s_{l(\tau)})$.

2.1 Unsupervised RL with Forward-Backward Representations

During the pretraining phase, BFM-Zero learns a compact representation of the environment by observing online reward-free interactions in the simulator and leveraging an offline dataset of unlabeled behaviors, resulting in a model that can be prompted to tackle a wide range of downstream tasks (e.g., tracking or reward maximization) in a zero-shot manner. To achieve this, we build on top of the recent FB-CPR algorithm [Tirinzoni et al., 2025] which combines the Forward-Backward (FB) method for zero-

shot RL [Touati and Ollivier, 2021] with online training and policy regularization on motion-capture data. This method falls in the broader category of unsupervised RL based on successor features [e.g., Touati and Ollivier, 2021, Touati et al., 2023, Pirotta et al., 2024, Park et al., 2024, Agarwal et al., 2024], which involves three components: (i) a latent task feature $\phi : S \to \mathbb{R}^d$ that embeds observation $s \in S$ into a d-dimensional vector, (ii) a policy $\pi_z : S \to A$ conditioned on a latent vector $z \in \mathbb{R}^d$, and (iii) latent-conditioned successor features [Barreto et al., 2017] F_z that encode the expected discounted sum of latent task features under the corresponding policy π_z, i.e, $F_z \simeq \mathbb{E}[\sum_t \gamma^t \phi(s_t) \mid \pi_z]$. We now explain how FB-CPR trains those components.

FB representations and FB-CPR. Among the different unsupervised RL approaches, forward-backward (FB) representations provide a principled unsupervised training objective for jointly learning latent task representations and their associated successor features. At a high level, FB learns a finite-rank approximation of long-term policy dynamics, where $\boldsymbol{B}$ captures the low-frequency features that best summarize the long-range temporal dependencies between states. Formally, given a training state distribution ρ, the FB framework learns two mappings: a forward mapping $\boldsymbol{F} : S \times A \times \mathbb{R}^d \to \mathbb{R}^d$ and a backward mapping $\boldsymbol{B} : S \to \mathbb{R}^d$ such that the long-term transition dynamics induced by the policy π_z decompose as:

$$M^{\pi_z}(\mathrm{d}s' \mid s, a) \simeq \boldsymbol{F}(s, a, z)^\top \boldsymbol{B}(s')\rho(\mathrm{d}s') \tag{2.1}$$

where for any region $X \subset S$ of the state space, $M^{\pi_z}(s' \in X \mid s, a) := \sum_t \gamma^t \Pr(s_t \in X \mid s, a, \pi_z)$ denotes the discounted visitation probabilities of reaching X under the policy π_z, starting from the state-action pair (s, a). Eq. 2.1 implies that $\boldsymbol{F}$ is the successor features of $\phi(s) := (\mathbb{E}_\rho[\boldsymbol{B}(s)\boldsymbol{B}(s)^\top])^{-1}\boldsymbol{B}(s)$ [Touati et al., 2023]. The learned representation ϕ defines a latent task space by inducing a family of linear reward functions of the form, i.e., $r_z(s) = \phi(s)^\top z$, In particular, each policy π_z is optimized to maximize $\mathbb{E}_\rho[\sum_t \gamma^t \phi(s_t)^\top z \mid \pi_z] = \boldsymbol{F}(s, a, z)^\top z$, i.e., $\boldsymbol{F}(s, a, z)^\top z$ is a Q-value function of π_z with reward $r = \phi^\top z$. Intuitively, $z \in Z$ defines a *task-centric* latent space associated with the task feature ϕ, where for each z, the corresponding π_z optimizes the linear combination of ϕ, $r_z = \phi^\top z$. As shown in Section 3.4, the Z space learned by `BFM-Zero` is smooth and semantic, and it enables both zero-shot inference and few-shot adaptation. Importantly, in contrast to standard RL approaches, the set of reward functions of interest $\{r_z\}$ is not given (e.g., motion tracking) but learned, and it can represent a wide range of tasks. FB-CPR [Tirinzoni et al., 2025] extends the general FB framework by introducing a latent-conditioned discriminator to regularize the unsupervised learning process to produce policies that are close to a set of demonstrated behaviors in a motion dataset $\mathscr{M}$. Furthermore, while FB algorithm is offline, FB-CPR is trained fully online and off-policy and does not require a full-coverage offline dataset.

2.2 BFM-Zero Pre-training for Humanoid Control

Before proceeding with the description of implementation details, we identify several design choices that are crucial for achieving sim-to-real transfer in unsupervised RL.

A) Asymmetric Training. To bridge the gap between simulation (full state) and real robot (partial observability), we train the policy on observation history $o_{t,H}$, while critics have access to privileged information $(o_{t,H}, s_t)$. This setup improves policy robustness under limited sensing while leveraging privileged critics to provide accurate value estimates. Using history narrows the information gap between proprioceptive actors and privileged critics and improves adaptability under domain randomization.

B) Scaling up to Massively Parallel Environments. Inspired by recent work on large-batch off-policy RL [Seo et al., 2025], we scale training across thousands of environments with large replay buffers and high update-to-data (UTD) ratios. This enables efficient unsupervised training of a diverse family of policies while retaining stability, a crucial step for scaling humanoid pretraining.

C) Domain Randomization (DR). To enhance robustness and adaptability, we randomize key physical parameters (link masses, friction coefficients, joint offsets, torso center-of-mass) and apply perturbations and sensor noise. This prevents overfitting to simulation dynamics and ensures that policies remain stable when deployed on real hardware (see Fig. 11 in Appendix).

D) Reward Regularization. In robotics [e.g., He et al., 2025a, Zakka et al., 2025], it is common to incorporate reward regularization techniques to avoid undesirable behaviors. For example, reaching the limit of the joint may lead to highly nonlinear behaviors that are difficult to model in simulation or even damage the robot's hardware.

We train BFM-Zero within an off-policy actor-critic scheme. The policy-conditional, *history-based*, *privileged* forward map $\boldsymbol{F}$ and *privileged* backward map $\boldsymbol{B}$ are trained to minimize the temporal difference loss derived from the Bellman equation for successor measures [Touati and Ollivier, 2021]. Let $\mathscr{D}$ the replay buffer of online interactions with the simulator and ν is an arbitrary distribution over Z, we consider the following FB objective:

$$\begin{aligned}\mathscr{L}(\boldsymbol{F},\boldsymbol{B}) = \mathbb{E}\Big[\Big(\boldsymbol{F}(o_{t,H},s_t,a_t,z)^\top \boldsymbol{B}(o^+,s^+) - \gamma \overline{\boldsymbol{F}}(o_{t+1,H},s_{t+1},a_{t+1},z)^\top \overline{\boldsymbol{B}}(o^+,s^+)\Big)^2\Big] \\ - 2\mathbb{E}\Big[\boldsymbol{F}(o_{t,H},s_t,a_t,z)^\top \boldsymbol{B}(o_{t+1},s_{t+1})\Big],\end{aligned}$$

where $z \sim \nu, (o_{t,H},s_t,a_t,o_{t+1,H},s_{t+1}) \sim \mathscr{D}$, $a_{t+1} = \pi(o_{t+1,H},z)$ and $(o^+,s^+) \sim \mathscr{D}$. $\overline{\boldsymbol{F}}$ and $\overline{\boldsymbol{B}}$ denote the stop-gradient operator.

The auxiliary *history-based*, *privileged* critic $\boldsymbol{Q_R}$ that imposes safety and physical feasibility constraints by incorporating N_{aux} penalty rewards is learned with a standard Bellman residual loss:

$$\mathscr{L}(\boldsymbol{Q_R}) = \mathbb{E}_{\substack{(o_{t,H},s_t,a_t,s_{t+1})\sim\mathscr{D} \\ z\sim\nu, a_{t+1}=\pi(o_{t+1,H},z)}}\left[\left(\boldsymbol{Q_R}(o_{t,H},s_t,a_t,z) - \sum_{k=1}^{N_{\text{aux}}} r_k(s_t) - \gamma \overline{\boldsymbol{Q_R}}(o_{t+1,H},s_{t+1},a_{t+1},z)\right)^2\right].$$

Finally, we employ the *history-based*, *privileged* discriminator critic $\boldsymbol{Q_D}$ that grounds the unsupervised training toward human-like behaviors by assigning rewards based on a latent-conditioned discriminator. This acts both as a style regularization as well as a bias in the online exploration process. As in [Tirinzoni et al., 2025], we employ a variational representation of the Jensen-Shannon divergence and train the discriminator $\boldsymbol{D}$ with a GAN-style objective:

$$\mathscr{L}(\boldsymbol{D}) = -\mathbb{E}_{\tau\sim\mathscr{M},(o,s)\sim\tau}[\log(\boldsymbol{D}(o,s,z_\tau))] - \mathbb{E}_{(o,s,z)\sim\boldsymbol{D}}[\log(1-\boldsymbol{D}(o,s,z))].$$

where $z_\tau = \frac{1}{l(\tau)}\sum_{(o,s)\in\tau}\boldsymbol{B}(o,s)$ is a zero-shot imitation embedding of the motion τ. We can then fit a *style* critic $\boldsymbol{Q_D}$ with a Bellman residual loss similar to the auxiliary critic with a reward $r_d(o_t,s_t,z) = \frac{\boldsymbol{D}(o_t,s_t,z)}{1-\boldsymbol{D}(o_t,s_t,z)}$. Bringing together these critiques results in the final actor loss.

$$\mathscr{L}(\pi) = -\mathbb{E}_{\substack{(o_{t,H},s_t)\sim\mathscr{D} \\ a_t=\pi(o_{t,H},z),z\sim\nu,}}\Big[\boldsymbol{F}(o_{t,H},s_t,a_t,z)^\top z + \lambda_D \boldsymbol{Q_D}(o_{t,H},s_t,a_t,z) + \lambda_R \boldsymbol{Q_R}(o_{t,H},s_t,a_t,z)\Big].$$

Model	Test env.	Test data	Track	Rwd	Pose
BFM-Zero-*priv*	Isaac (no DR)	LAFAN1	1.0749	299.3	1.0291
BFM-Zero	Isaac (DR)	LAFAN1	1.1015	221.9	1.1387
BFM-Zero	Mujoco (DR)	LAFAN1	1.0789	207.3	1.1041
BFM-Zero	Mujoco (DR)	AMASS	1.0342		1.4735

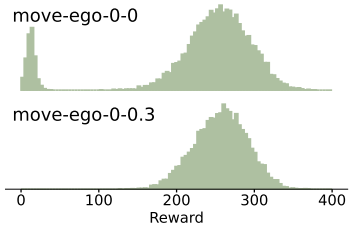


Figure 3: Tracking, reward, and goal-reaching performance across models for different testing configurations (left), and example distributions of reward evaluation scores for BFM-Zero in Isaac (DR) (right). Each metric is averaged over tasks. We consider the average return over episodes lasting 500 steps for reward, the average joint position error E_{mpjpe} averaged over the whole motion for tracking, and the error E_{mpjpe} averaged over the episode for goal-reaching.

Zero-shot inference. At test time, BFM-Zero can be used to solve different tasks in *zero-shot* fashion, i.e., without performing additional task-specific learning, planning, or fine-tuning. Given an *arbitrary* reward function $r(s)$, the corresponding Q function of π_z can be formulated as

$$Q_r^{\pi_z}(s,a) = \int_{s'} M^{\pi_z}(\mathrm{d}s'|s,a) r(s') \simeq \mathbb{E}_{s'\sim\rho}[\boldsymbol{F}(s,a,z)^\top \boldsymbol{B}(s') r(s')] = \boldsymbol{F}(s,a,z)^\top \mathbb{E}_{s'\sim\rho}[\boldsymbol{B}(s') r(s')].$$

Since $\boldsymbol{F}(s,a,z)^\top z$ is the Q function of π_z, we have $z_r = \mathbb{E}_{s'\sim\rho}[\boldsymbol{B}(s) r(s)]$. In practice, we can leverage a sample-based estimate, given by $z_r = \frac{1}{N}\sum_i r(s_i)\boldsymbol{B}(s_i)$ where $s_i \in \mathscr{D}$ and $\mathscr{D} = \{(s_i, r_i)\}$ is obtained by subsampling the online replay buffer. For a goal-reaching task, we have $z_g = B(s_g)$. Finally, for **tracking** a motion $\tau = \{s_1, \dots, s_n\}$, a sequence of policies $\{z_t\}$ is obtained as $z_t = \sum_{t'=t}^{t+H} \boldsymbol{B}(s_{t'})$, where H is a look-ahead horizon [Pirotta et al., 2024].

Few-Shot Adaptation. We can leverage optimization techniques for adaptation in latent space Z using online interaction with the simulator at test time. We demonstrate this by refining a static pose or an entire motion to maximize $J(z) = \sum_{t=0}^{T-1} \left(r_{\text{task}}(s_t) - \alpha_R \sum_{k=1}^{N_{\text{aux}}} r_k(o_t, s_t, a_t) \right)$. For **single-pose adaptation**, we use the zero-shot policy $z_0 = B(s_g, o_g)$ as initial point and apply the Cross-Entropy Method (CEM) [Rubinstein, 1999, Rubinstein and Kroese, 2004]. For **trajectory-level adaptation**, we warm-start from a tracked motion sequence and perform zero-order, sampling-based trajectory optimization over a *sequence* of latent prompts, $\mathbf{z}_{t:t+H-1}$, using a dual-loop annealing schedule in the spirit of DIAL-MPC [Xue et al., 2025]. This procedure consistently stabilizes challenging segments and reduces motion-tracking error, while retaining the human-like prior given by the discriminator without finetuning networks.

3 Experiments

In this section, we thoroughly evaluate BFM-Zero both in simulation and in real. We train BFM-Zero in a simulated version of Unitree G1 using IsaacLab [Mittal et al., 2023] at 200 Hz, while the control frequency is 50 Hz. For the behavior dataset, we use the LAFAN1 dataset [Harvey et al., 2020] retargeted to the Unitree G1 robot. The LAFAN1 dataset contains 40 several-minute-long motions. We also demonstrate generality of BFM-Zero on a Booster T1 humanoid (App. D.2).

3.1 Zero-shot Validation in Simulation

In this section, we quantitatively assess the performance and robustness of BFM-Zero along different dimensions in simulation.

Asymmetric learning and domain randomization. We consider a *privileged* version of BFM-Zero where all components of the algorithm receive privileged information. We train this model in a simulated environment with nominal dynamical parameters (*No DR*), and we test it in the very same configuration. This serves as an idealized configuration similar to the problems where unsupervised RL was previously shown to work [Tirinzoni et al., 2025], although it leads to a model that is *not deployable* on the real robot. We then compare to BFM-Zero trained and tested on a domain randomized version of the environment (*Sim DR*), which corresponds to the model actually deployed on the real robot. Overall, BFM-Zero is 2.47%, 25.86%, 10.65% worse than BFM-Zero-priv across tracking, reward, and pose reaching tasks. This shows that despite the algorithmic changes made in BFM-Zero compared to FB-CPR, the learning dynamics is still correct and the model retains a satisfactory performance compared to its idealized version. Interestingly, reward tasks suffer from a larger drop in performance. This is in part due to the sparse nature of the reward functions we consider, which makes them less forgiving to suboptimal behaviors and amplify any model error. We also conjecture that this may be related to the reward inference process with domain randomized data. In Fig. 3 we also show the distribution of the performance of BFM-Zero for two representative reward functions across repetitions of the inference process[3] and episodes. While for `move-ego-0.3` the performance is fairly consistent, for `move-ego-0.0`, we notice that a few instances obtained very poor performance. We conjecture that this is related to the increased randomness of the data observed during training due to domain randomization, which makes inference with a small subsampled dataset more brittle and prone to failure.

Sim-to-sim performance. We evaluate the robustness of BFM-Zero to the dynamics of the humanoid by testing it in Mujoco. We notice that performance difference is limited (i.e., all variations are less than 7%), showing that the domain randomization at training and the history components in the actor and critics contribute to a good level of robustness and adaptivity.

Out-of-distribution tasks. Finally, we evaluate BFM-Zero on a different set of tracking and pose reaching tasks obtained from the AMASS dataset [Mahmood et al., 2019]. We consider 175 out-of-distribution motions from the CMU subset of the AMASS and 10 manually-selected poses from the motions in the entire AMASS dataset. We run tests in Mujoco to combine different dynamics and out-of-distribution tasks. While a direct comparison of performance between LAFAN1 and AMASS tasks may be misleading due to the specific nature of the motions and poses used in the evaluation, we notice that overall BFM-Zero is able to successfully generalize and complete tracking and pose reaching even when exposed to tasks that are not represented in the training data.

3.2 Zero-shot Validation on the Real Robot

Finally, we deploy the BFM-Zero model zero-shot on a real Unitree G1 robot. In real-world validation, we aim to **1)** qualitatively confirm the model's tracking, reward optimization, and goal reaching capa-

[3] In the reward inference, we use a dataset of states randomly subsampled from the training dataset. As a result, multiple repetitions of the process may return different policies.

Figure 4: Real-World Validation of **Tracking**. *Left:* Highly dynamic dancing. *Middle:* Frequently turning during walking. *Right:* **Naturally** recover to continue track the motion.

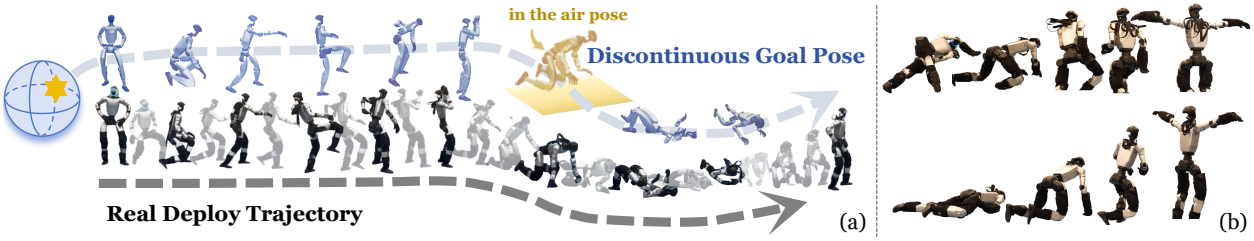


Figure 5: Real-World Validation of **Goal Reaching**. (a) Continuously goal-reaching: the blue/yellow pose denotes the goal pose, while black marks the real robot pose, and gray visualizes the transition between each pose. (b) Transition from any pose to T-pose.

bilities on a few selected tasks; **2)** assess its robustness to perturbations and failures (e.g., falling). *All results in this section come from **one** model.*

Tracking. As shown in Fig. 1 and Fig. 4, BFM-Zero enables the robot to track various motions, including styled walking motions, highly dynamic dances, fighting and sports. Even when becoming unstable or during a fall (*Right*), it demonstrates remarkably gentle, natural, and safe behavior while recovering and continues tracking seamlessly. This capability stems not merely from robustness gained through disturbance training, but mostly from *TD-based off-policy training* and the use of a GAN-based reward which explicitly encourages human-likeness and regularization terms that enable it to draw upon a rich skill library—much like a human—to adapt and complete tracking seamlessly. Additionally, to evaluate the coverage and generalization capability, we used real videos and retargeted them to the G1. Despite the suboptimal motion quality and discontinuities introduced by occlusions of monocular videos and artifacts in video estimation, the system is robust to lower quality data and can still successfully track these motions.

Goal Reaching. For the goal-reaching task, we extract a sequence of target poses by randomly sampling the goal states and discarding their velocity components. The zero-shot latent of these poses are then permuted and sequentially provided to the policy. As illustrated in Fig. 5, the robot consistently converges to a natural configuration that closely approximates the target pose, even when the target is infeasible (the Yellow one in Fig. 5). Moreover, the resulting trajectory exhibits smooth and natural transitions without the need for explicit interpolation, whether between successive and discontinuous targets(Fig. 5.a) or from an arbitrary pose to the T-pose(Fig. 5.b), demonstrating the smoothness of the learned skill space.

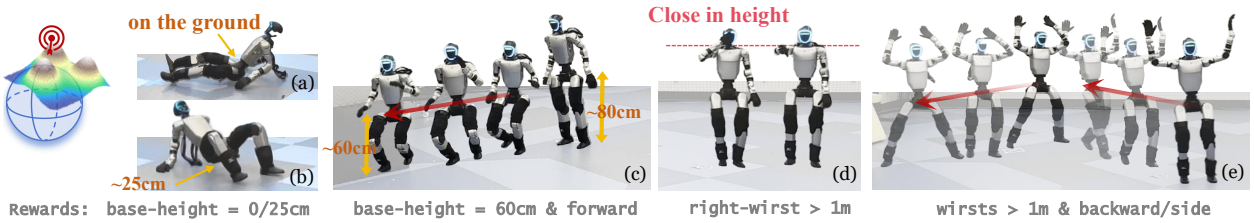


Figure 6: Real-World **Reward Optimization**. The red arrow represents the base velocity tracking target. (a) `sitting`; (b) `crouch-0.25`; (c) `move-low0.6-ego-0-0.7`; (d) Diverse behaviors from *one* reward `raisearm-m-l`; (e) combing `raisearm-m-l` with `move-ego-180-0.3` and `move-ego-90-0.7`.

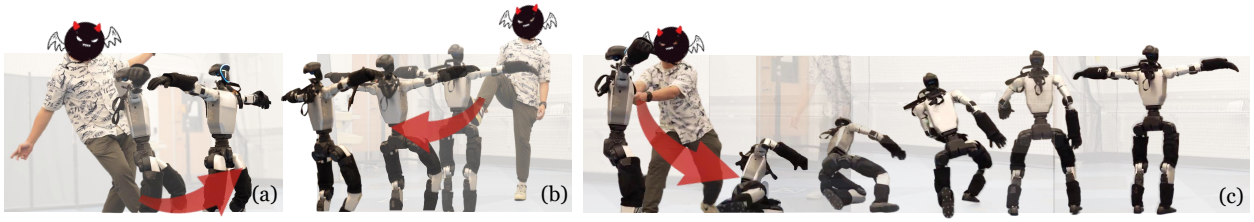


Figure 7: Disturbance Rejection: (a) Keeps steady when kicked in the leg. (b) Absorbs a hard push with one smooth rear step. (c) *Naturally* stands up and returns to T-pose after being yanked down.

Reward Optimization. We evaluate reward optimization in the real world with three task families: (i) locomotion rewards that specify base velocities and angular velocities, (ii) arm-movement rewards that command wrist height, and (iii) pelvis-height rewards that request sitting, crouching, or low-movement (Fig. 6(a–c)); reward definitions in Appendix C. With simple reward definitions, the robot faithfully executes base-height, base-velocity, and arm-movement commands. Composite skills can be derived from simply linear combination of the rewards (e.g. going backward while raising arms), demonstrating controllable skill-level interpolability. Also, given a specific reward, averaging over different mini-batches from the replay buffer yields a set of latent variables that represents a diverse collection of potential optimal modes as shown in Fig. 6(d). Formulating objectives through reward functions makes our policy intuitive for human users and receptive to language prompts.

Disturbance Rejection. One notable advantage of our policy is its strong compliance and robustness. As illustrated in Fig. 1 and 7, our framework enables the robot to withstand severe disturbances—such as fierce pushes, kicks, or even being dragged to the ground, while recovering in a natural, human-like manner. For example, after a strong forward push, the robot instinctively closes its arms, takes several rapid steps in a running-like pose, and then slowly slows down before reopening its arms (Fig. 1). This level of robustness goes beyond the typical demonstrations seen in previous works: rather than fiercely reacting to the disturbances, our policy autonomously adapts. Although it receives only a single latent z from the static T-pose as input, it can automatically deviate from the reference posture, adopt a dynamic recovery pose, and eventually return to tracking the original T-pose just as a human would.

3.3 Efficient Adaptation for `BFM-Zero`

In this section we show how we leverage adaptation to improve the zero-shot inference performance under dynamics shift.

Figure 8: Few-Shot Adaptation: (a) Single-pose adaptation improving single-leg standing under an additional payload. (b) Trajectory adaptation reduces tracking error.

Single Pose Adaptation. We perform *few-shot single-pose adaptation* in simulation to learn to stand on a single leg while carrying a payload. In simulation we increase the weight of the torso link by **4 Kg**. Starting from the zero-shot latent z^{init}, we apply 20 iterations of CEM to obtain $z^{\star}$, augmenting the rollout objective with a sparse task term $r = \mathbf{1}_{\{h_{\text{right foot}}>0.15\text{ m} \wedge \text{no-contact}\}}$, which encourages right-foot clearance while avoiding unintended contacts. We deploy $z^{\star}$ on the real robot with a 4 Kg mass rigidly attached to the torso. As shown in Figure 8 (a), without adaptation, the motion driven by z^{init} destabilizes and produces an environmental collision within 5 s. In contrast, the optimized prompt $z^{\star}$ maintains single-leg balance for over 15 s. These results indicate that prompt-level optimization alone can compensate for the payload-induced dynamics shift, without fine-tuning the model parameters.

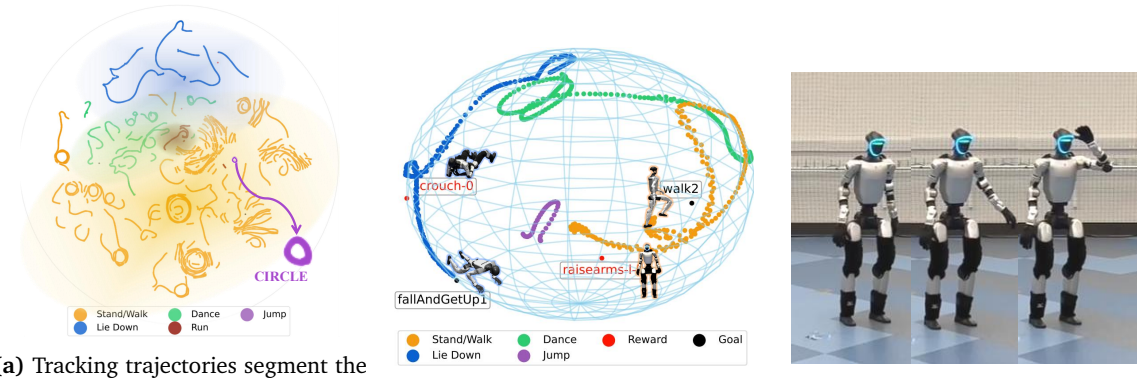
Trajectory Adaptation. For trajectory adaptation, we focus on optimizing a leaping motion under altered ground friction. We perform dual-annealing trajectory optimization [Xue et al., 2025] in simulation using the explicit tracking reward defined in [Luo et al., 2023]. We used sampling with particle count $N = 2048$, temperature schedules $\beta_1 = 0.85$ and $\beta_2 = 0.9$, and optimization iterations $M = 6$. The reward curve and before/after adaptation key-point tracking performance is shown in Fig. 8(b), showing that our method significantly improves tracking accuracy, reducing error by $\sim$*29.1*%.

3.4 The Latent Space Structure of BFM-Zero

As mentioned in Sect. 2.1, BFM-Zero provides an **interpretable** and **structured** representation of the behaviors of a humanoid robot. This representation not only facilitates understanding of the policy space but also enables instantaneous interpolation of existing skills without retraining.

Visualizing the Latent Space. To examine the structure of the latent space, we sample latent vector trajectories and project them onto a two-dimensional plane (Fig. 9a) to visualize the space, and also use a three-dimensional sphere to present representative latent generated for *tracking, reward optimization and goal reaching*(Fig. 9b) using t-SNE [van der Maaten and Hinton, 2008]. We can see the latent space is organized by motion style: semantically similar trajectories cluster, revealing a shared task centric structure.

Motion Interpolation on the Latent Space. The structured nature of $\mathcal{Z}$ enables smooth interpolation between latent representations. We can leverage Spherical Linear Interpolation [Jafari and Molaei, 2014] to generate intermediate latent vectors along the geodesic arc between the two end-points. To evaluate interpolated behaviors, we feed the resulting in-between $z_{t=0.5}$ into the BFM-Zero policy, and



(a) Tracking trajectories segment the latent space (2D).

(b) Representative latents (3D).

(c) Interpolation visualization

Figure 9: Latent space visualization and analysis.

deploy it on both simulated and real humanoid robots. As shown in Fig. 9c, the interpolated policy produces *semantically meaningful* intermediate skills in a *zero-shot* manner. These behaviors compose immediately—*no additional training* required.

4 Discussion

In this paper we showed for the first time that off-policy unsupervised RL is a viable approach to train a behavioral foundation model for whole-body control of a real humanoid robot. While BFM-Zero shows a remarkable level of generalization and robustness, it still suffers from several limitations: **1)** The scope and performance of the behaviors expressed by BFM-Zero is connected to the motions used in training. Investigating the connection between the size of motion datasets, simulated datasets, architecture and model performance (e.g., quantity and quality of the learned behaviors) and consolidating it into scaling laws is important to guide future iterations of this approach. **2)** While history-based actor and critics and domain randomization reduced the sim-to-real gap, we believe algorithms with better online adaptation capabilities are needed to reliably express more complex movements. **3)** While we performed a preliminary investigation of test-time adaptation, a more thorough understanding of fast adaptation and fine-tuning of these models is needed to broaden their practical applicability.

5 Acknowledgment

We would like to thank Tairan He and Haotian Lin for valuable discussions, and Chenyuan Hu for assistance with the experiments. Guanya Shi holds concurrent appointments as an Assistant Professor at Carnegie Mellon University and as an Amazon Scholar. This paper describes work performed at Carnegie Mellon University and is not associated with Amazon.

References

Siddhant Agarwal, Harshit Sikchi, Peter Stone, and Amy Zhang. Proto successor measure: Representing the behavior space of an rl agent. *arXiv preprint arXiv:2411.19418*, 2024.

Firas Al-Hafez, Guoping Zhao, Jan Peters, and Davide Tateo. Locomujoco: A comprehensive imitation learning benchmark for locomotion. In *6th Robot Learning Workshop, NeurIPS*, 2023.

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith LLontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: an open foundation model for generalist humanoid robots. *CoRR*, abs/2503.14734, 2025.

Zixuan Chen, Xialin He, Yen-Jen Wang, Qiayuan Liao, Yanjie Ze, Zhongyu Li, S. Shankar Sastry, Jiajun Wu, Koushil Sreenath, Saurabh Gupta, and Xue Bin Peng. Learning smooth humanoid locomotion through lipschitz-constrained policies. *CoRR*, abs/2410.11825, 2024.

Zixuan Chen, Mazeyu Ji, Xuxin Cheng, Xuanbin Peng, Xue Bin Peng, and Xiaolong Wang. GMT: general motion tracking for humanoid whole-body control. *CoRR*, abs/2506.14770, 2025.

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.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024.

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. *arXiv preprint arXiv:2501.02116*, 2025.

Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher J. Pal. Robust motion in-betweening. *ACM Trans. Graph.*, 39(4):60, 2020.

Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, Linxi Fan, and Yuke Zhu. HOVER: versatile neural whole-body controller for humanoid robots. *CoRR*, abs/2410.21229, 2024.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, Zeji Yi, Guannan Qu, Kris Kitani, Jessica K. Hodgins, Linxi Fan, Yuke

Zhu, Changliu Liu, and Guanya Shi. ASAP: aligning simulation and real-world physics for learning agile humanoid whole-body skills. *CoRR*, abs/2502.01143, 2025a.

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 *Conference on Robot Learning*, pages 1516–1540. PMLR, 2025b.

Xialin He, Runpei Dong, Zixuan Chen, and Saurabh Gupta. Learning getting-up policies for real-world humanoid robots, 2025c. URL https://arxiv.org/abs/2502.12152.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura M. Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *CoRR*, abs/2504.16054, 2025.

Mehdi Jafari and Habib Molaei. Spherical linear interpolation and bézier curves. *General Scientific Researches*, 2(1):13–17, 2014.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Paul Foster, Pannag R. Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *CoRL*, volume 270 of *Proceedings of Machine Learning Research*, pages 2679–2713. PMLR, 2024.

Qiayuan Liao, Takara E. Truong, Xiaoyu Huang, Guy Tevet, Koushil Sreenath, and C. Karen Liu. Beyondmimic: From motion tracking to versatile humanoid control via guided diffusion, 2025. URL https://arxiv.org/abs/2508.08241.

Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *ICCV*, pages 10861–10870. IEEE, 2023.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris M. Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. In *ICLR*. OpenReview.net, 2024.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: archive of motion capture as surface shapes. 2019.

Diganta Misra. Mish: A self regularized non-monotonic activation function. In *BMVC*. BMVA Press, 2020.

Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics Autom. Lett.*, 8(6):3740–3747, 2023.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with hilbert representations. *arXiv preprint arXiv:2402.15567*, 2024.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.*, 37(4):143, 2018.

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions on Graphics*, 41 (4):1–17, July 2022. ISSN 1557-7368. doi: 10.1145/3528223.3530110. URL http://dx.doi.org/10.1145/3528223.3530110.

Matteo Pirotta, Andrea Tirinzoni, Ahmed Touati, Alessandro Lazaric, and Yann Ollivier. Fast imitation via behavior foundation models. In *ICLR*. OpenReview.net, 2024.

Ilija Radosavovic, Tete Xiao, Bike Zhang, Trevor Darrell, Jitendra Malik, and Koushil Sreenath. Real-world humanoid locomotion with reinforcement learning. *Sci. Robotics*, 9(89), 2024a.

Ilija Radosavovic, Bike Zhang, Baifeng Shi, Jathushan Rajasegaran, Sarthak Kamat, Trevor Darrell, Koushil Sreenath, and Jitendra Malik. Humanoid locomotion as next token prediction. *CoRR*, abs/2402.19469, 2024b.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and computing in applied probability*, 1(2):127–190, 1999.

Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2004.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

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

Younggyo Seo, Carmelo Sferrazza, Haoran Geng, Michal Nauman, Zhao-Heng Yin, and Pieter Abbeel. Fasttd3: Simple, fast, and capable reinforcement learning for humanoid control. *CoRR*, abs/2505.22642, 2025.

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauzá, Michiel Blokzijl, Steven Bohez, Konstantinos Bousmalis, Anthony Brohan, Thomas Buschmann, Arunkumar Byravan, Serkan Cabi, Ken Caluwaerts, Federico Casarini, Oscar Chang, José Enrique Chen, Xi Chen, Hao-Tien Lewis Chiang, Krzysztof Choromanski, Davide D'Ambrosio, Sudeep Dasari, Todor Davchev, Coline Devin, Norman Di Palo, Tianli Ding, Adil Dostmohamed, Danny Driess, Yilun Du, Debidatta Dwibedi, Michael Elabd, Claudio Fantacci, Cody Fong, Erik Frey, Chuyuan Fu, Marissa Giustina, Keerthana Gopalakrishnan, Laura Graesser, Leonard Hasenclever, Nicolas Heess, Brandon Hernaez, Alexander Herzog, R. Alex Hofer, Jan Humplik, Atil Iscen, Mithun George Jacob, Deepali Jain, Ryan Julian, Dmitry Kalashnikov, M. Emre Karagozler, Stefani Karp, J. Chase Kew, Jerad Kirkland, Sean Kirmani, Yuheng Kuang, Thomas Lampe, Antoine Laurens, Isabel Leal, Alex X. Lee, Tsang-Wei Edward Lee, Jacky Liang, Yixin Lin, Sharath Maddineni, Anirudha Majumdar, Assaf Hurwitz Michaely, Robert

Moreno, Michael Neunert, Francesco Nori, Carolina Parada, Emilio Parisotto, Peter Pastor, Acorn Pooley, Kanishka Rao, Krista Reymann, Dorsa Sadigh, Stefano Saliceti, Pannag Sanketi, Pierre Sermanet, Dhruv Shah, Mohit Sharma, Kathryn Shea, Charles Shu, Vikas Sindhwani, Sumeet Singh, Radu Soricut, Jost Tobias Springenberg, Rachel Sterneck, Razvan Surdulescu, Jie Tan, Jonathan Tompson, Vincent Vanhoucke, Jake Varley, Grace Vesom, Giulia Vezzani, Oriol Vinyals, Ayzaan Wahid, and Stefan Welker. Gemini robotics: Bringing AI into the physical world. *CoRR*, abs/2503.20020, 2025.

Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. Calm: Conditional adversarial latent models for directable virtual characters. In *ACM SIGGRAPH 2023 Conference Proceedings*, SIGGRAPH '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701597. doi: 10.1145/3588432.3591541. URL https://doi.org/10.1145/3588432.3591541.

Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. Maskedmimic: Unified physics-based character control through masked motion inpainting. *ACM Trans. Graph.*, 43(6):209:1–209:21, 2024.

Andrea Tirinzoni, Ahmed Touati, Jesse Farebrother, Mateusz Guzek, Anssi Kanervisto, Yingchen Xu, Alessandro Lazaric, and Matteo Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models. In *ICLR*. OpenReview.net, 2025.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In *NeurIPS*, pages 13–23, 2021.

Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *The Eleventh International Conference on Learning Representations*, 2023.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

Haoru Xue, Chaoyi Pan, Zeji Yi, Guannan Qu, and Guanya Shi. Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4974–4981. IEEE, 2025.

Kangning Yin, Weishuai Zeng, Ke Fan, Zirui Wang, Qiang Zhang, Zheng Tian, Jingbo Wang, Jiangmiao Pang, and Weinan Zhang. Unitracker: Learning universal whole-body motion tracker for humanoid robots. *CoRR*, abs/2507.07356, 2025.

Kevin Zakka, Baruch Tabanpour, Qiayuan Liao, Mustafa Haiderbhai, Samuel Holt, Jing Yuan Luo, Arthur Allshire, Erik Frey, Koushil Sreenath, Lueder A. Kahrs, Carmelo Sferrazza, Yuval Tassa, and Pieter Abbeel. Mujoco playground. *CoRR*, abs/2502.08844, 2025.

Weishuai Zeng, Shunlin Lu, Kangning Yin, Xiaojie Niu, Minyue Dai, Jingbo Wang, and Jiangmiao Pang. Behavior foundation model for humanoid robots, 2025. URL https://arxiv.org/abs/2509.13780.

Zhikai Zhang, Chao Chen, Han Xue, Jilong Wang, Sikai Liang, Yun Liu, Zongzhang Zhang, He Wang, and Li Yi. Unleashing humanoid reaching potential via real-world-ready skill space. *CoRR*, abs/2505.10918, 2025.

Yifan Zhong, Xuchuan Huang, Ruochong Li, Ceyao Zhang, Yitao Liang, Yaodong Yang, and Yuanpei Chen. Dexgraspvla: A vision-language-action framework towards general dexterous grasping. *CoRR*, abs/2502.20900, 2025.

Contents

1 Introduction **2**

2 BFM-Zero for Humanoid Whole-body Control **3**

2.1 Unsupervised RL with Forward-Backward Representations 4

2.2 BFM-Zero Pre-training for Humanoid Control 5

3 Experiments **7**

3.1 Zero-shot Validation in Simulation 8

3.2 Zero-shot Validation on the Real Robot 8

3.3 Efficient Adaptation for BFM-Zero 10

3.4 The Latent Space Structure of BFM-Zero 11

4 Discussion **12**

5 Acknowledgment **12**

A Related Work **19**

B Training details **19**

B.1 Training Hyperparameter Settings 19

B.2 Network Architectures 21

B.3 BFM-Zero Algorithm Details 22

B.4 Training Environments 22

C Tasks and Metrics **23**

D Additional Results **25**

D.1 Data Size and Model Size 25

D.2 Application of BFM-Zero on Booster T1 26

A Related Work

In recent years, learning-based methods have made significant progress in whole-body control for humanoid robots. The largest body of work has focused on simulated humanoids. While these methods have demonstrated impressive capabilities in generating complex and dynamic behaviors using reinforcement learning [Peng et al., 2018, Luo et al., 2023, 2024, Tessler et al., 2024], sim-to-real transfer remains a critical challenge in deploying learned policies on real-world humanoid robots. Various strategies have been proposed to bridge this gap, including domain randomization, system identification, asymmetric training, etc. However, the majority of these methods focus on single-task learning, where a policy is trained to perform a specific task, such as walking, running and get up [Radosavovic et al., 2024a,b, Chen et al., 2024, Seo et al., 2025, Zakka et al., 2025, He et al., 2025c].

Recently, mostly 2025, there has been a surge of interest in developing multi-task and generalist humanoid control policies that can perform a wide range of tasks [He et al., 2024, 2025a, Zhang et al., 2025, Zeng et al., 2025, Yin et al., 2025, Chen et al., 2025]. The majority of these methods builds on top of approaches developed for simulated humanoids, and enhance them to be robust enough for sim-to-real transfer. While ASAP [He et al., 2025a] pre-train motion tracking policies in simulation and deploy them on the real robot to collect data to train a delta (residual) action model, the most common approach is to first train a motion tracking policy (or multiple policies) in simulation, and then distill it into a single multi-task policy that can perform all the skills in the motion dataset. Common approaches for distillation include using a conditional variational autoencoder to learn a latent space of skills and doing online distillation [He et al., 2024, Yin et al., 2025, Zeng et al., 2025, Chen et al., 2025, Zhang et al., 2025] or using diffusion models [Liao et al., 2025]. However, all these methods require two stages of training to enable promptable policies, they are inherently limited by the quality of the motion since the base policies are trained to track the motion, and they relay on on-policy RL algorithms. Our method represents a significant departure from this paradigm by directly learning a promptable multi-task policy using an off-policy RL algorithm, which offer a much more reach and structured space of skills, and is not limited by the quality of the motion dataset.

B Training details

B.1 Training Hyperparameter Settings

The agent interacts with the environment via episodes of fix length $T = 500$ steps. The algorithm has access to the dataset $\mathscr{M}$ containing observation-only motions. Similarly to [Tirinzoni et al., 2025], the initial state distribution of an episode is a mixture between randomly generated falling positions and states in $\mathscr{M}$ (motion initialization). We use prioritization to sample motions from $\mathscr{M}$ and, inside a motion, the state is uniformly sampled. We use an exponential prioritization scheme based on the agent's ability to track a motion. To have a more fine-grained prioritization, we split the 40 LAFAN1 [Harvey et al., 2020] motions into chunks of 10 seconds. Every N_{eval} interaction steps, we evaluate all the motions and update the priorities base on the earth mover's distance [Rubner et al., 2000, EMD]. For

each motion $m \in \mathscr{M}$, the priority is given by

$$p(m) \propto 2^{\max\left\{0.5;\, \min\left\{\mathrm{EMD}(m),2\right\}\right\}\cdot 4}$$

We take inspiration from the recipe in FastTD3 [Seo et al., 2025] to scale up unsupervised off-policy RL to using massively parallel environments. We use standard MLPs for all the components of the model, even for handling history. We simulate N_{env} parallel (and independent) environments at each step. We scale the buffer size accordingly to the number of environments, following the rule $N_{\text{buffer}} \times N_{\text{env}} \times T$. We use a batch size of N_{batch} and we use an update-to-data ratio of N_{ups} gradient steps per (parallel) environment step. We train the model for a total number of environment steps $N_{\text{train}} = \frac{N_{\text{grad}} N_{\text{env}}}{N_{\text{ups}}}$. We report the value of these parameters in Tab. 1, the missing parameters are as in [Tirinzoni et al., 2025].

Parameter	**Value**
Environment and Training Setup	
History Length H	4
Episode Length T	500
N_{env}	1024
N_{batch}	1024
N_{ups}	16
N_{grad}	3M
N_{train}	$\approx$ 192M
N_{buffer}	10
N_{eval}	$N_{\text{train}}/20$
Buffer Size (transitions)	$\approx$ 5M
Discount Factor	0.98
Number of Seeding Steps	$10 \cdot N_{\text{env}}$
Fall Initialization Probability	0.3
Learning and Regularization	
Sequence Length (Trajectory Sampling)	8
Latent Dimension d	256
Discriminator Reg. Coef. α_D	0.05
Reward Reg. Coef. α_R	0.02
Gradient Penalty	10
Learning Rate F	$3 \cdot 10^{-4}$
Learning Rate B	10^{-5}
Learning Rate D	10^{-5}
Learning Rate Actor π	$3 \cdot 10^{-4}$
Learning Rate Q_D	$3 \cdot 10^{-4}$
Learning Rate Q_R	$3 \cdot 10^{-4}$
Orthonormality Loss Coefficient	100
Inference	
Number of samples for reward inference	400000
Tracking look ahead in sim	Seq. length
Tracking look ahead in real	3 (real)

Table 1: Training settings.

B.2 Network Architectures

We use a residual architecture for the actor and the critics with blocks akin to those of transformer architectures [Vaswani et al., 2017], involving residual connections, layer normalization, and Mish activation functions [Misra, 2020]. We use an ensemble composed of two networks for critics. For discriminator and backward map we use a standard MLP with ReLu activation (see Fig. 10). Refer to Tab. 2 for more details.

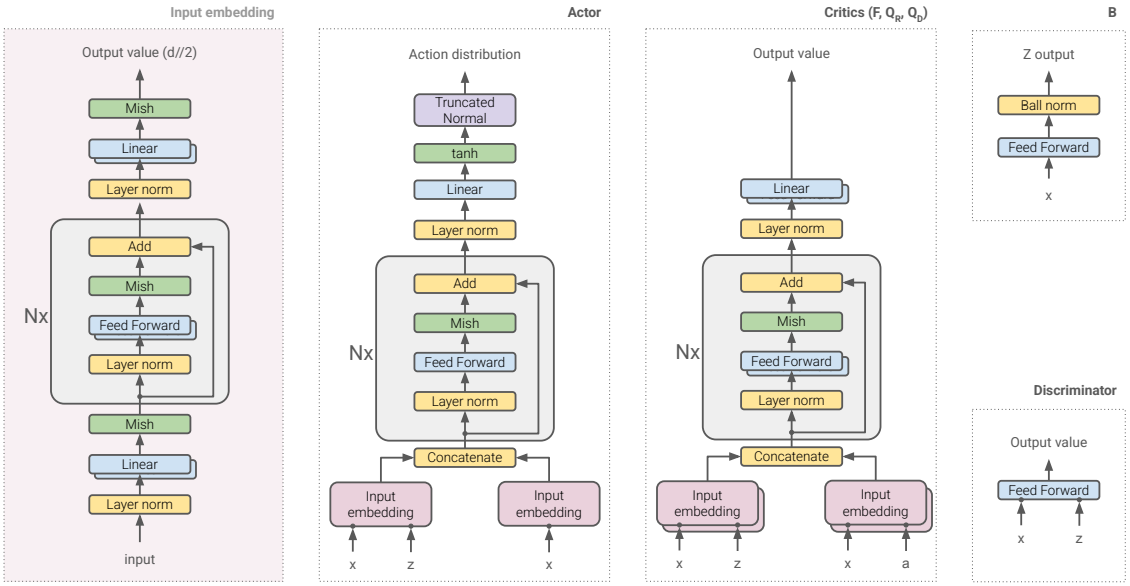


Figure 10: Visual representation of the network architectures.

Hyperparameter	**Critics (F, Q_D, Q_R)**	**Actor**	**Discriminator**	**B**
Input Variables	(x, a, z)	(x, z)	(x, z)	(x)
Output Dim	F: d, Q_D, Q_R: 1	29	1	d
Observation Variable x	$(o_{t,H}, s_t)$	$o_{t,H}$	(s_t, o_t)	(s_t, o_t)
Embedding Residual Blocks	4	4	–	–
Embedding Hidden Units	2048	2048	–	–
Residual Blocks	6	6	–	–
Feed Forward Hidden Layers	1	1	2	1
Feed Forward Hidden Units	2048	2048	1024	256
Activations	Mish	Mish	ReLU	ReLU
Number of Parallel Networks	2	1	1	1
Num. Parameters (no target)	F: 135.8M, Q_D, Q_R: 134.8M	31.9M	2.9M	0.2M
Total Parameters		**440.5M**		

Table 2: Network architecture parameters used for real tests. s_t is the privileged information and o_t is the proprioceptive information. $o_{t,H} = \{o_{t-H}, a_{t-H}, \ldots, o_t\}$ denotes the history of proprioceptive states and actions. We exclude target networks when counting the number of parameters.

B.3 BFM-Zero Algorithm Details

We provide here a sketch of BFM-Zero in (Alg. 1). We report the algorithm without parallel networks for clarity. For clarity, we also report the FB loss here. Let $a'_i \sim \pi(x'_i, z_i)$ where $x_i = (o_{i,H}, s_i)$, then

$$\begin{aligned}
\ell_{\text{fb}} =& \frac{1}{2n(n-1)} \sum_{i \neq k} \Big(F(x_i, a_i, z_i)^\top B(s'_k, o'_k) - \gamma \overline{F}(x'_i, a'_i, z_i)^\top \overline{B}(s'_k, o'_k) \Big)^2 \\
& - \frac{1}{n} \sum_i F(x_i, a_i, z_i)^\top B(o'_i, s'_i) \\
& + \frac{1}{2n(n-1)} \sum_{i \neq k} \Big(B(s'_i, o'_i)^\top B(s'_k, o'_k) \Big)^2 - \frac{1}{n} \sum_{i \in [n]} B(s'_i, o'_i)^\top B(s'_i, o'_i) \\
& + \frac{1}{n} \sum_{i \in [n]} \Big(F(x_i, a_i, z_i)^\top z_i - \overline{B}(s'_i, o'_i) \Sigma_{\overline{B}} z_i - \gamma \overline{F}(x'_i, a'_i, z_i)^\top z_i \Big)^2
\end{aligned} \tag{B.1}$$

Algorithm 1 BFM-Zero Pre-Training

Initialize empty train buffer: $\mathscr{D}_{\text{online}} \leftarrow \emptyset$
Initialize expert buffer $\mathscr{M}$ with action-free trajectories
for $t = 1, \dots$ **do**
  //Online interaction
  Sample $\boldsymbol{z}_t = \{z_e\}_{e=1}^{N_{\text{env}}} \in \mathbb{R}^{N_{\text{env}} \times d}$ (if needed)
  Execute $\boldsymbol{a}_t \sim \pi(\boldsymbol{o}_{t,H}, \boldsymbol{z}_t) \in \mathbb{R}^{N_{\text{env}} \times A}$ in the *simulated* environments
  Store $(\boldsymbol{s}_t, \boldsymbol{o}'_{t,H}, \boldsymbol{a}_t, \boldsymbol{s}'_t, \boldsymbol{o}'_{t+1,H}, \boldsymbol{z}_t)$ in $\mathscr{D}_{\text{online}}$
  //Update
  for $j = 1, \dots, N_{\text{ups}}$ **do**
    Sample a batch of $n = N_{\text{batch}}$ transitions $\{(o_{i,H}, s_i, a_i, o'_{i,H}, s'_i, z_i)\}_{i=1}^n$ from $\mathscr{D}_{\text{online}}$
    Sample a batch of $\frac{n}{T_{\text{seq}}}$ sequences $\{(w_{j,1}, w_{j,2} \dots, w_{j,T_{\text{seq}}})\}_{j=1}^{\frac{n}{T_{\text{seq}}}}$ from $\mathscr{M}$ where $w = (s_t, o_t)$
    //Encode expert and update discriminator
    $z_j \leftarrow \frac{1}{T_{\text{seq}}} \sum_{t=1}^{T_{\text{seq}}} B(w_{j,t})$; $z_j \leftarrow \sqrt{d} \frac{z_j}{\|z_j\|_2}$
    $\ell_{\text{discriminator}} = -\frac{1}{n} \sum_{j=1}^{\frac{n}{T_{\text{seq}}}} \sum_{t=1}^{T_{\text{seq}}} \log D(w_{j,t}, z_j) - \frac{1}{n} \sum_{i=1}^n \log(1 - D(s_i, o_i, z_i))$
    //Update representation F and B so that $F(s,a;z)^\top B(s') \approx M^{\pi_z}(ds'|s,a)$
    Refer to Eq. B.1
    //note that D does not use history
    Compute discriminator reward: $r_i^D \leftarrow \log(D(s_i, o_i, z_i)) - \log(1 - D(s_i, o_i, z_i)), \quad \forall i \in [n]$
    Let $x_i = (o_{i,H}, s_i)$ and sample $u_i \sim \pi(o_{i,H}, z_i)$ for all $i \in [n]$. Then
    $\ell_{\texttt{critic}_D} = \frac{1}{n} \sum_{i \in [n]} \big(Q_D(x_i, a_i, z_i) - r_i^D - \gamma \overline{Q_D}(x'_i, a_i, z_i) \big)^2$
    $\ell_{\texttt{critic}_R} = \frac{1}{n} \sum_{i \in [n]} \big(Q_R(x_i, a_i, z_i) - \sum_k r_k^{\text{aux}}(x'_i) - \gamma \overline{Q_R}(x'_i, a_i, z_i) \big)^2$
    $\ell_{\texttt{actor}} = -\frac{1}{n} \sum_{i \in [n]} \Big(F(x_i, u_i, z_i)^\top z_i + \alpha_D Q_D(x_i, u_i, z_i) + \alpha_R Q_R(x_i, u_i, z_i) \Big)$
    //Update target networks

B.4 Training Environments

To better facilitate sim-to-real transfer, we incorporated domain randomization, additive observation noise and regularization rewards in the training environment. Refer to Fig 11 for details.

Domain Randomization

Parameter	Range
COM Offset [m]	$\mathcal{U}([-0.02, 0.02])$
Link Mass	$\mathcal{U}([0.95, 1.05])$
Friction	$\mathcal{U}([-0.5, 1.25])$
Default Joint Pos [m]	$\mathcal{U}([-0.02, 0.02])$
Push Robots [m/s]	$\mathcal{U}([0, 0.5])$

Additive Observation Noise

Observation	Range
$q_t - \bar{q}$	$\mathcal{U}([-0.01, 0.01])$
$\dot{q}_t$	$\mathcal{U}([-0.5, 0.5])$
grav_t	$\mathcal{U}([-0.05, 0.05])$
$\dot{\omega}_t^{\text{root}}/4$	$\mathcal{U}([-0.05, 0.05])$

Regularization Rewards

Name	Weight
DoF Limit	-10
Action Rate	-0.1
Self Contact	-1
Feet Orientation	-0.4
Ankle Roll	-4
Feet Slip	-2

Figure 11: Details in training environment.

C Tasks and Metrics

In this section, we provide a complete description of the tasks and metrics.

Goal-based evaluation We have manually extracted 21 "stable" poses (i.e., states with zero velocities) from the train dataset (i.e., LAFAN1) and 10 poses from the test dataset (i.e., AMASS). We report the selected poses from LAFAN1 in Fig 12. To evaluate how close is the agent to the goal pose, we use the joint error defined as following

$$E_{\text{mpjpe}}(e, g) = \frac{1}{|e|} \sum_{t=1}^{|e|} \|q_t(e) - q(g)\|_2$$

where e is an episode and q is the joint position (i.e., 29D). We report the average across goals. The episodes are fixed in length $H = 500$.

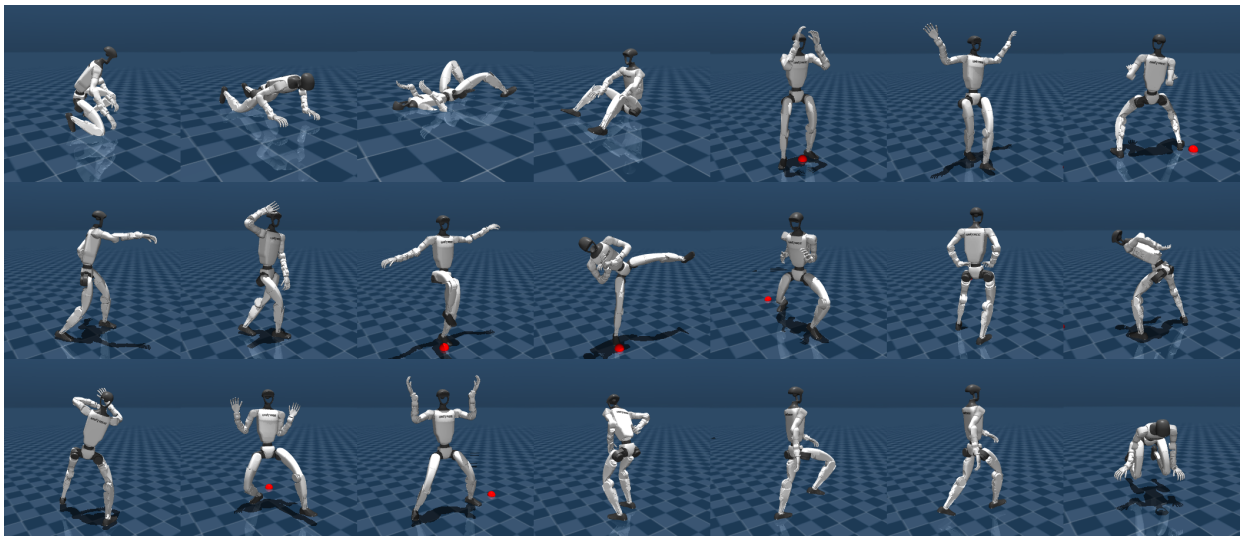


Figure 12: Goal poses selected from frames of the LAFAN1 dataset [Harvey et al., 2020].

Tracking evaluation This evaluation aims to assess the ability of the model to imitate a sequence of poses, ideally matching both positions and velocities. We evaluate the agent both on the train dataset (i.e., LAFAN1) and on out-of-distribution motions selected from AMASS (retargeted to G1). In particular, we randomly selected 175 motions from the CMU dataset of AMASS. For evaluation, we use the same metric as in goal evaluation, i.e.,

$$E_{\text{mpjpe}}(e,m) = \frac{1}{|e|}\sum_{t=1}^{|e|} \|q_t(e) - q_t(m)\|_2$$

and we report the average across motions.

Reward evaluation We define 6 reward categories inspired by [Tirinzoni et al., 2025]. The reward can be expressed as a function of the next state and normalized in $[0,1]$.

Standing. We evaluate the agent's ability to stand with the pelvis at different heights. `move-ego-0-0` requires pelvis above 60cm and zero velocity, while `move-ego-low0.5-0-0` requires the pelvis to be between 50cm and 65cm.

Locomotion. This category includes rewards related that requires the agent to move at a certain speed, in a certain direction and at a certain height. We consider 5 representative rewards (`move-ego-0-0.7`, `move-ego-90-0.7`, `move-ego-(-90)-0.7`, `move-ego-0-0.3`, `move-ego-180-0.3`) which include forward, lateral and backward movement. We additionally test also walking forward but with the pelvis at a low height (`move-ego-low0.6-0-0.7`).

Rotation. We require the robot to rotate along the vertical axis (i.e., while standing). We consider rotating clockwise and counterclockwise (i.e., `rotate-z-5-0.5` and `rotate-z-(-5)-0.5`).

Ground poses. To further stress the ability of the model to control the vertical position, we define rewards requiring the agent to sit on the ground (`sitting`) or having the pelvis slightly above the ground (`crouch-0.25` is about 25cm above the ground).

Arm raise. We require the robot to stand in a steady position and to reach a certain vertical position with the arms (measured at the wrists). We consider low ($z \in [0.6m, 0.8m]$) and medium ($z > 1m$) positions for the wrists, with soft margins (`raisearms-l-l`, `raisearms-l-m`, `raisearms-m-l`, `raisearms-m-m`).

Combined rewards. We finally evaluate the ability of the agent to maximize rewards that require combining multiple skills. In particular, we test combinations of locomotion and rotation with arm movements. We selected 8 combinations of rewards.

Overall, we tested 24 rewards and evaluated perfomance via the cumulative return over episodes of $T = 500$ steps. The initial state of an episode is the default pose.

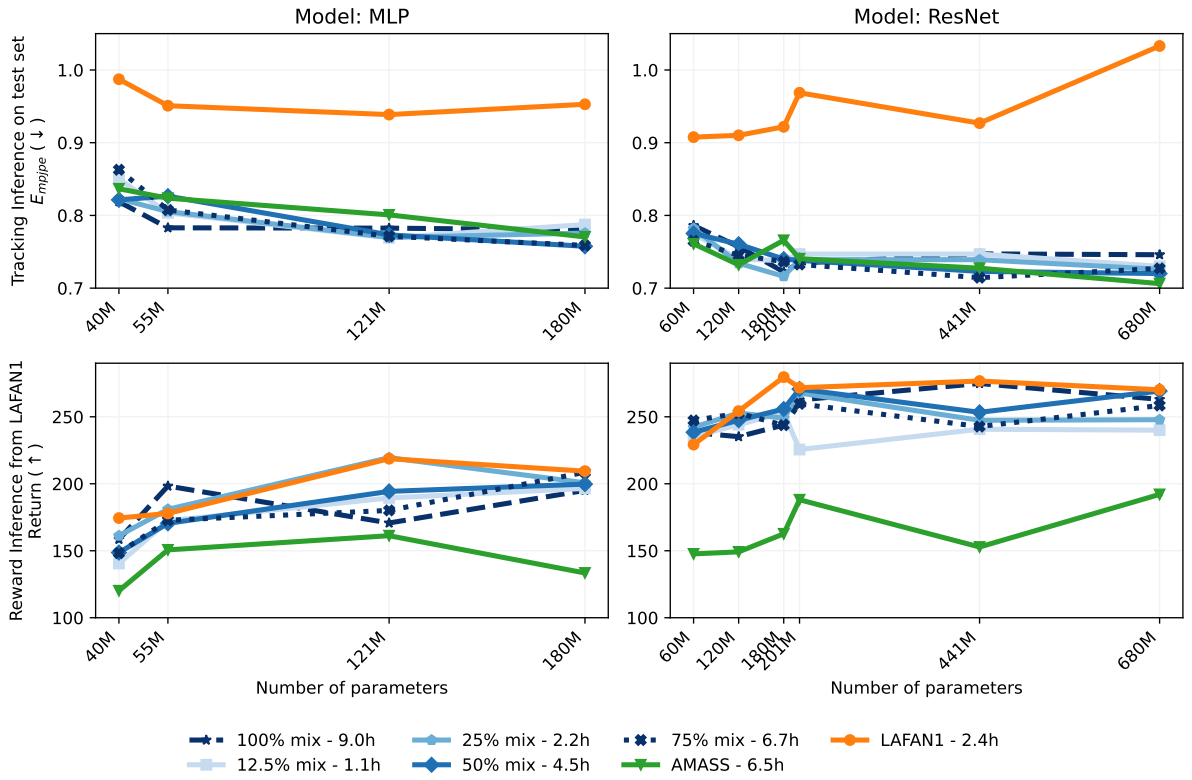


Figure 13: Tracking and reward performance on the test set for different models and datasets. The lower the better for tracking and the higher the better for reward.

D Additional Results

D.1 Data Size and Model Size

We perform ablations on both the data and model size. For training the model in the main paper, we used only the LAFAN1 dataset [Harvey et al., 2020]. In these ablations, we additionally leverage motions from the CMU and BMLHandball subsets of AMASS [Mahmood et al., 2019]. We consider the individual datasets (referred to as LAFAN1 and AMASS in the figure), as well as datasets obtained by merging X percent of the two datasets (with $X = \{12.5\%, 25\%, 50\%, 75\%, 100\%\}$). We evaluate different network architectures, including simple feed-forward networks and residual architectures with a varying number of blocks (see Tab. 3). For tracking, we use the same test dataset as in [Tirinzoni et al., 2025], but we removed motions from CMU and BMLHandball to ensure complete separation from the training datasets. For reward inference, we use 600,000 samples from the LAFAN1 dataset for all configurations. We report the results of our ablation in Fig. 13 over a single seed.

As we increase the total capacity of the model, tracking performance improves for almost all of the training mocap datasets. LAFAN1 is the only case where performance saturates quite early. We believe this is because the training dataset is a subset of the AMASS dataset, and despite being separated from the training data, it is likely much closer to the motions in CMU and BMLHandball than to those in

		Number of Parameters						
Architecture	Model	π	Q_R	B	Q_D	D	F	Total
ResNet	3-block, 2048dim	19.3M	59.2M	201k	59.2M	2.9M	60.3M	201.1M
ResNet*	6-block, 2048dim	31.9M	134.8M	201k	134.8M	2.9M	135.9M	440.5M
ResNet	9-block, 2048dim	44.5M	210.4M	201k	210.4M	2.9M	211.5M	679.9M
ResNet	3-block, 1024dim	5.5M	17.0M	201k	17.0M	2.9M	17.6M	60.2M
ResNet	6-block, 1024dim	8.6M	36.0M	201k	36.0M	2.9M	36.5M	120.1M
ResNet	9-block, 1024dim	11.8M	54.9M	201k	54.9M	2.9M	55.4M	180.1M
MLP	2-layer, 1024dim	4.4M	10.7M	201k	10.7M	2.9M	11.2M	40.1M
MLP	2-layer, 2048dim	15.1M	34.0M	201k	34.0M	2.9M	35.0M	121.2M
MLP	4-layer, 1024dim	6.5M	14.9M	201k	14.9M	2.9M	15.4M	54.8M
MLP	4-layer, 2048dim	23.5M	50.8M	201k	50.8M	2.9M	51.8M	179.9M

Table 3: Configurations of the architectures and total number of parameters. $\star$ denotes the configuration used in the main paper.

LAFAN1. We can further notice that residual architectures achieve better performance w.r.t. simple MLP architectures, and we can scale residual architectures to larger sizes. Furthermore, we found training to be instable when scaling MLP to larger architectures.

Similarly, we observe a mild improvement trend for reward inference when increasing the model size. However, training with LAFAN1 (in some proportion) appears to be important in this case, as reward performance drops when we train only with the subset of AMASS. We also evaluated reward inference performance using both the training buffer and the training motion set. In both cases, the average performance decreases, with a much more significant drop when using the training buffer. We believe this may be due to the fact that samples in the buffer are collected with domain randomization, whereas the motion buffers are not randomized. Selecting the optimal dataset for reward inference could be an interesting direction for future research.

D.2 Application of BFM-Zero on Booster T1

We additionally evaluate the generality of our framework by testing BFM-Zero on Booster T1 humanoid robot. The LAFAN1 dataset is retargeted to T1 using LocoMujoco [Al-Hafez et al., 2023] and we train the policy with exact same hyper-parameters as G1. The algorithm shows strong generalization ability, allowing T1 also to perform natural walking and expressive dancing motions, as shown in Figure 15.

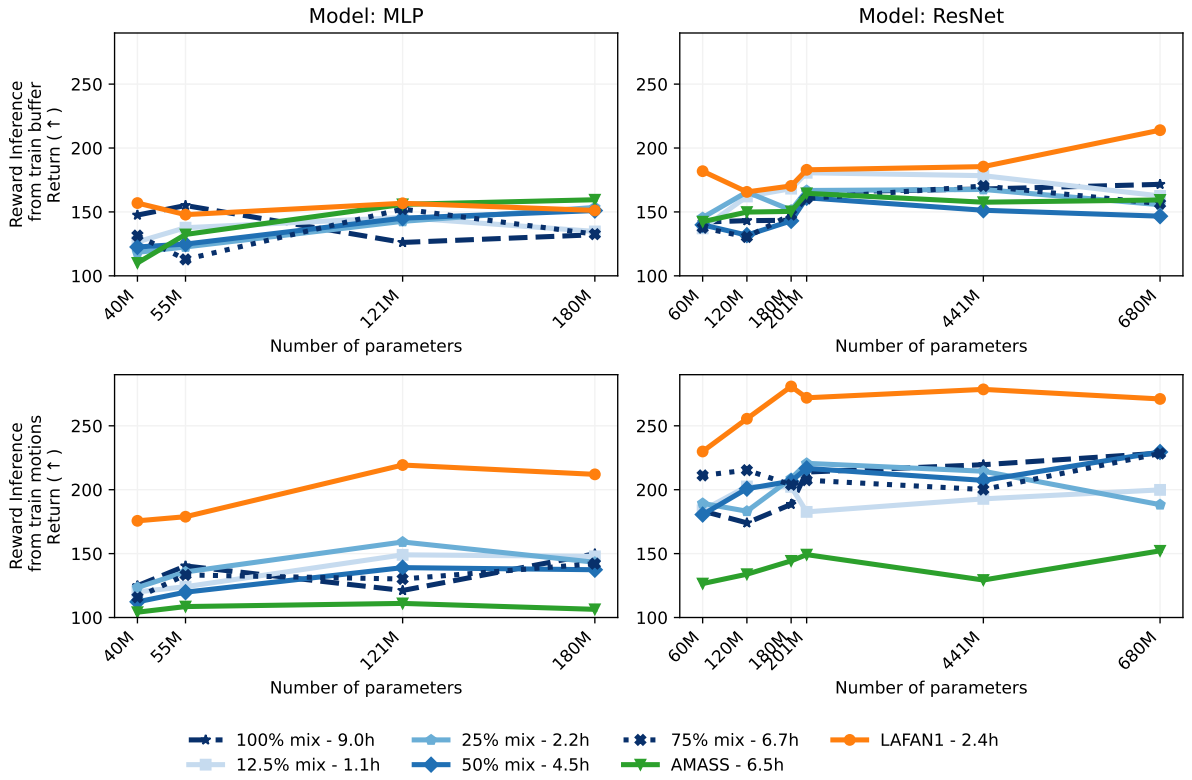


Figure 14: Reward inference performance when using the experience generated by the agent (i.e., online replay buffer) or the motion dataset used for training. We get better reward performance when using the motion dataset, in particular when using LAFAN1 (see Fig. 13).

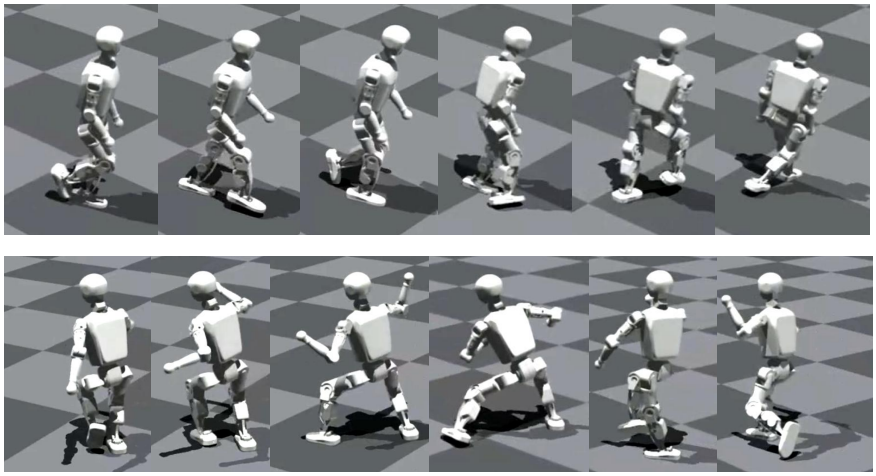


Figure 15: Application of BFM-Zero on Booster T1.