UFO: A General Unsupervised Reinforcement Learning Framework for Humanoid COntrol

RoboParty Lab Team

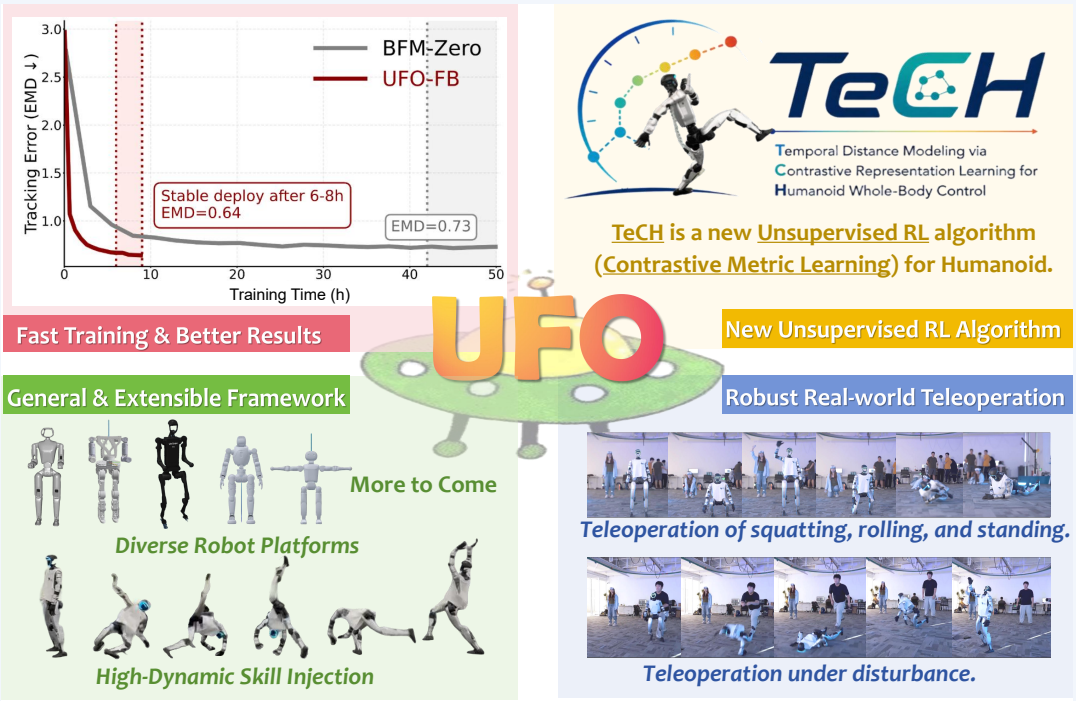


Figure 1: Overview of **UFO**. **UFO** enables efficient training, integrates a novel unsupervised RL algorithm (**TeCH**), provides a general and extensible framework for diverse humanoid robots, and demonstrates robust real-world teleoperation under external disturbances.

Website: https://roboparty.github.io/UFO
Code: https://github.com/Roboparty/UFO
Last Updated: July 14, 2026

Abstract: Humanoid behavioral foundation models (BFMs) based on unsupervised reinforcement learning enable reusable whole-body skills without task-specific rewards, but existing infrastructures suffer from slow training, limited algorithm support, poor extensibility, and insufficient real-world validation. We present **UFO**, an open-source framework for unsupervised reinforcement learning in humanoid control. **UFO** (1) improves the training efficiency of Forward-Backward learning, reducing training time from three days to 6-8 hours while achieving better performance; (2) integrates **TeCH**, a contrastive temporal-distance representation learning algorithm, demonstrating support for unsupervised RL algorithms beyond FB; (3) provides a general and extensible framework supporting diverse robots and multi-dataset training, enabling high-dynamic skill injection; and (4) enables robust real-world teleoperation and disturbance recovery by learning a broad, connected behavior space through unsupervised exploration. We hope **UFO** serves as an open, reproducible research platform that accelerates the development of BFMs and unsupervised RL for humanoid control.

Contents

1 Introduction

Humanoid robots serve as ideal physical platforms for general AI due to their natural compatibility with human environments. Their high degrees of freedom support flexible whole-body loco-manipulation, yet complicate scalable motor learning. Traditional methods using handcrafted rewards perform well only on limited tasks and lack robustness and generalization in complex scenarios. To address this, researchers shift toward learning general behavioral priors to obtain reusable low-level motor skills, instead of optimizing for separate tasks.

In recent years, reinforcement learning has advanced sim-to-real motor skill transfer for humanoid robots. Centered on whole-body motion tracking, these methods enable robots to learn dynamic, reusable skills for locomotion and loco-manipulation tasks. Most existing approaches formulate whole-body control as motion tracking yet this tracking-centric paradigm introduces two key limitations. First, they demand massive high-dimensional motion data and large-scale parallel environments, leading to prohibitive training costs. For example, SONIC (Luo et al., 2025) requires 128 GPUs for three consecutive days of training. Second, constrained dataset coverage fails to span the full control space of high-degree-of-freedom humanoids. Lacking structured and smooth representations, existing methods limit skill generalization and reuse, and cannot efficiently generate robust responses against external disturbances.

To address these limitations, recent works have explored off-policy unsupervised reinforcement learning for humanoid control. Unlike tracking-centric on-policy methods, this paradigm can efficiently reuse experience through large-scale replay buffers and acquire reusable behaviors without task-specific rewards. A representative approach is Forward-Backward (FB) representation learning (Touati and Ollivier, 2021a), which factorizes long-horizon behaviors into structured latent representations, enabling unsupervised skill discovery and composition. Introduced to humanoid control by BFM-Zero (Li et al., 2025), FB learning demonstrated the potential of off-policy exploration to learn smooth and composable behavior spaces, achieving strong skill interpolation, robustness, and generalization.

Despite these promising results, several challenges remain before unsupervised reinforcement learning infrastructure (Li et al., 2025) can serve as a practical foundation for humanoid control.

- **Limited Training Efficiency and High Memory Requirements.** Existing behavioral foundation models (Li et al., 2025) require a high-memory GPU and approximately three days of training in Isaac Lab, limiting research efficiency and reproducibility.
- **Limited Algorithmic Coverage.** Existing work mainly focuses on FB learning for humanoid control, while the potential of other unsupervised RL paradigms remains largely unexplored.
- **Limited Extensibility across Robots and Skills.** Current implementations (Li et al., 2025) are tightly coupled to specific robot platforms and datasets, making adaptation to new morphologies and diverse skills challenging.
- **Insufficient Real-World Validation.** Existing infrastructure lacks comprehensive real-world validation, particularly for teleoperation under external disturbances, leaving practical deployment capabilities underexplored.

In this work, we present **UFO**, an open-source framework for unsupervised reinforcement learning in humanoid control. Our main contributions are summarized as follows:

- **Reduced Training Time with Better Performance.** We implement the main algorithm of BFM-Zero (Li et al., 2025) in **UFO** (**UFO-FB**), which completes training in less than 12 hours using eight RTX 4090 GPUs, eliminating the dependence on high-memory GPUs while consistently outperforming BFM-Zero.
- **New Unsupervised RL Algorithm.** We introduce a novel unsupervised RL algorithm based on contrastive temporal-distance representation, **TeCH**, and seamlessly integrate it into **UFO**. The resulting **UFO-TeCH** achieves strong performance across simulation and real-world experiments, demonstrating the flexibility of **UFO** and that humanoid whole-body control can be built upon unsupervised RL algorithms beyond Forward-Backward learning.
- **General and Extensible Framework.** **UFO** supports flexible robot configurations and arbitrary motion datasets with given injection strategy, enabling seamless adaptation to diverse humanoid platforms and agile behaviors such as cartwheels.
- **Robust Real-world Teleoperation.** **UFO** supports teleoperation and demonstrates robust policy execution under external disturbances, validating the practicality of unsupervised behavioral foundation models for real-world humanoid applications.

Compared with prior unsupervised learning infrastructures, **UFO** is more reproducible and extensible while substantially broadening the scope of unsupervised reinforcement learning algorithms for humanoid control. We validate **UFO** across agile motion learning, teleoperation, and disturbance recovery, demonstrating the effectiveness and practicality of unsupervised behavioral foundation models. In the following sections, we present the technical details of these four contributions and demonstrate their effectiveness through experimental evaluation.

2 Related Work

2.1 Behavioral Foundation Models for Humanoid Control

Recent advances in humanoid control have shifted from task-specific policy learning toward behavioral foundation models that aim to acquire reusable whole-body skills from large-scale motion data. Existing approaches typically learn motion tracking policies in simulation and transfer the learned behaviors to downstream tasks such as locomotion, manipulation, teleoperation, and motion generation (Peng et al., 2018; Luo et al., 2023, 2024; Tessler et al., 2024; Serifi et al., 2024; He et al., 2024b, 2025a; Zhang et al., 2025a; Zeng et al., 2025; Yin et al., 2025; Chen et al., 2025; He et al., 2024a, 2025b; Cheng et al., 2024; Dugar et al., 2025; Fu et al., 2024).

Most behavioral foundation models rely on imitation learning together with on-policy reinforcement learning, where policies are optimized using reference motions or task-specific objectives (He et al., 2024b; Yin et al., 2025; Zeng et al., 2025; Chen et al., 2025; Zhang et al., 2025a; Liao et al., 2025). While these methods achieve impressive tracking performance, they require large-scale motion datasets and computationally expensive training, and their learned behaviors remain strongly tied to the training distribution. In contrast, our work explores unsupervised reinforcement learning as an alternative paradigm for learning reusable humanoid behavioral priors.

2.2 Unsupervised Reinforcement Learning

Unsupervised reinforcement learning (URL) aims to learn diverse and reusable behaviors without task-specific rewards. Early skill discovery methods, including VIC (Gregor et al., 2016), DIAYN (Eysenbach et al., 2018), and VALOR (Kim et al., 2023), maximize mutual information between latent skills and trajectories to encourage behavior diversity. Other approaches employ intrinsic motivation (Pathak et al., 2017; Burda et al., 2018; Lee et al., 2020) or contrastive representation learning (Yarats et al., 2021; Liu and Abbeel, 2022) to improve exploration and representation quality. Although successful on low-dimensional control tasks, these methods have seen limited adoption in high-degree-of-freedom humanoid systems.

Recently, Forward-Backward (FB) representation learning (Touati and Ollivier, 2021b) was introduced to humanoid control through BFM-Zero (Li et al., 2025), demonstrating that off-policy unsupervised reinforcement learning can learn reusable behavioral foundation models with strong skill interpolation and generalization. More recently, Temporal Distance Representation (TLDR) (Bae et al., 2024) proposed a contrastive representation learning framework that models temporal reachability instead of successor features, providing a simpler objective and dense progress-aware rewards for goal-conditioned reinforcement learning. These advances suggest that humanoid behavioral foundation models can be built upon unsupervised reinforcement learning algorithms beyond FB.

Despite these promising developments, follow-up work remains scarce. Existing implementations are largely limited to a single representation learning algorithm and are tightly coupled to specific robot platforms, making reproduction, comparison, and extension difficult (Li et al., 2025). Our work addresses this gap by providing a unified, open-source framework for unsupervised reinforcement learning in humanoid control. Built upon this framework, we substantially improve the training efficiency of FB learning and integrate **TeCH**, our humanoid adaptation of the TLDR algorithm (Bae et al., 2024), demonstrating that the framework readily supports diverse unsupervised reinforcement learning algorithms beyond Forward-Backward learning.

3 Problem Formulation

We consider the problem of *unsupervised reinforcement learning* for humanoid control, where the objective is to pretrain a general-purpose control policy from large-scale unlabeled motion data, without relying on task-specific reward functions or manually designed behaviors. During pretraining, the agent learns reusable motion representations and control skills through self-exploration while leveraging motion demonstrations, enabling zero-shot execution of diverse downstream tasks or serving as a skill-conditioned policy.

We formulate the real-world humanoid robot control problem as a Partially Observable Markov Decision Process (POMDP), defined as a five-tuple (S, O, A, P, γ), where S denotes the full state space, O the observation space, A the action space, $P(s_{t+1} \mid s_t, a_t)$ the state transition dynamics, and $\gamma \in (0, 1)$ the discount factor. For a humanoid robot with 29 degrees of freedom, the action $a \in A \subset \mathbb{R}^{29}$ consists of target values for a proportional-derivative (PD) controller over all degrees of freedom. The privileged state information $s \in \mathbb{R}^{463}$ includes root height, body pose, body orientation, linear velocity, and angular velocity. The observable state is defined as $o_t = \{q_t - \bar{q}, \dot{q}_t, \omega_t^{\text{root}}/4, g_t\} \in \mathbb{R}^{64}$, where $q_t \in \mathbb{R}^{29}$ denotes joint positions normalized with respect to the nominal pose $\bar{q}$, $\dot{q}_t \in \mathbb{R}^{29}$ denotes joint velocities, $\omega_t^{\text{root}} \in \mathbb{R}^3$ denotes root angular velocity, and $g_t \in \mathbb{R}^3$ denotes the projected gravity vector in the root frame. We further define the observation history as $o_{t,H} = \{o_{t-H}, a_{t-H}, \ldots, o_t\} \in \mathbb{R}^{93 \cdot H + 64}$, which consists of proprioceptive

states and actions. Except for root height, all state components are normalized with respect to the current heading direction and root position. During the pretraining stage, the agent has access to an unlabeled motion dataset $\mathcal{M} = \{\tau\}$, where each trajectory contains both observation and privileged state sequences, i.e.,$\tau = (o_1, s_1, \ldots, o_{l(\tau)}, s_{l(\tau)})$, where $l(\tau)$ denotes the length of trajectory τ.

4 Reduced Training Time with Better Performance

Existing humanoid unsupervised reinforcement learning frameworks typically require several days of training to obtain a deployable policy. We redesign both the training infrastructure and optimization pipeline to significantly reduce wall-clock training time while simultaneously improving policy performance.

4.1 How to Reduce Training Time?

We improve training efficiency from both the simulation and distributed training perspectives.

Lightweight simulation backend. We adopt mjlab (Zakka et al., 2026) as our simulation backbone, which provides significantly higher simulation throughput than conventional simulators while maintaining accurate humanoid dynamics. This is particularly important for unsupervised RL, where broad state-space exploration requires visiting substantially more diverse interaction data than learning a single tracking task.

Synchronous distributed training. We implement a custom synchronous data-parallel training framework based on `torchrunx` and `torch.distributed`. Each GPU maintains an independent set of simulation environments and replay buffer, while model gradients are synchronized through manual all-reduce averaging after every optimization step. This design enables efficient multi-GPU scaling with minimal communication overhead. This design scales both behavioral exploration and off-policy experience collection across GPUs.

4.2 How to Achieve Better Performance?

The improved training throughput allows us to substantially scale up reinforcement learning.

Large-scale experience collection. We increase the number of parallel simulation environments, replay buffer capacity, and optimization batch size, enabling the policy to learn from more diverse experience and improving optimization stability. This improved behavioral coverage is essential for learning reusable latent representations.

Hyperparameter optimization. We further refine the training hyperparameters for the large-scale distributed setting, resulting in faster convergence and improved final tracking performance. These hyperparameters directly affect experience reuse and the policy's responsiveness to changing latent goals.

4.3 Simulation Results

4.3.1 Evaluation Metrics

We evaluate our method on three tasks: motion tracking, goal reaching, and reward optimization.

For motion tracking, following prior work, we evaluate on the LAFAN1 dataset using two metrics: (1) Earth Mover's Distance (EMD) (Rubner et al., 2000), which measures the similarity between the generated and reference motion distributions (lower is better), and (2) joint mean absolute error (Joint MAE), which measures the average joint configuration error with respect to the reference motion:

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

where e denotes an episode, m is the target motion trajectory, and $q(\cdot)$ represents the 29-dimensional joint configuration. Results are averaged over all motion clips.

For goal reaching, we manually extract 21 stable poses from the LAFAN1 training set, where a stable pose is defined as a state with zero velocity. We measure the average joint error between the agent and the target pose:

$$E_{\text{mae}}(e, g) = \frac{1}{|e|} \sum_{t=1}^{|e|} \|q_t(e) - q(g)\|, \tag{2}$$

where g denotes the target pose. Results are averaged over all target poses. Each evaluation episode has a fixed horizon of $H = 500$.

Reward evaluation is performed on 24 predefined reward functions. For each reward, the policy is rolled out from the default initial pose for 500 environment steps. At every step, we recompute the corresponding normalized task reward from the next state. The score for each reward is the cumulative normalized return over the 500-step episode, and the final reward score is the average of this return across all 24 rewards.

4.3.2 Training Efficiency

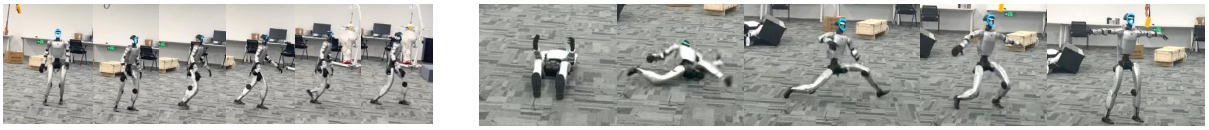
Table 1: Training efficiency comparison. Training time is measured as the wall-clock time required to obtain a policy suitable for real-world deployment, beyond which further training yields no significant performance improvement. Throughput is reported as total simulation frames per second (FPS).

Method	Hardware	Num. of Env.	Training Time	Throughput (↑)
BFM-Zero	H200	1024	45h	763.6 FPS
UFO-FB	H200	1024	21h	993.4 FPS
UFO-FB	8×H200	8×1024	6h	4824.7 FPS
UFO-FB	8×RTX 4090	8×512	14h	1284.0 FPS

Table 1 compares the training efficiency of different hardware configurations. Following prior work, we measure training time as the wall-clock time required to obtain a policy suitable for real-world deployment, beyond which further training yields no significant performance improvement. This interval is illustrated in Figure 3, where the shaded region denotes a stable performance plateau with consistently successful real-world deployment. Under the same single-GPU setting

(H200, 1024 environments), our implementation achieves approximately 1.3× higher simulation throughput than the original BFM-Zero infrastructure. This improvement mainly comes from the lightweight simulation pipeline in mjlab (Zakka et al., 2026). Furthermore, our distributed training implementation scales efficiently across multiple GPUs, reducing the training time to **only 6 hours** on eight H200 GPUs. Importantly, **UFO** also trains efficiently on commodity hardware: using eight RTX 4090 GPUs, it produces a sim-to-real deployable humanoid policy within approximately 12-14 hours, removing the dependence on expensive high-memory GPUs.

Unless otherwise specified, we use the 8×H200 configuration as the default setting for all subsequent experiments. Notably, the policies used in the following experiments were trained for only 3-4 hours, yet already achieved sufficiently strong performance (Figure 2).



(a) Motion Tracking (b) Goal Reaching

Figure 2: Real-world deployment of policies trained for only 3-4 hours using the default 8×H200 configuration. Despite the short training time, the learned policies already exhibit robust whole-body behaviors across diverse scenarios.

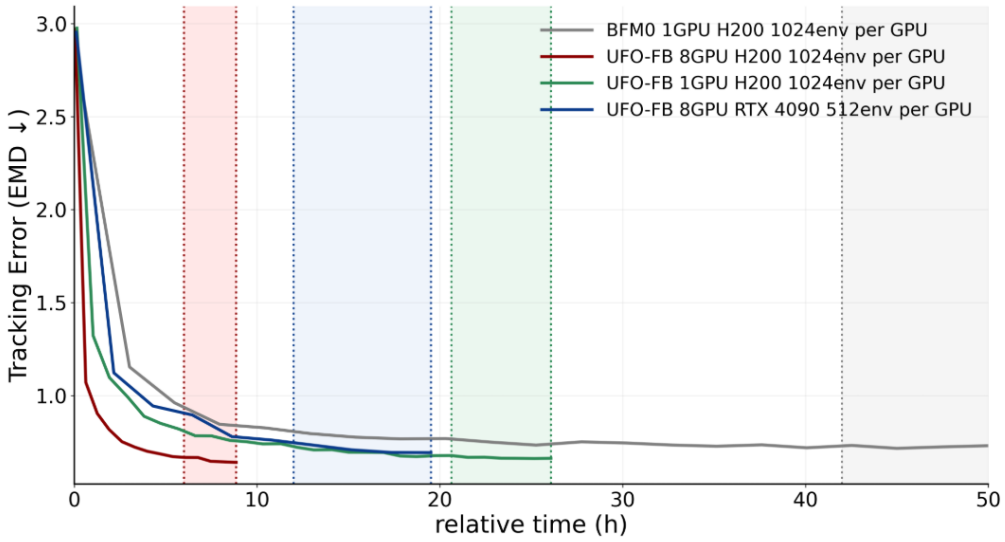


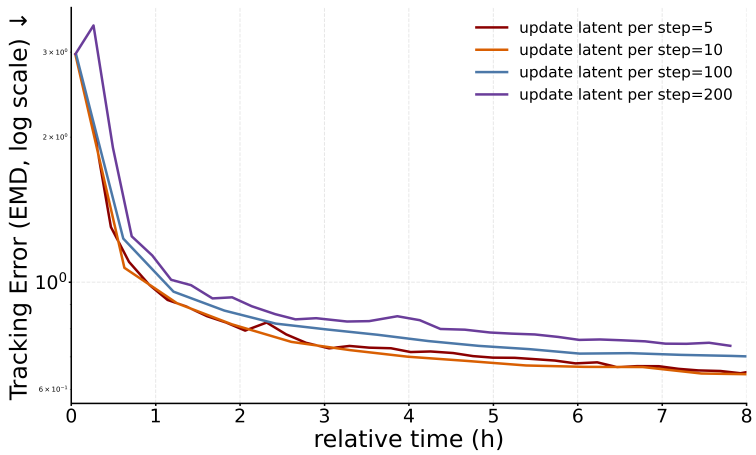
Figure 3: Training efficiency and effects comparison.

4.3.3 Policy Performance

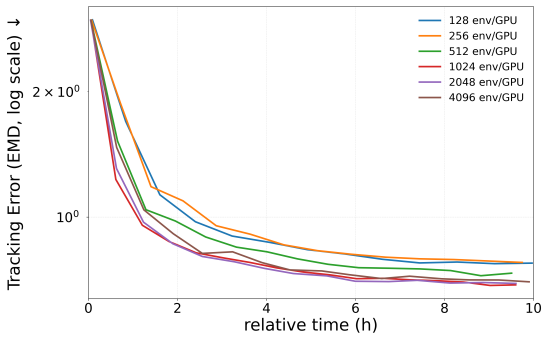
Table 2: Policy performance comparison. The first two metrics are tracking metrics.

Method	Hardware	Num. of Env.	EMD (↓)	$E_{\text{mae}}^{\text{track}}$ (↓)	$E_{\text{mae}}^{\text{goal}}$ (↓)	Return (↑)
BFM-Zero	H200	1024	0.731	0.1510	0.139	207.3
UFO-FB	H200	1024	0.668	0.1266	0.130	243.0
UFO-FB	8×H200	8×1024	0.639	0.1178	0.115	240.8
UFO-FB	8×RTX 4090	8×512	0.693	0.1254	0.124	232.2

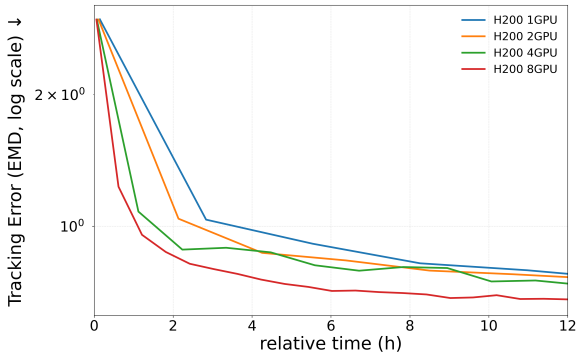
Table 2 reports the quantitative evaluation results. Besides substantially improving training



(a) Update latent frequencies ablation on 8×H200 (log-EMD).



(b) Environment scaling on 8×H200 (log-EMD).



(c) GPU scaling on H200 (log-EMD).

Figure 4: Log-scale tracking error (EMD) versus relative training time under environment scaling and GPU scaling settings.

efficiency, **UFO** consistently achieves better policy performance than the released BFM-Zero model. Even under the same single-GPU configuration, our implementation produces lower tracking error. With distributed training on eight H200 GPUs, **UFO** achieves the best overall performance, reducing the Earth Mover's Distance (EMD) from 0.731 to 0.639 while also improving the joint tracking error. Notably, the policy trained on eight RTX 4090 GPUs still achieves an EMD below 0.7, outperforming the released BFM-Zero model despite using commodity GPUs. These results demonstrate that our framework improves both training efficiency and final policy quality without requiring specialized hardware.

4.4 Ablation Studies

We conduct ablation studies to validate the effectiveness of our design choices.

Latent update frequency. We compare latent update frequencies (Figure 4a) of 200, 100, 10, and 5, and observe that tracking accuracy consistently improves as `update-z` decreases. This trend is expected: a smaller `update-z` means the policy updates its motion-conditioning representation more frequently, enabling faster responses to reference trajectory changes and reducing tracking errors caused by temporal compression and state lag.

At the same time, however, the compression ratio becomes lower, and the resulting motions are

generally less smooth and less soft than those under highly compressed settings. Moreover, in real-world deployment, an overly small `update-z` increases control difficulty and is more likely to cause stepping adjustments and jitter. Therefore, we typically adopt `update-z` $= 10 \sim 100$ as a practical trade-off between tracking accuracy and deployment stability.

Environment scaling analysis. Figure 4b compares log-EMD trajectories under different per-GPU environment counts on the same 8×H200 setup, while keeping the per-GPU buffer size fixed. Increasing the total number of environments improves exploration efficiency. Within a fixed wall-clock budget, reaching a sufficiently large environment count is beneficial; however, gains become marginal beyond about 1024 env/GPU. This indicates that final tracking accuracy does not scale proportionally with environment count, which differs from the monotonic scaling trend often reported in supervised learning.

GPU scaling (including environment and buffer) analysis. Figure 4c shows the log-EMD trajectories under different GPU counts on H200, where total environment count and effective buffer capacity increase with the number of GPUs. Scaling from 1 to multiple GPUs substantially improves wall-clock convergence in the early and middle stages, indicating better training throughput and faster policy refinement.

5 New Unsupervised RL Algorithm: UFO-TeCH

Previous humanoid unsupervised reinforcement learning frameworks are typically built around a single representation learning algorithm. In contrast, **UFO** provides a unified infrastructure that supports multiple representation learning paradigms. As a demonstration, we adapt Temporal Distance Representation (TLDR) (Bae et al., 2024) to humanoid whole-body control and develop **TeCH** (Temporal Distance Modeling via Contrastive Representation Learning for Humanoid Whole-Body Control), a temporal-distance-based contrastive representation learning algorithm built upon the TLDR framework (Figure 5a). **UFO-TeCH** denotes the complete instantiation of **UFO** using **TeCH**, while reusing the same training, evaluation, and deployment pipeline. This clear separation between representation learning algorithms and system infrastructure highlights the extensibility of **UFO**. In the following, we first introduce **TeCH** and then present the training and evaluation of **UFO-TeCH**.

5.1 Overview of TeCH

TeCH comprise two core components: an unsupervised pre-training phase and a zero-shot inference pipeline (Section 5.5).

For the pretraining (Section 5.4) of **TeCH**, the learning objectives are twofold: 1) a latent representation space that captures the temporal structure of behavior trajectories. In this space, states or trajectories with temporal proximity and similar behavior patterns are mapped to adjacent positions, empowering the policy to generalize based on trajectory-level behavior semantics (Section 5.2); 2) a policy conditioned on latent task representations. We can map macroscopic robot poses into this structured space to achieve unified control over diverse motion targets and behavior patterns with goal progress reward (Section 5.3).

To achieve zero-shot sim-to-real transfer of all unsupervised RL policy, **TeCH** in **UFO** follow (Li

et al., 2025) introduce an adversarial objective during training to align humanoid robot behaviors with the human data distribution and add auxiliary training objectives to enforce critical constraints for sim-to-real transfer.

5.2 Temporal Distance Representation Learning

Formally, given a state transition (s_t, s_{t+1}), we construct pseudo-goals by sampling future states:

$$g_t = \text{FutureSample}(s_{t+1}). \tag{3}$$

By sampling temporally distant future states as pseudo-goals, the learned representation is supervised beyond one-step transitions and encouraged to capture long-horizon temporal dependencies. Consequently, the latent space models multi-step environmental dynamics rather than merely fitting instantaneous state transitions.

We then map both observed states and pseudo-goals into a shared latent space through a shared encoder:

$$\phi_x = \phi_\psi(s_t), \quad \phi_y = \phi_\psi(s_{t+1}), \quad \phi_g = \phi_\psi(g_t). \tag{4}$$

The encoder is optimized with a dual-objective contrastive loss:

$$\mathcal{L}_{\text{TE}}(\psi, \lambda) = -\mathcal{L}_{\text{dist}}(\phi_x, \phi_g) + \lambda\, \mathcal{L}_{\text{step}}(\phi_x, \phi_y), \tag{5}$$

where the contrastive loss $\mathcal{L}_{\text{dist}}$ explicitly separates temporally distant state pairs in the latent space, encouraging the encoder to distinguish states according to their temporal reachability. Meanwhile, the smoothness loss $\mathcal{L}_{\text{step}}$ regularizes consecutive states to preserve locally continuous latent dynamics and prevent abrupt feature changes. Jointly optimizing these two objectives produces a temporally structured latent geometry, in which latent distances faithfully reflect multi-step transition relationships and the temporal reachability of real robot trajectories.

5.3 Goal Progress Reward with Temporal Distance

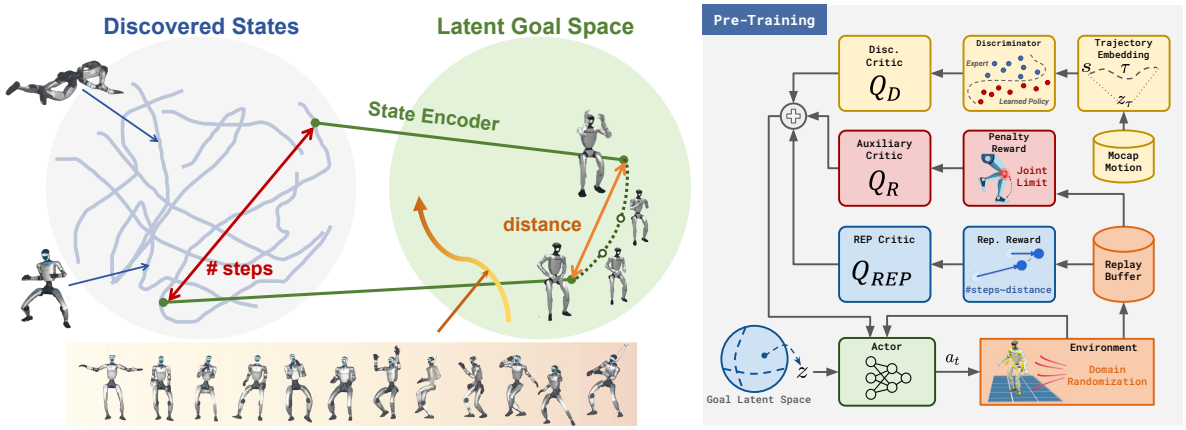
For the policy π_θ, the key objective is to measure progress toward a latent goal z. Let $\bar{z}(\cdot)$ denote the projected latent representation. The reward is defined as:

$$r_{\text{TeCH}}(s_t, s_{t+1}, z) = \|z - \bar{z}(s_t)\|_2 - \|z - \bar{z}(s_{t+1})\|_2. \tag{6}$$

This formulation has a clear intuition: if the next state is closer to the goal in the latent space, the reward is **positive**; otherwise, it is negative. Consequently, the policy is continuously driven to optimize toward the direction of goal proximity, resulting in a dense and directionally informative learning signal.

5.4 Pre-training Pipeline of UFO-TeCH

As illustrated in Figure 5b, the pretraining pipeline consists of three iterative modules: *online interaction*, *experience replay*, and *multi-objective optimization*. These components form a closed training loop that continuously operates across parallel simulation environments.



(a) Overview of the proposed framework. (b) Detailed pre-training pipeline.

Figure 5: (a) Overview of the proposed framework. (b) The pre-training pipeline employs three parallel critics to jointly optimize the policy: a discriminative critic for humanoid motion regularization, an auxiliary critic providing sim-to-real transfer reward, and a representation critic that regularizes state embeddings to align with the latent goal.

During online interaction, the policy executes actions conditioned on a latent variable z and collects transitions of the form $(s_t, o_{t-k:t}, a_t, s_{t+1}, z_t)$, which are stored in an online replay buffer. Importantly, the latent variable z is not drawn from a single source but is instead constructed from a mixture of three complementary signals, jointly balancing exploration diversity, goal reachability, and behavioral priors:

1. Randomly sampled latent variables (e.g., from Gaussian or hyperspherical priors), which expand the coverage of the latent task space and encourage behavioral diversity;
2. Latent representations associated with online samples, which align learning with goals that are currently reachable by the policy;
3. Latent representations derived from expert trajectories, which inject human-motion priors and provide stable anchors in the latent space.

During the **experience replay** stage, the learner simultaneously samples transition batches from the online replay buffer and trajectory segments from unlabeled expert demonstrations. These samples are used to construct the discriminator learning signal and the representation learning objective, respectively.

The training then proceeds to the **multi-objective optimization** stage, where parameters are updated sequentially according to Discriminator $\rightarrow$ Encoder(ϕ_ψ) $\rightarrow$ Main Critic and Auxiliary Critic $\rightarrow$ Actor $\rightarrow$ Target Network Soft Update. This closed-loop procedure is repeated until convergence.

From an optimization perspective, the Actor is optimized by three complementary Q-functions: the representation Q-function $\boldsymbol{Q_{\mathbf{REP}}}$, the style Q-function $\boldsymbol{Q_D}$, and the auxiliary Q-function $\boldsymbol{Q_R}$. Here, $\boldsymbol{Q_{\mathbf{REP}}}$ is learned from the TeCH progress reward r_{TeCH}, while the temporal encoder is optimized separately by $\mathcal{L}_{\text{TE}}$ to learn the latent representation that defines this reward. The resulting Actor objective is

$$\mathcal{L}(\pi) = -\mathbb{E}_{\substack{(o_{t,H}, s_t)\sim\mathcal{D} \\ a_t=\pi(o_{t,H},z),\, z\sim\nu}} \Big[\lambda_{\text{REP}} \boldsymbol{Q_{\mathbf{REP}}} + \lambda_D \boldsymbol{Q_D} + \lambda_R \boldsymbol{Q_R}\Big]. \tag{7}$$

5.5 Zero-shot Inference in UFO-TeCH

During the testing stage, **UFO-TeCH** can solve a variety of tasks in a *zero-shot* manner, without any additional task-specific learning, planning, or fine-tuning. We first embed different downstream tasks into the latent space, and consider two representative types of tasks.

- **Goal Reaching**: Given discontinuous goal states, we map them into multiple policy conditions z, enabling the robot to switch between different target poses.
- **Trajectory Tracking**: We aggregate future states within a reference trajectory window and map them into a sequence of latent conditions $\{z_t\}$, allowing the policy to progressively track the target motion.

UFO-TeCH does not belong to the successor-feature-based unsupervised reinforcement learning paradigm and therefore does not provide a unified value-based inference interface. As a result, it typically cannot directly support reward optimization tasks such as those in BFM-Zero (Li et al., 2025). In contrast, **UFO-TeCH** introduces temporal-distance-aware representations that explicitly encode the temporal reachability relationship with respect to the current state. Consequently, in zero-shot tracking scenarios, **UFO-TeCH** often exhibits faster and tighter trajectory tracking behavior.

Goal Reaching. For the goal reaching task, the corresponding latent vector is formulated as:

$$z_g = \phi(s_g), \tag{8}$$

where $\phi(\cdot)$ denotes the temporal distance encoder of **UFO-TeCH**.

Tracking. For motion trajectory tracking, given a trajectory $\tau = \{s_1, \ldots, s_n\}$, the sequence of latent variables $\{z_t\}$ for the policy is defined as:

$$z_t = \sum_{t'=t}^{t+H} w_{t'}\, \phi(s_{t'}), \tag{9}$$

where H represents the look-ahead time window, and $w_{t'}$ are discount weights satisfying $w_{t'} \geq 0$ and $\sum_{t'=t}^{t+H} w_{t'} = 1$.

5.6 Simulation Results of UFO-TeCH

In this section, we evaluate **UFO-TeCH** comprehensively in simulation. The simulation frequency is set to 200 Hz, while the control frequency is 50 Hz. For the motion dataset, we use the LAFAN1 dataset (Harvey et al., 2020), which has been retargeted to the Unitree G1 (Unitree, 2025) robot and contains 40 motion clips of several minutes each. Importantly, accuracy alone does not fully reflect the advantages of **UFO-TeCH**. As demonstrated in the following experiments, **UFO-TeCH** further enables quick and direct discontinuous goal reaching, faster and robust recovery from falls and robust zero-shot execution from arbitrary initial states.

5.6.1 Motion Tracking

This evaluation measures the imitation accuracy of motion sequences. We evaluate the policies on both the training dataset (LAFAN1 (Harvey et al., 2020), 40 clips, approximately 2–3 hours

in total) and the test dataset (100-Style (Mason, 2023), 3–4k clips, approximately 18–19 hours in total), where all motions are retargeted to the G1 robot. We use joint position error (E_{mae}) as the evaluation metric.

Unlike prior frameworks such as SONIC (Luo et al., 2025), which are specifically designed for motion tracking and rely on task-specific tracking rewards, BFM-Zero and **UFO** (**UFO**-FB and **UFO**-**TeCH**) instead explore unsupervised reinforcement learning for humanoid control. They learn a unified latent space of reusable skills, where motion tracking emerges as just one downstream application alongside capabilities such as discontinuous goal reaching and hierarchical planning.

Nevertheless, we compare against the state-of-the-art general-purpose motion tracking controller SONIC Luo et al. (2025). We evaluate the publicly released model under the same environment and test sets, reporting E_{mae} and the standard deviation across motion clips. For fairness, BFM-Zero, **UFO**, and SONIC Luo et al. (2025) are evaluated without early termination, i.e., metrics are computed over full trajectories. Since SONIC typically struggles to recover quickly after severe failures or falls, we additionally report SONIC (TER.), where metrics are computed only over segments before termination.

Table 3: Motion tracking error for **UFO** and baselines (mean ± std).

Method	Motion Tracking Error (E_{mae}) (↓)	
	LAFAN1	100-Style
SONIC[b]	0.2916 ± 0.1887	0.1522 ± 0.1049
SONIC (TER.)[a]	0.1081 ± 0.0213	0.1355 ± 0.0351
BFM-Zero[b]	0.1510 ± 0.0255	0.1674 ± 0.0459
UFO-FB[b]	0.1178 ± 0.0575	0.1278 ± 0.0403
UFO-**TeCH**[b]	0.1272 ± 0.0582	0.1385 ± 0.0392

[a] SONIC (TER.) is evaluated only on segments before termination.
[b] SONIC, BFM-Zero, and **UFO** are evaluated on full trajectories.

From the tracking results, we observe that SONIC Luo et al. (2025) performs poorly in recovery after falls and during ground transitions, which substantially degrades its overall tracking performance on the test set. In contrast, BFM-Zero and **UFO** consistently recover balance and resume stable tracking even under fully out-of-distribution conditions, such as external perturbations, falls, and random initializations. We attribute this advantage to off-policy unsupervised exploration: unsupervised RL frameworks explore a much broader distribution of non-standard states and dynamic transitions during training, resulting in a smoother, more continuous, and more robust skill space. As further demonstrated in our real-world experiments, **UFO** also recovers robustly from strong physical disturbances and falls.

Even in terms of tracking accuracy alone, **UFO** achieve performance comparable to SONIC (TER.), as shown in Table 3. Notably, SONIC already outperforms GMT Chen et al. (2025), Any2Track Zhang et al. (2025b), and BeyondMimic Liao et al. (2025) in its original evaluation. We also emphasize the significant difference in training efficiency: SONIC Luo et al. (2025) uses 128 GPUs for 7 days, along with large-scale motion datasets and massive parallel environments for training. In contrast, our method is trained on only 8 GPUs within several hours using a small amount of LAFAN1 data, substantially reducing GPU-hours and environment samples.

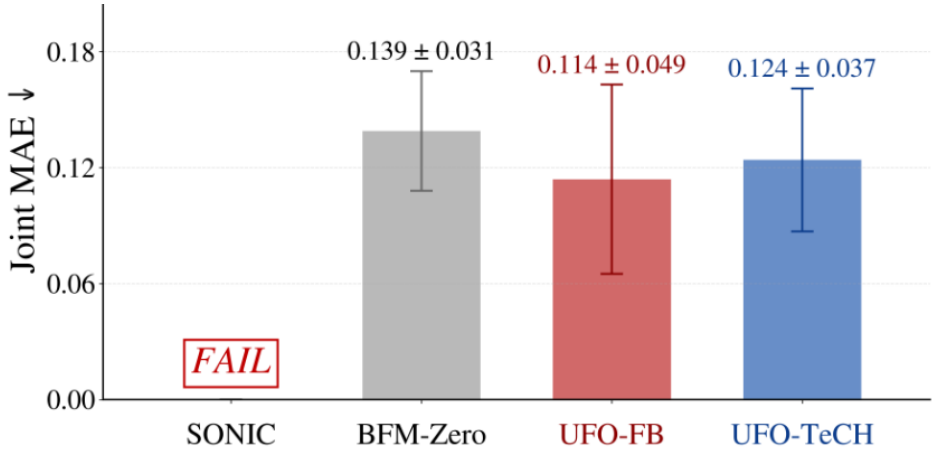


Figure 6: Comparison of joint MAE across methods. Lower is better. SONIC failed to produce valid results (FAIL), while BFM-Zero, **UFO**-FB and **UFO-TeCH** achieved achieved smooth and accurate transitions.

5.6.2 Goal Reaching

Figure 6 compares the joint-level mean absolute error between BFM-Zero, **UFO-FB** and **UFO-TeCH** on the goal-reaching task. The results show that both methods achieve comparable accuracy, while **UFO-TeCH** reaches the target more efficiently with a slight degradation in precision. In contrast, tracking-based methods such as SONIC (Luo et al., 2025) struggle with discontinuous goal transitions, highlighting the limitation of purely tracking-driven objectives for discontinuous goal-reaching.

5.7 Real-world Validation.

As shown in Figures 7 and Figure 8, **UFO-TeCH** achieves strong real-world performance in both motion tracking and goal reaching, with only minor degradation from simulation while maintaining stable and natural behaviors. As shown in Figure 9, **UFO-TeCH** also demonstrates strong robustness to external disturbances. When large pushes cause the robot to fall, it can autonomously stand up and resume the target behavior, similar to BFM-Zero and **UFO**-FB, whereas the imitation-based baseline SONIC (Luo et al., 2025) fails to recover. This robustness stems from the unified latent goal space learned during unsupervised pre-training, which naturally encodes recovery behaviors without dedicated fall-recovery training.

Compared with **UFO**-FB, **UFO-TeCH** recovers from falls more rapidly, enabling the robot to stand up and regain stable motion in fewer transition steps. This behavior results from the temporal-distance objective, which explicitly optimizes representations for efficient state transitions. As a trade-off, the recovered motions are sometimes less human-like due to the emphasis on recovery efficiency.

6 General and Extensible Framework

Unlike humanoid reinforcement learning systems whose implementation is tied to a specific robot model, a fixed motion dataset, or a closed set of behaviors, **UFO** is designed as a general and extensible framework along two complementary axes: robot embodiment and skill coverage.

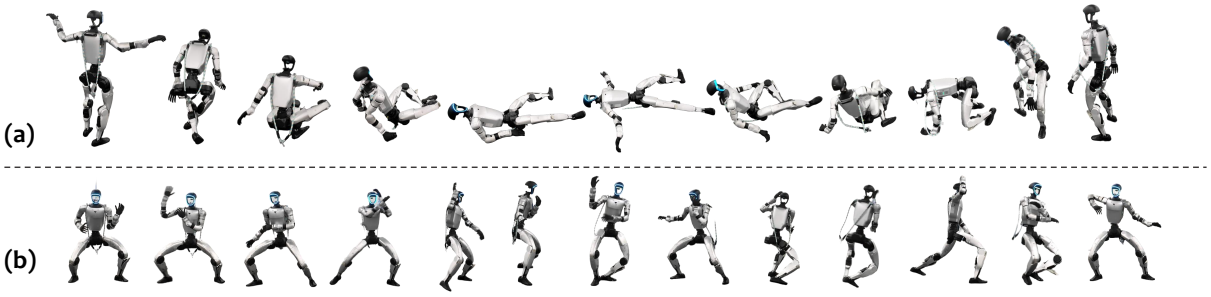


Figure 7: Motion tracking performance. **UFO-TeCH** accurately tracks diverse reference motions while producing smooth and physically plausible whole-body behaviors.

Figure 8: Goal reaching performance. **UFO-TeCH** enables flexible goal reaching by leveraging temporal-distance representations, allowing the policy to reliably reach target states from arbitrary initial configurations.

At the embodiment level, **UFO** separates the learning algorithm from robot-specific assumptions by exposing explicit interfaces for robot morphology, motion data, training configuration, and inference/export. This allows the same training code to be instantiated across humanoid robots with different kinematic structures and action dimensions. At the behavioral level, **UFO** supports stable skill expansion by allowing new motion datasets to be injected into the training distribution without overwriting the diversity of the original foundation dataset. Together, these two design choices allow **UFO** to adapt not only to new humanoid platforms, but also to new behaviors that are rare or absent in the original motion corpus.

6.1 How to Adapt to Multiple Robots?

Many humanoid reinforcement learning pipelines are coupled to a specific robot morphology, with robot-dependent assumptions embedded in joint ordering, action dimension, motion representation, reward computation, and inference code. **UFO** decouples these assumptions from the learning algorithm through a unified robot configuration interface. A new robot is specified by an explicit configuration file that defines its actuated joints, body semantics, actuator limits, default state, and contact structures, allowing the same training code to be instantiated across robots with different kinematic structures and action dimensions

To reduce bring-up effort, **UFO** provides an XML-assisted configuration pipeline that parses a MuJoCo model and generates draft RobotTrainingSpec and Hydra configuration files. When URDF information is available, the draft can be enriched with actuator limits, joint dynamics, and semantic hints. Motion data are introduced through a standardized RobotState interface: externally retargeted motions are imported as root pose and joint positions in the robot's XML/control-joint order, then converted into full-motion inference caches and clipped training caches through a shared data manifest.

We evaluate this interface on humanoid robots with different morphologies, including Unitree G1 (Unitree, 2025), Booster K1 (Booster, 2025), Unitree H1 (Unitree Robotics, 2023),

Figure 9: Robust recovery from disturbances. **UFO-TeCH** rapidly recovers from external perturbations and resumes the target behavior without additional training, highlighting the robustness of the learned representation.

RP1 (Roboparty, 2025), RPO (Roboparty, 2025), and AgiBot X2 (AGIBOT, 2025) (some of the simulation demos are shown in Figure 10). These robots differ in body layout, actuator count, joint ordering, and available motion data, allowing us to test whether **UFO** can reuse the same training and inference code after changing only the robot configuration and robot-specific motion data.

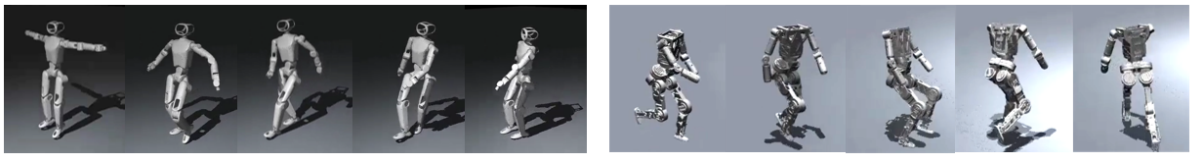


Figure 10: Simulation demonstrations on multiple humanoid robots with different morphologies. By changing only the robot configuration and robot-specific motion data, **UFO** reuses the same training and inference pipeline across all platforms.

6.2 How to Inject New Skills?

UFO also supports behavioral extensibility: new skills can be added without discarding the broad behavior distribution learned from the original foundation dataset. This is important for agile behaviors such as cartwheels, which are rare in large-scale motion corpora and are therefore difficult to learn under standard sampling.

A straightforward approach is to merge a skill-specific dataset with the original motion pool and apply prioritized sampling over all trajectories. However, this strategy makes the effective training distribution depend on motion difficulty. Because agile motions are rapidly changing and difficult to track, they tend to receive high sampling priorities throughout training. The policy is therefore repeatedly trained on a narrow set of difficult trajectories, which destabilizes actor-critic optimization. In our cartwheel experiments, merged-data baselines with different cartwheel proportions produced non-finite actor and Q-function metrics after approximately 14–28M environment steps. In a representative failed run, this divergence was preceded by severe physical instability, including invalid torso height and extremely large contact forces.

To avoid this failure mode, **UFO** uses fixed-ratio dataset mixing for skill injection. At each motion sampling step, the dataset identity is sampled from a fixed categorical distribution, and prioritized sampling is applied only within the selected dataset. This separates the inter-dataset sampling probability from the within-dataset motion priority. For cartwheel injection, we use a 0.95/0.05 mixture between the foundation dataset and the cartwheel dataset. This preserves exposure to the original diverse motion corpus while assigning a controlled amount of training probability to the new agile skill. With this strategy, **UFO** remains stable to 384M environment steps and learns cartwheel behaviors while maintaining the broader foundation behavior distribution.

Figure 11: Injecting new agile skills (cartwheel) into a behavior foundation model. Our fixed inter-dataset sampling ratio stabilizes training and enables successful skill acquisition, whereas naive merged prioritized sampling leads to instability.

7 Robust Real-World Teleoperation

7.1 Teleoperation System

To support real-world evaluation, **UFO** provides a modular teleoperation system that integrates motion capture, latent goal generation, policy inference, and robot control under a unified deployment framework. The operator wears a Pico VR headset together with two foot trackers to capture whole-body motion. The captured human pose is first retargeted to the humanoid kinematic model at 50 Hz and subsequently encoded into a latent goal representation through the learned representation encoder.

The deployment system adopts a distributed communication architecture. Motion retargeting (Ze et al., 2025), latent representation inference, policy execution, and robot control are implemented as independent modules and communicate through lightweight ZeroMQ interfaces. Depending on the deployment configuration, the representation encoder and policy can run either onboard the robot or on a remote workstation connected via Ethernet. The robot is controlled through the Unitree SDK2 interface with CycloneDDS communication. To ensure reliable real-world operation, the framework incorporates communication watchdogs, packet timeout detection, command validation, and emergency stopping mechanisms, enabling robust teleoperation under communication interruptions and external disturbances.

7.2 Real-world Validation

We conducted extensive and diverse real-world evaluations, covering a wide range of behaviors including locomotion, upper-body manipulation, squatting, half-squatting, kneeling, ground rolling, and rapid stand-up motions (Figure 12, Figure 13, Figure 14). We further demonstrated simple interaction tasks (Figure 15), such as taking a box from a human or kicking a box away.

More interestingly, as shown in Figure 16, **UFO** enables stable teleoperation even under strong external disturbances. **To the best of our knowledge, robust teleoperation under severe disturbances has not been demonstrated by previous open-source humanoid learning infrastructures.** During our evaluation, we conducted dozens of disturbance trials involving strong pushes, pulls, and kicks from multiple people. Despite these aggressive perturbations, **UFO** consistently maintained stable teleoperation or rapidly recovered after falls, whereas Mimic-Lite (EGalahad, 2025), TeleOpIt (BotRunner64, 2025), and SONIC (Luo et al., 2025) were unable to maintain stable teleoperation under such conditions or during rapid stand-up and fall-recovery behaviors.

Figure 12: Whole-body teleoperation-A. Teleoperated whole-body motions including diverse locomotion such as running and backward walking.

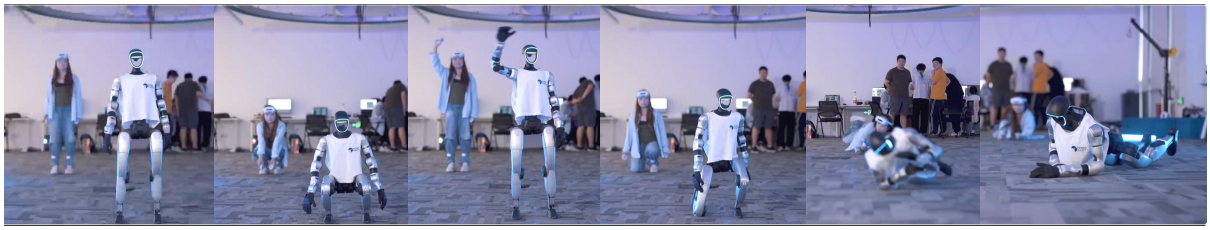


Figure 13: Whole-body teleoperation-B. Teleoperated whole-body motions including standing, squatting, waving hand, kneeling and rolling, demonstrating stable execution across diverse postures.

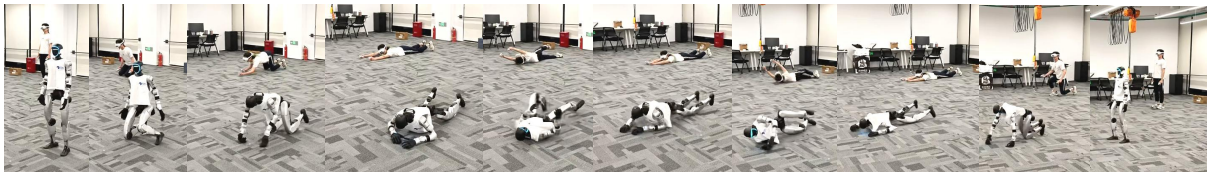


Figure 14: Whole-body teleoperation-C. Teleoperated ground rolling and rapid stand-up behaviors, demonstrating agile whole-body control and recovery.

Figure 15: Interactive teleoperation. Human-robot interaction tasks including hugging with human, receiving and kicking a box.

Figure 16: Robust teleoperation under disturbances. Stable teleoperation under repeated large external perturbations, including strong pushes, pulls, and kicks, with rapid recovery from falls.

It is worth noting that, compared with dedicated teleoperation frameworks, the teleoperation accuracy and motion quality of **UFO** still have room for improvement. In particular, the robot is less stable during locomotion, especially when walking backward or sideways, and its global tracking accuracy remains somewhat limited. We attribute these limitations primarily to three factors: (1) the current teleoperation pipeline incurs noticeable latency due to the large backward mapping network; (2) the pre-training dataset consists only of LAFAN1, which lacks sufficient locomotion and real-world interaction data; and (3) the fully unsupervised training paradigm precludes the use of task-specific termination conditions, such as terminating episodes when excessive root drift occurs. Incorporating more diverse locomotion and real-world demonstration data, together with future improvements to the teleoperation pipeline, is expected to further enhance teleoperation performance.

8 Conclusion

In this work, we presented **UFO**, an open-source framework for unsupervised reinforcement learning in humanoid control. Unlike previous infrastructures designed around a single algorithm and robot platform, **UFO** provides a unified, scalable, and extensible foundation that supports flexible robot configurations, arbitrary motion datasets, distributed multi-GPU training, and real-world deployment.

Using this framework, we significantly improved the efficiency of Forward-Backward representation learning, reducing training time from approximately three days to less than 12 hours on commodity GPUs while consistently achieving better policy performance. We further demonstrated the flexibility of **UFO** by integrating **TeCH**, a contrastive temporal-distance representation learning algorithm, showing that high-quality humanoid behavioral foundation models can be built upon unsupervised reinforcement learning algorithms beyond Forward-Backward learning. Finally, real-world teleoperation and disturbance recovery experiments validated the robustness and practicality of the learned policies.

We hope **UFO** serves as a reproducible research platform for the community and accelerates the development of scalable behavioral foundation models for humanoid robots. Future work will extend the framework to broader classes of unsupervised reinforcement learning algorithms, larger-scale motion datasets, and more diverse real-world humanoid tasks.

Acknowledgments

Robotparty Lab acknowledges the shared infrastructure and robot platform support behind this technical report.

References

AGIBOT. Agibot x2 humanoid robot. https://store.agibot.com/products/x2, 2025. Accessed: 2026-07-14.

Junik Bae, Kwanyoung Park, and Youngwoon Lee. Tldr: Unsupervised goal-conditioned rl via temporal distance-aware representations, 2024. URL https://arxiv.org/abs/2407.08464.

Booster. Booster T1 humanoid, 2025. URL https://www.booster.tech/zh/.

BotRunner64. Teleopit: A lightweight and scalable whole-body teleoperation framework for humanoid robots. https://github.com/BotRunner64/Teleopit, 2025. GitHub repository. Accessed: 2026-07-12.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018. URL https://arxiv.org/abs/1810.12894.

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.

Pranay Dugar, Aayam Shrestha, Fangzhou Yu, Bart van Marum, and Alan Fern. Learning multi-modal whole-body control for real-world humanoid robots, 2025. URL https://arxiv.org/abs/2408.07295.

EGalahad. Mimic-lite, 2025. URL https://github.com/EGalahad/mimic-lite.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function, 2018. URL https://arxiv.org/abs/1802.06070.

Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans, 2024. URL https://arxiv.org/abs/2406.10454.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control, 2016. URL https://arxiv.org/abs/1611.07507.

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

Tairan He, Zhengyi Luo, Wenli Xiao, Chong Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Learning human-to-humanoid real-time whole-body teleoperation. *arXiv preprint arXiv:2403.04436*, 2024a.

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, 2024b.

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.

Jongmin Kim, Youngwoon Lee, Pieter Abbeel, and Guanya Shi. Variational curriculum reinforcement learning for unsupervised discovery of skills, 2023. URL `https://arxiv.org/abs/2303.16342`.

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching, 2020. URL `https://arxiv.org/abs/1906.05274`.

Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, Alessandro Lazaric, Matteo Pirotta, and Guanya Shi. Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning, 2025. URL `https://arxiv.org/abs/2511.04131`.

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`.

Hao Liu and Pieter Abbeel. Cic: Contrastive intrinsic control for unsupervised skill discovery, 2022. URL `https://arxiv.org/abs/2202.00161`.

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.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Sirui Chen, Fernando Castañeda, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Zi Wang, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi "Jim" Fan, and Yuke Zhu. Sonic: Supersizing motion tracking for natural humanoid whole-body control, 2025. URL `https://arxiv.org/abs/2511.07820`.

Ian Mason. 100STYLE: A motion capture dataset of 100 locomotion styles, 2023. URL `https://www.ianxmason.com/100style/`.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction, 2017. URL `https://arxiv.org/abs/1705.05363`.

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.

Roboparty. Roboparty github repository. `https://github.com/Roboparty`, 2025. Accessed: 2026-07-14.

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.

Agon Serifi et al. Vmp: Versatile motion priors for robustly tracking motion on physical characters. *Computer Graphics Forum*, 43(8), 2024. doi: 10.1111/cgf.15175. URL https://doi.org/10.1111/cgf.15175.

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.

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

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards, 2021b. URL https://arxiv.org/abs/2103.07945.

Unitree. Humanoid G1, 2025. URL https://www.unitree.com/cn/g1/.

Unitree Robotics. Unitree h1: Universal humanoid robot. https://www.unitree.com/cn/h1, 2023. Accessed: 2026-07-14.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations, 2021. URL https://arxiv.org/abs/2102.11271.

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, Qiayuan Liao, Brent Yi, Louis Le Lay, Koushil Sreenath, and Pieter Abbeel. mjlab: A lightweight framework for gpu-accelerated robot learning, 2026. URL https://arxiv.org/abs/2601.22074.

Yanjie Ze, João Pedro Araújo, Jiajun Wu, and C. Karen Liu. Gmr: General motion retargeting, 2025. URL https://github.com/YanjieZe/GMR. GitHub repository.

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, 2025a.

Zhikai Zhang, Jun Guo, Chao Chen, Jilong Wang, Chenghuai Lin, Yunrui Lian, Han Xue, Zhenrong Wang, Maoqi Liu, Jiangran Lyu, Huaping Liu, He Wang, and Li Yi. Track any motions under any disturbances, 2025b. URL https://arxiv.org/abs/2509.13833.