P-BFM: PERCEPTIVE BEHAVIORAL FOUNDATION MODELS FOR HUMANOID CONTROL VIA UNSUPERVISED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

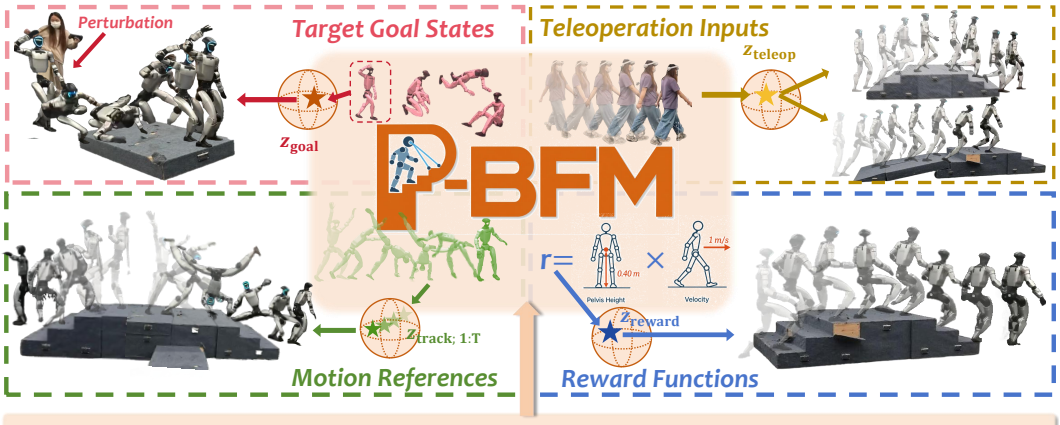

Figure 1: Unified intent representation in **P-BFM**. Motion references, teleoperation inputs, target states, and reward functions are mapped into a shared terrain-agnostic intent space. A single perceptive policy realizes the same intent through distinct behaviors adapted to different terrains. Videos: `https://anonymous.4open.science/w/review-site-61558-CD2E/`

ABSTRACT

Behavioral foundation models (BFMs) provide humanoids with reusable repertoires of whole-body behaviors; in real-world humanoid control, the same behavioral intent may require fundamentally different motions under different terrain constraints. Covering such variation with terrain-matched demonstrations requires enumerating a combinatorial space of behaviors and terrain geometries. We introduce **P-BFM**, a perceptive behavioral foundation model that factorizes humanoid control into terrain-invariant intent and terrain-conditioned realization. A forward–backward representation maps motion references, target states, and reward functions into a shared intent space without terrain information, while a perceptive policy learns how to realize each intent under the surrounding terrain geometry. We further identify a complementary limitation of purely latent control: behavioral latents can preserve local motion patterns while losing precise global progress. We therefore introduce Explicit Root Instructions (ERI), which separates global transport (e.g., heading and planar velocity) from behavioral intent, while leaving local whole-body adaptation to the perceptive policy. This decomposition leads to a characteristic capability absent from conventional BFMs: one fixed intent induces different, terrain-appropriate behaviors, including changes in step height, stride, posture, foothold placement, and contacts, without terrain-specific demonstrations. Across challenging terrains, **P-BFM** improves motion fidelity, root-control precision, and perturbation recovery over strong baselines. A single frozen policy further supports motion tracking, goal reaching, reward inference, teleoperation, and terrain-aware recovery, and transfers zero-shot to a Unitree G1 using onboard depth sensing.

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

1 INTRODUCTION

Humanoid robots are designed to operate in human-centric environments, with versatile embodiments suited to diverse physical tasks. Realizing this potential requires a reusable skill space and a unified controller capable of executing a broad range of skills. Recent advances in Behavior Foundation Models (BFMs) have made substantial progress in this direction, enabling a single controller to execute diverse behaviors through a shared skill space learned from large-scale Mocap datasets.

However, for humanoid robots operating in the real world, generality should also encompass perceiving and adapting to their surroundings. Many tracking-based BFMs rely heavily on demonstrations, and extending them to perceptive control substantially expands the data space to combinations of terrain geometry and motion. Covering this space with paired demonstrations is difficult, limiting generalization to unseen combinations. Prior approaches Wang et al. (2026) address this bottleneck through terrain-aware motion augmentation, but the underlying optimization can yield suboptimal references, particularly for behaviors involving rich contact with the ground.

To address this limitation, we propose a **Perceptive Behavior Foundation Model** (**P-BFM**), which learns through autonomous interaction with complex terrains rather than relying solely on terrain-matched motion imitation. We build on forward–backward (FB) representations, an unsupervised RL (Reinforcement Learning) framework that jointly learns behavioral representations and a latent-conditioned policy, allowing task-level reward functions to be mapped to latent variables for policy execution. These rewards map robot states to scalar values and specify behavioral intents, such as raising an arm or lying down. This exploration-driven framework enables the robot to acquire diverse behaviors without requiring demonstrations for terrain–motion combination.

Within this framework, the reward-specified intent should remain consistent across terrains, while its execution adapts to the terrain geometry. We therefore separate intent encoding from perception-driven execution. The backward representation encodes body-state features without terrain inputs, allowing the same reward specification to yield the same latent intent across terrains. The policy uses depth observations to realize intents specified by flat-ground rewards across diverse terrains. However, weak root-motion supervision in unsupervised learning can lead to heading drift or stepping in place before obstacles. We therefore introduce ERI (Explicit Root Instructions), which adds target heading and base-velocity instructions to improve root tracking while preserving terrain adaptation.

Experiments in simulation and the real world demonstrate that **P-BFM** outperforms all evaluated baselines in precision and robustness across terrains, supporting motion tracking, goal reaching, reward inference, fall recovery, and teleoperation within a shared perceptive policy. By realizing the same intent through terrain-adaptive behaviors without terrain-specific demonstrations, **P-BFM** enables humanoid robots to execute a shared repertoire of skills across different environments.

Our contributions are summarized as follows:

- We propose **P-BFM**, which uses unsupervised RL and FB representations to unify motions, goal states, and rewards in a terrain-agnostic intent space, enabling perceptive control across terrains without terrain-specific demonstrations.
- We introduce ERI, which complements unsupervised behavior learning with explicit heading and base-velocity supervision, enabling precise root control during terrain-adaptive execution.
- We demonstrate superior precision and robustness over existing baselines through extensive experiments and ablations, and validate diverse terrain-adaptive behaviors on a physical humanoid.

2 RELATED WORK

2.1 BEHAVIORAL FOUNDATION MODELS AND HUMANOID WHOLE-BODY CONTROL

Learning from human motion offers a practical foundation for expressive and diverse whole-body humanoid control. BeyondMimic Liao et al. (2025) combines robust motion tracking with guided diffusion to synthesize and compose motion primitives for downstream tasks. Similarly, NVIDIA's SONIC Luo et al. (2026) scales motion tracking with large-scale human motion data to learn a unified whole-body controller for diverse humanoid behaviors. Complementarily, unsupervised RL explores behaviors without tracking specific motion references, aiming to learn reward-inferable

representations through self-exploration. Recent methods based on forward–backward representations Touati & Ollivier (2021), including Meta Motivo Tirinzoni et al. (2025), BFM-Zero Li et al. (2025), and UFO RoboParty Lab Team (2026), organize behaviors in a shared latent space where reward functions specify behavioral latents. They demonstrate diverse whole-body behaviors, perturbation recovery, and flexible task specification, but lack environmental perception for terrain-aware control. Their self-exploration capacity nevertheless offers a promising foundation for terrain adaptation without terrain-specific demonstrations. Building on this, **P-BFM** integrates depth perception with latent-conditioned control and learns through interaction with diverse terrains, allowing the same intent to be realized through terrain-adaptive behaviors.

2.2 Perceptive Locomotion and Terrain Adaptation

Perceptive humanoid control uses environmental observations to anticipate obstacles and adapt movement to terrain geometry Long et al. (2024); He et al. (2025); Zhu et al. (2026); Wu et al. (2026); Zhuang et al. (2026); Wang et al. (2026); Yu et al. (2026); Duan et al. (2024); Song et al. (2025); Sun et al. (2026a); Zhuang et al. (2024); Cui et al. (2026); Sun et al. (2026b); Wang et al. (2025). PIM Long et al. (2024), AME He et al. (2025), Hiking in the Wild Zhu et al. (2026) and SSR Yu et al. (2026) integrate terrain maps or depth observations for robust locomotion. Data-driven approaches, such as Perceptive Humanoid Parkour Wu et al. (2026) and Deep Whole-body Parkour Zhuang et al. (2026), enable dynamic whole-body skills but still depend on sufficient motion coverage and terrain-compatible demonstrations. PMT Wang et al. (2026) reduces this dependence through offline reference optimization and teacher–student training, but separating reference synthesis from policy learning can introduce discrepancies and complicate extension to contact-rich behaviors such as getting up. In contrast, **P-BFM** learns terrain adaptation through unsupervised RL and autonomous interaction, without terrain-specific demonstrations or reference synthesis. A shared policy uses latent intents to specify behavior and depth observations to adapt execution.

3 Preliminary

3.1 Problem Formulation

We formulate perceptive humanoid control as a partially observable Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, P, \Omega, \gamma)$, where $\mathcal{S}$, $\mathcal{O}$, and $\mathcal{A}$ denote the state, observation, and action spaces, P denotes transition dynamics, Ω is the observation model, and $\gamma \in [0, 1)$ is the discount factor. The policy's observations include o_t^p and h_t^v, where o_t^p contains proprioceptive information, previous actions, and their history, and $h_t^v \in \mathbb{R}^{256}$ is the temporally aggregated visual feature from 8 depth frames, each represented by a 256-dimensional vector. The action $a_t \in \mathbb{R}^{29}$ specifies joint targets executed by a low-level PD controller. The policy receives o_t^p, including a four-frame history of proprioception and previous actions, visual features h_t^v, an estimated root velocity $\hat{\mathbf{v}}_t^{\text{yaw}}$, a latent z_t, and global heading and planar-velocity instructions $c_{h,t}$ and $c_{v,t}$, yielding $a_t \sim \pi_\theta(\cdot \mid o_t^p, h_t^v, \hat{\mathbf{v}}_t^{\text{yaw}}, z_t, c_{h,t}, c_{v,t})$. The latent z_t specifies the desired behavior, visual features guide terrain adaptation, and global instructions specify heading and velocity requirements.

Our goal is to learn a shared latent-conditioned policy that executes the behavior specified by z_t across diverse terrains while tracking global instructions, without task-specific policy optimization. Training may use privileged information, including true velocity and terrain information, whereas deployment relies only on available sensor observations, the latent z_t, and global instructions.

3.2 Preliminaries on Forward–Backward Representations

Forward–backward (FB) representations (Touati & Ollivier, 2021) provide a reward-free framework for learning a family of latent-conditioned policies. For clarity, we introduce the standard formulation with fully observed states. Let ρ be a reference state distribution. FB learns a forward representation $F : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}^d$ and a backward representation $B : \mathcal{S} \to \mathbb{R}^d$, where each latent $z \in \mathbb{R}^d$ indexes a policy π_z.

The discounted successor-state measure of π_z is $M^{\pi_z}(ds' \mid s, a) = \sum_{k=0}^{\infty} \gamma^k \Pr(s_{t+k+1} \in ds' \mid s_t = s, a_t = a, \pi_z)$. FB approximates this measure with the low-rank factorization

$$M^{\pi_z}(ds' \mid s, a) \approx F(s, a, z)^\top B(s')\, \rho(ds'). \tag{1}$$

For a downstream state reward $\tilde{r}$, the corresponding latent is $z_{\tilde{r}} = \mathbb{E}_{s\sim\rho}[B(s)\tilde{r}(s)] \approx \frac{1}{N}\sum_{i=1}^{N} B(s_i)\tilde{r}(s_i)$, with $s_i \sim \rho$. This gives $Q_{\tilde{r}}^{\pi_{z_{\tilde{r}}}}(s, a) \approx F(s, a, z_{\tilde{r}})^\top z_{\tilde{r}}$, so the policy can

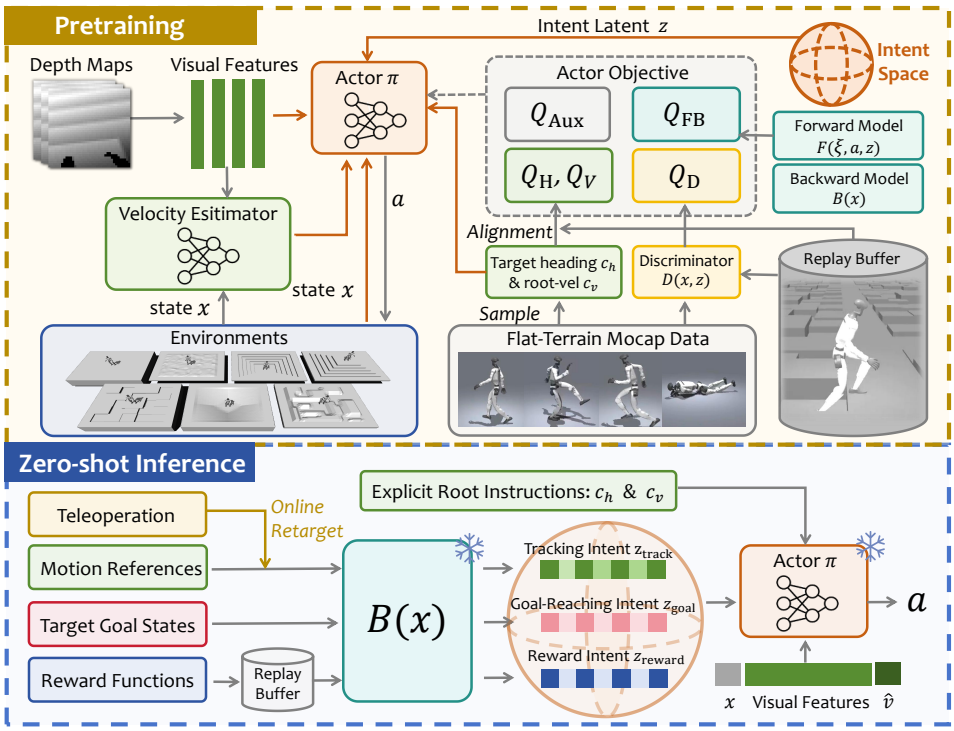

Figure 2: Overview of **P-BFM**. During pretraining, we learn a terrain-agnostic intent space through FB learning and a perceptive actor that interprets the intent latent into terrain-adaptive behavior. At inference, different task specifications are encoded into latents, and the same actor executes them across terrains without task-specific training. Orange arrows denote inputs to the actor.

favor actions with high inner-product values. In **P-BFM**, rewards are defined on body-state features as $\tilde{r}(s) = r(\chi(s))$. This lets the reward-to-latent mapping specify behavioral intent, while the perception-conditioned policy handles terrain-dependent execution.

4 Method

Our goal is to learn a single low-level humanoid controller that can realize diverse intents while adapting to complex terrain geometry. As shown in Fig. 2, **P-BFM** consists of a terrain-agnostic intent space and a terrain-aware perceptive actor: the intent space specifies what behavior to perform, and the actor determines how to execute it with perception.

During pretraining, we encode depth observations into visual features for terrain perception (Sec. 4.1), learn the intent space with FB representation, and train the actor with discriminator and auxiliary critics (Sec. 4.3). The FB objective enables zero-shot inference, the discriminator improves motion quality, and auxiliary critics facilitate sim-to-real transfer. To address the weak global-motion supervision in unsupervised RL, we introduce ERI (Explicit Root Instructions), which combines root heading and base-velocity instructions with intent latents to preserve commanded progress during terrain-adaptive execution (Sec. 4.2). At inference, different task specifications are mapped to corresponding intent latents, including tracking intent, goal-reaching intent, and reward intent (Sec. 4.4). The same perceptive actor then generates terrain-adaptive actions conditioned on the inferred latent, without task-specific policy optimization.

4.1 Perception Integration

To execute the same intent across terrains, the actor requires deployable terrain perception. Following the temporal aggregation and preprocessing design of Hiking in the Wild Zhu et al. (2026), we maintain a 37-frame depth buffer, from which we sample every 5 frames to obtain $K = 8$ observations with ages $(35, 30, 25, 20, 15, 10, 5, 0)$, while applying a per-episode temporal offset of 0–1 frames to model sensing latency. Each 36×64 depth frame is cropped to 18×32, filtered with a 3×3 Gaussian kernel ($\sigma = 1$), clipped to $[0, 2.5]$ meters, normalized. The resulting sequence

is encoded by a shared CNN and aggregated with a GRU into h_t^v, which is used by the policy and velocity estimator to adapt execution to surrounding geometry.

Let o_t^p denote proprioceptive inputs, including joint and inertial measurements, previous actions, and proprioceptive history. The estimator $\hat{\mathbf{v}}_t^{\text{yaw}} = E_\eta(o_t^p, h_t^v)$ predicts the three-dimensional root linear velocity in a yaw-aligned frame. It receives no intent latent, global instructions, or true velocity, making it deployable. The actor is conditioned as

$$a_t \sim \pi_\theta(\cdot \mid o_t^p, h_t^v, \text{sg}(\hat{\mathbf{v}}_t^{\text{yaw}}), z_t, c_{h,t}, c_{v,t}), \tag{2}$$

where sg stops gradients, z_t specifies the intent, and $(c_{h,t}, c_{v,t})$ are the global instructions introduced in Sec. 4.2. Training may use privileged terrain and true velocity for critics or auxiliary losses, while deployment uses only onboard observations.

4.2 ERI: Explicit Root Instructions in Unsupervised RL

Although the intent is inferred from body-state features that include root states, unsupervised RL provides weak supervision for precise root-motion tracking. The policy may therefore preserve local patterns while drifting in heading or losing forward progress on challenging terrains. We address this with ERI (Explicit Root Instructions), which augments the intent-conditioned policy with target heading and base-velocity instructions, providing direct root-motion supervision while preserving perception-driven terrain adaptation.

For tracking intent, z_t encodes the behavior from a short reference window, while ERI provides explicit target root heading and planar velocity from the same reference. Together, they encourage the robot to follow the reference motion instead of drifting or stepping in place before obstacles. Let $\psi_t, \psi_t^\star$ denote current and target headings, and $\mathbf{v}_{t,xy}^\star$ the target world-frame planar velocity. We encode the instructions as

$$\begin{aligned} c_{h,t} &= m_{h,t} \begin{bmatrix} 1 - \cos(\Delta\psi_t) \\ \sin(\Delta\psi_t) \end{bmatrix}, \quad \Delta\psi_t = \psi_t^\star - \psi_t, \\ c_{v,t} &= \begin{bmatrix} m_{v,t} R(-\psi_t) \mathbf{v}_{t,xy}^\star \\ m_{v,t} \end{bmatrix}, \end{aligned} \tag{3}$$

where $R(-\psi_t)$ rotates the velocity instruction into the current heading frame, and $m_{h,t}, m_{v,t} \in \{0, 1\}$ mark valid targets. Zero velocity is treated as a valid stopping instruction; random or goal-conditioned latents do not automatically receive reference-velocity targets.

To train the policy to follow ERI, critics $Q_H(\xi, a, z)$ and $Q_V(\xi, a, z)$ learn from

$$\begin{aligned} r_{H,t} &= -m_{h,t} \left[1 - \cos(\psi_{t+1}^\star - \psi_{t+1})\right], \\ r_{V,t} &= -m_{v,t}\, \ell_{\text{Huber}} \left(\frac{\|\mathbf{v}_{t+1,xy}^{\text{sim}} - \mathbf{v}_{t,xy}^\star\|_2}{\sigma_v} \right). \end{aligned} \tag{4}$$

The heading reward uses the next reference heading, while the velocity reward compares post-action planar velocity with the current instruction. Since Q_V constrains only planar velocity, the policy can still adapt vertical motion to the terrain. Here, $\ell_{\text{Huber}}(u) = u^2/2$ for $0 \le u \le 1$ and $u - 1/2$ otherwise, and σ_v sets the velocity-error scale.

4.3 Pretraining of P-BFM

Pretraining learns two coupled components: a terrain-agnostic intent space and a terrain-aware actor. We implement $B(s) = B_\phi(x)$, where $x = \chi(s)$ contains proprioceptive and privileged body-state information, but no depth, terrain maps, or instructions. In contrast, $F_\psi(\xi, a, z)$ receives history, terrain information, and instructions through ξ. The FB score is $Q_{\text{FB}}(\xi, a, z) = F_\psi(\xi, a, z)^\top z$, thus the same target body state or reward is encoded into the same latent regardless of terrain, while F_ψ and the actor handle terrain-dependent execution.

The policy is trained by online interaction with varied simulated terrains. Forty flat-ground LaFAN sequences provide behavior prompts and positives for the conditional discriminator $D_\omega(x, z)$, without terrain-specific motion synthesis. We train F and B with the FB Bellman objective and orthonormality regularization. The discriminator reward Q_D, physical auxiliary critic Q_A, and ERI critics Q_H, Q_V encourage natural motion, physical regularity, and root tracking.

The actor maximizes the FB value and auxiliary critics:

$$\mathcal{L}_\pi = -\overline{Q}_{\mathrm{FB}} - \alpha \sum_{k\in\{D,A,H,V\}} \lambda_k \overline{Q}_k, \qquad \alpha = \mathrm{sg}(\mathbb{E}[|Q_{\mathrm{FB}}|])\,. \tag{5}$$

Here, $\overline{Q}_k$ averages conservative critic estimates, and H, V are used only for valid ERI targets.

We also train the deployable velocity estimator with simulator velocity:

$$\mathcal{L}_{\mathrm{est}} = \frac{1}{3}\mathbb{E}\left[\|\hat{\mathbf{v}}_t^{\mathrm{yaw}} - \mathbf{v}_{t,\mathrm{sim}}^{\mathrm{yaw}}\|_2^2\right]. \tag{6}$$

We minimize $\mathcal{L}_\pi + \lambda_{\mathrm{est}}\mathcal{L}_{\mathrm{est}}$. The estimation loss updates the estimator and depth encoder, while policy gradients are stopped through the estimated velocity.

4.4 Zero-shot Inference of P-BFM

With all parameters fixed, **P-BFM** supports three intent types: *tracking intent*, *goal-reaching intent*, and *reward intent*, derived from motion references, target goal states, and reward functions, respectively. All intents are encoded through the frozen backward representation:

$$z = \begin{cases} \mathcal{P}\left(\dfrac{1}{L_t}\displaystyle\sum_{k=0}^{L_t-1} B_\phi(x_{t+k}^{\mathrm{ref}})\right), & \text{tracking intent,} \\ \mathcal{P}\big(B_\phi(x^\star)\big), & \text{goal-reaching intent,} \\ \mathcal{P}\left(\dfrac{1}{N}\displaystyle\sum_{i=1}^{N} r(x_i) B_\phi(x_i)\right), & \text{reward intent.} \end{cases} \tag{7}$$

Here, x_{t+k}^{ref} denotes an aligned reference body state, $x^\star$ is a target body state, and $x_i \sim \mu$ are replay body-state features. For a reference sequence of length T, we use $L_t = \min(8, T-t)$. The reward $r(x)$ specifies the desired behavior, and $\mathcal{P}$ normalizes nonzero vectors to norm $\sqrt{d}$, matching the latent scale used in training. Teleoperation uses the same tracking interface with online retargeted body-state targets. At deployment, the policy executes its mean action and uses ERI heading and planar-velocity instructions whenever valid.

5 Experiments

We evaluate whether **P-BFM** can translate different forms of behavioral intent into robust terrain-adaptive control. Following Sec. 4.4, we consider three intent types: *goal-reaching intent*, *tracking intent*, and *reward intent*. These intents are derived from target goal states, motion references, and reward functions, respectively. They correspond to three main evaluation settings: goal reaching, motion tracking, and reward inference. We further include teleoperation to test whether the learned policy can follow online user instructions while adapting to real-world terrain.

Across these settings, our evaluation is organized around four questions:

- Does **P-BFM** outperform existing general-purpose humanoid controllers in motion tracking on challenging terrains?
- Can **P-BFM** generate terrain-adaptive behaviors from the same intent across diverse terrains?
- How do explicit root instructions improve control precision and terrain traversal?
- How does **P-BFM** perform on a real humanoid robot using onboard sensing?

Experimental Setup and Metrics. We train **P-BFM** using Unitree G1 in MJLab Zakka et al. (2026) for 192 million environment steps on NVIDIA H200×8 GPUs in approximately 10 hours and perform zero-shot inference with all model parameters frozen. Optimization settings are provided in Appendix A.5.

We report joint-angle MAE E_{jpos}, root horizontal-velocity error $E_{\mathrm{vel}}^{\mathrm{root}}$, root-heading error $E_{\mathrm{head}}^{\mathrm{root}}$, foot sliding (FS; mean displacement in cm at 50 Hz), and tracking recovery rate (RR). Recovery requires heading error $< 30^\circ$ and horizontal-velocity error $\leq 0.5\,\mathrm{m/s}$ within 5 s after perturbation

Table 1: Comparison with existing perception-based general-purpose humanoid controllers on seen motions and a subset of 994 unseen SOMA motions.

Method	Seen motions					Unseen motions				
	E_{jpos} (rad) ↓	$E_{\text{vel}}^{\text{root}}$ (m/s) ↓	$E_{\text{head}}^{\text{root}}$ (°) ↓	FS (cm) ↓	RR (%) ↑	E_{jpos} (rad) ↓	$E_{\text{vel}}^{\text{root}}$ (m/s) ↓	$E_{\text{head}}^{\text{root}}$ (°) ↓	FS (cm) ↓	RR (%) ↑
UFO	0.15 ± 0.06	0.65 ± 0.46	49.6 ± 29.3	0.092 ± 0.050	3.4	0.16 ± 0.06	0.37 ± 0.27	28.9 ± 23.9	0.066 ± 0.047	16.6
PMT	0.21 ± 0.04	**0.20** ± 0.08	8.8 ± 7.7	0.078 ± 0.029	2.4	0.46 ± 0.09	0.86 ± 0.27	84.4 ± 22.5	0.205 ± 0.103	0.5
P-BFM	**0.11** ± 0.04	0.25 ± 0.12	**6.9** ± 5.4	**0.043** ± 0.026	**41.2**	**0.13** ± 0.06	**0.21** ± 0.12	**6.7** ± 4.7	**0.026** ± 0.016	**61.4**

*For seen-motion evaluation, PMT is evaluated on its only released reference trajectory, whereas UFO and **P-BFM** are evaluated on 40 LaFAN sequences.

removal.Both conditions must hold continuously for at least 1 s and remain satisfied until the 10 s rollout ends. Trials remaining within tolerance also count as successful. We evaluate all methods on 7 terrain types: flat, rough, ascending stairs, descending stairs, platforms, inverted pyramid slopes, and box obstacles. Each type has 10 difficulty levels, with flat terrain unchanged across levels. Appendix A.6 details the terrains; Appendices A.8, A.9, and A.10 define the metrics, recovery protocol, and aggregation.

5.1 Comparison with representative humanoid control baselines.

To answer the first question, we compare **P-BFM** with general-purpose humanoid controllers UFO and PMT. UFO is an unsupervised RL framework based on FB representations, while PMT adapts generic reference motions to terrains through offline optimization and incorporates terrain perception into reference-motion tracking. Under matched robot models and terrain conditions, we evaluate tracking accuracy, motion fidelity, and recovery on challenging terrains.

Since PMT's full training dataset is not publicly available, its motion coverage cannot be verified. We therefore evaluate its released model on the only reference trajectory provided, while evaluating **P-BFM** and UFO on a broader set of 40 LaFAN sequences. Reference-motion coverage and trial counts are detailed in Appendix A.7. As shown in Table 1, despite this broader motion coverage, **P-BFM** achieves lower joint-position and heading errors, less foot sliding, and substantially higher recovery rates than the baselines. Although the differing evaluation sets preclude a strictly matched comparison, these results suggest higher motion style reproduction accuracy and robustness for **P-BFM**. We attribute these gains to end-to-end learning through autonomous exploration, which enables motion adaptation through direct interaction with diverse terrains rather than offline reference-motion optimization.

Under matched evaluation conditions on the unseen motion dataset, **P-BFM** consistently outperforms the baselines in motion fidelity, root-instruction tracking, and recovery, demonstrating stronger generalization beyond its training motions. The lower heading and velocity errors highlight the effectiveness of explicit root instructions, while the higher recovery rate suggests that unsupervised exploration helps the policy learn to recover from diverse off-nominal states.

5.2 Terrain-Adaptive Behavior from a Single Intent

We examine whether **P-BFM** can realize the same reward-inferred intent through terrain-dependent motion patterns. Using forward walking as an example, we define a reward function for flat-ground walking and infer the corresponding latent intent through reward inference. This single latent is then evaluated across multiple terrains. As shown in Fig. 3, the policy does not simply reproduce a fixed walking pattern. Instead, it autonomously changes step height, stride length, body posture, and foothold selection according to the perceived terrain geometry. It is worth noting that no terrain-specific motion demonstrations are provided during training. The adaptive behaviors emerge from autonomous exploration in complex environments rather than from terrain-matched reference motions, demonstrating that **P-BFM** can discover distinct ways to realize the same reward-inferred intent under different terrain constraints.

5.3 Ablation Study

We evaluate ERI by separately removing heading and velocity instructions from **P-BFM**. Table 2 shows that both instructions substantially improve their respective tracking metrics on seen and unseen motions, with relatively modest differences in the remaining tracking and foot-skating metrics.



Figure 3: Terrain-adaptive behavior from a single forward-walking intent. Given the same reference-derived intent, **P-BFM** produces distinct locomotion patterns across terrains, including stair ascent/descent, slope traversal, discrete foothold selection, and rough-terrain stepping.

Table 2: Ablation study of global instructions on seen and unseen motions. "w/o head" removes the global heading instruction, while "w/o vel" removes the global base-velocity instruction.

Method	Seen motions					Unseen motions				
	E_{jpos} (rad) ↓	$E_{\text{vel}}^{\text{root}}$ (m/s) ↓	$E_{\text{head}}^{\text{root}}$ (°) ↓	FS (cm) ↓	RR (%) ↑	E_{jpos} (rad) ↓	$E_{\text{vel}}^{\text{root}}$ (m/s) ↓	$E_{\text{head}}^{\text{root}}$ (°) ↓	FS (cm) ↓	RR (%) ↑
w/o head	0.12 ± 0.04	0.28 ± 0.14	19.3 ± 14.2	0.045 ± 0.026	20.7	**0.13** ± 0.06	0.22 ± 0.12	14.6 ± 14.3	0.026 ± 0.018	39.5
w/o vel	0.12 ± 0.04	0.41 ± 0.26	7.7 ± 6.9	**0.032** ± 0.018	22.4	**0.13** ± 0.06	0.32 ± 0.19	**6.3** ± 4.9	**0.020** ± 0.012	42.9
P-BFM	**0.11** ± 0.04	**0.25** ± 0.12	**6.9** ± 5.4	0.043 ± 0.026	**41.2**	**0.13** ± 0.06	**0.21** ± 0.12	6.7 ± 4.7	0.026 ± 0.016	**61.4**

Fig. 4 illustrates the role of each instruction: without heading instructions, the robot gradually drifts from the intended direction; without velocity instructions, it reproduces local stepping motions but makes little forward progress when facing challenging terrain. In contrast, **P-BFM** preserves the reference behavior while maintaining the intended heading and forward progress, demonstrating the importance of both instructions for effective terrain traversal. ERI thus combines unsupervised behavioral exploration with supervised instruction tracking, preserving behavioral diversity while enforcing explicit root-motion requirements.

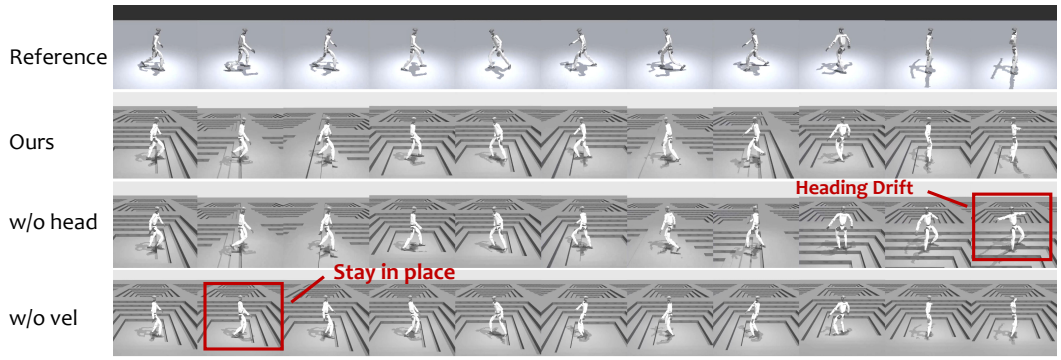

Figure 4: Qualitative ablation. Without heading instructions, the robot exhibits heading drift; without velocity instructions, it steps in place. **P-BFM** maintains the intended heading and progress.

5.4 Real-world Validation

We deploy **P-BFM** on a Unitree G1 humanoid equipped with a RealSense D435i depth camera to examine its transfer from simulation to the real world. The experiments cover teleoperation, motion tracking, goal reaching, reward inference, and terrain-adaptive recovery from external perturbations. Camera settings and depth preprocessing are reported in Appendix A.12.

Teleoperation. Fig. 5 shows real-world teleoperation with perception-driven terrain adaptation. In Fig. 5(a), the robot follows teleoperation-derived walking intent and adapts it to stair traversal. Fig. 5(b) shows contact-rich interaction with stairs, where the robot preserves the intended whole-body behavior despite frequent body-terrain contacts. Fig. 5(c) further shows recovery from an unexpected fall, demonstrating robustness beyond nominal execution.

Also. as illustrated in the upper-left panel of Fig. 1, the same teleoperation input leads to different executions on stairs and slopes: higher foot lifting and discrete foothold placement on stairs, and smoother continuous contact adjustment on slopes. These terrain-dependent gait patterns preserve the intended behavior rather than literally reproducing the human motion.



Figure 5: Real-world teleoperation with perception-driven terrain adaptation.

Motion Tracking. As shown in Fig. 1(c), **P-BFM** tracks expressive whole-body motion references, including dynamic cartwheels and dances, on and around an elevated platform. Despite terrain constraints, the robot preserves the reference motion style while adapting foot placement, body posture, and contacts through perception. This demonstrates that tracking intent can be realized in terrain-aware ways within the same policy.

Goal Reaching. As shown in the upper-left part of Fig. 1, **P-BFM** supports any-to-any goal reaching and adaptive behavior across terrain variations. Fig. 6(a) further presents diverse goal-reaching results on different terrains. Starting from different initial configurations, the robot reaches target poses such as sitting, lying down, standing, and arm-raising on stairs, slopes, gaps, and uneven platforms. These results show that goal-reaching intent is not tied to a specific terrain or initial state. During the transition, the robot adapts its contacts and body posture to the surrounding geometry, demonstrating terrain-aware goal reaching.



(a) Various goal reaching on various terrains (b) Autonomously step over an obstacle (c) Autonomously step on stairs

Figure 6: Real-world demonstrations of goal reaching and terrain-aware recovery.

Reward Inference. In lower-right part of Fig. 1, the reward specifies forward velocity and pelvis-height preferences. **P-BFM** infers the corresponding latent intent and executes it on stair-like terrain, maintaining forward progress while adapting its steps to height changes. This shows that reward-derived intents can be grounded in perceptive terrain-adaptive control without additional planning, post-training or motion optimization.

Adaptive Behavior to Perturbation. Fig. 6(b) and Fig. 6(c) show terrain-aware recovery under external perturbations. When kicked near an obstacle, the robot lifts its swing leg and steps over the block to recover balance. When pushed near stairs, it autonomously places its feet on the stair surfaces and steps upward instead of colliding with the step edges. These behaviors show that perception helps convert disturbances into feasible terrain-adaptive recovery motions.

6 Conclusion

We introduced **P-BFM**, a Perceptive Behavior Foundation Model that unifies motion references, target states, and rewards in a terrain-agnostic intent space. A shared perceptive policy realizes these intents across terrains, while ERI complements unsupervised behavior learning with explicit root-motion supervision. Simulation and real-world experiments demonstrate motion tracking, goal reaching, reward inference, teleoperation, and terrain-adaptive recovery without terrain-specific demonstrations. However, behavioral generalization remains limited by the diversity of motion priors and terrains explored during training. Future work will investigate broader motion datasets and more challenging terrains, particularly those requiring complex whole-body contacts.

AI USE STATEMENT

We used generative AI tools for language editing and translation, literature retrieval and reference organization, and feedback on our methodology, experimental protocols, and interpretation of results. AI tools also assisted in drafting and revising selected passages and preparing draft schematic figures. The authors take full responsibility for the final methodology, experiments, results, claims, text, and figures presented in this work.

REFERENCES

Xinru Cui, Linxi Feng, Yixuan Zhou, Haoqi Han, Zhe Liu, and Hesheng Wang. Pilot: A perceptive integrated low-level controller for loco-manipulation over unstructured scenes, 2026. URL https://arxiv.org/abs/2601.17440.

Helei Duan, Bikram Pandit, Mohitvishnu S. Gadde, Bart van Marum, Jeremy Dao, Chanho Kim, and Alan Fern. Learning vision-based bipedal locomotion for challenging terrain, 2024. URL https://arxiv.org/abs/2309.14594.

Junzhe He, Chong Zhang, Fabian Jenelten, Ruben Grandia, Moritz BÄcher, and Marco Hutter. Attention-based map encoding for learning generalized legged locomotion, 2025. URL https://arxiv.org/abs/2506.09588.

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. *arXiv preprint arXiv:2511.04131*, 2025. URL https://arxiv.org/abs/2511.04131.

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

Junfeng Long, Junli Ren, Moji Shi, Zirui Wang, Tao Huang, Ping Luo, and Jiangmiao Pang. Learning humanoid locomotion with perceptive internal model. *arXiv preprint arXiv:2411.14386*, 2024. URL https://arxiv.org/abs/2411.14386.

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 Jeon, Tairan He, Haoru Xue, Wenli Xiao, Zi Wang, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi Fan, and Yuke Zhu. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, July 2026.

RoboParty Lab Team. UFO: An unsupervised reinforcement learning framework for humanoid control. Software and project release, 2026. URL https://github.com/Roboparty/UFO. Project page: https://roboparty.github.io/UFO/.

Haolin Song, Hongbo Zhu, Tao Yu, Yan Liu, Mingqi Yuan, Wengang Zhou, Hua Chen, and Houqiang Li. Gait-adaptive perceptive humanoid locomotion with real-time under-base terrain reconstruction, 2025. URL https://arxiv.org/abs/2512.07464.

Jingkai Sun, Gang Han, Pihai Sun, Wen Zhao, Jiahang Cao, Jiaxu Wang, Qiang Zhang, and Yijie Guo. Dpl: Depth-only perceptive humanoid locomotion via realistic depth synthesis and cross-attention terrain reconstruction. *IEEE Robotics and Automation Letters*, 11(8):9407–9414, August 2026a. ISSN 2377-3774. doi: 10.1109/lra.2026.3703584. URL http://dx.doi.org/10.1109/LRA.2026.3703584.

Wandong Sun, Yongbo Su, Leoric Huang, Alex Zhang, Dwyane Wei, Mu San, Daniel Tian, Ellie Cao, Baoshi Cao, Yang Liu, Finn Yan, Ethan Xie, and Zongwu Xie. Now you see that: Learning end-to-end humanoid locomotion from raw pixels, 2026b. URL https://arxiv.org/abs/2602.06382.

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 *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2504.11054.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/003dd617c12d444ff9c80f717c3fa982-Abstract.html.

Dewei Wang, Xinmiao Wang, Xinzhe Liu, Jiyuan Shi, Yingnan Zhao, Chenjia Bai, and Xuelong Li. More: Mixture of residual experts for humanoid lifelike gaits learning on complex terrains, 2025. URL https://arxiv.org/abs/2506.08840.

Zifan Wang, Yizhao Li, Teli Ma, Qiang Zhang, Yudong Fan, Hao Xu, Shuo Yang, and Junwei Liang. Perceptive behavior foundation model: Adapting human motion priors to robot-centric terrain. *arXiv preprint arXiv:2606.08059*, 2026. URL https://arxiv.org/abs/2606.08059.

Zhen Wu, Xiaoyu Huang, Lujie Yang, Yuanhang Zhang, Xi Chen, Pieter Abbeel, Rocky Duan, Angjoo Kanazawa, Carmelo Sferrazza, Guanya Shi, and C. Karen Liu. Perceptive humanoid parkour: Chaining dynamic human skills via motion matching. *arXiv preprint arXiv:2602.15827*, 2026. URL https://arxiv.org/abs/2602.15827.

Ruiqi Yu, Yiwen Wang, Yuan Hao, Jun WU, and Qiuguo Zhu. Ssr: Scaling surefooted and symmetric humanoid traversal to the open world, 2026. URL https://arxiv.org/abs/2605.30770.

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.

Shaoting Zhu, Ziwen Zhuang, Mengjie Zhao, Kun-Ying Lee, and Hang Zhao. Hiking in the wild: A scalable perceptive parkour framework for humanoids, 2026. URL https://arxiv.org/abs/2601.07718.

Ziwen Zhuang, Shenzhe Yao, and Hang Zhao. Humanoid parkour learning, 2024. URL https://arxiv.org/abs/2406.10759.

Ziwen Zhuang, Shaoting Zhu, Mengjie Zhao, and Hang Zhao. Deep whole-body parkour, 2026. URL https://arxiv.org/abs/2601.07701.

A Additional Experimental Details

A.1 Network Architectures

The behavioral latent has dimension $d = 256$, and the actor outputs 29 joint-action components. The backward representation and behavioral latents are normalized to Euclidean norm $\sqrt{d} = 16$. Table 3 summarizes the observation groups.

Table 3: Observation groups used by the full model. Privileged inputs are used during training and are excluded from the deployed actor.

Observation group	Dimension
Proprioceptive state	64
Previous action	29
Four-step proprioception–action history	372
Heading instruction	2
Horizontal-velocity instruction and validity flag	3
Estimated current root velocity	3
Privileged body state	463
Privileged terrain features	273
Ground-truth velocity supervision	3
Aggregated visual feature	256

The 64-dimensional proprioceptive state comprises 29 relative joint positions, 29 joint velocities, projected gravity, and root angular velocity. Each historical observation contains these 64 features and 29 action components, giving $4 \times 93 = 372$ history features.

Residual blocks. Let $L_{p,q}$ denote an affine map from $\mathbb{R}^p$ to $\mathbb{R}^q$. We use pre-normalized projection and residual blocks:

$$P_{p,q}(x) = \mathrm{Mish}\left(L_{p,q}(\mathrm{LN}_p(x))\right), \tag{8}$$

$$H_{p,q}(x) = L_{p,q}(\mathrm{LN}_p(x)), \tag{9}$$

$$R_w(x) = x + P_{w,w}(x). \tag{10}$$

Each residual block contains one affine layer. The output projection H has no activation.

Forward representation and scalar critics. The forward representation F and scalar critics Q_D, Q_A, Q_H, Q_V share the same input organization, but have separate parameters. Here, Q_A denotes the physical auxiliary critic, also written as Q_{aux}.

Their training observation vector $o_c \in \mathbb{R}^{1209}$ concatenates proprioception, privileged body state, previous action, history, privileged terrain features, heading and velocity instructions, and ground-truth velocity features. Two branches encode $[o_c, z] \in \mathbb{R}^{1465}$ and $[o_c, a] \in \mathbb{R}^{1238}$, respectively. Each branch has the form

$$E_{p,w,k} = P_{w,w/2} \circ R_w^k \circ P_{p,w}, \tag{11}$$

where R_w^k denotes k separately parameterized residual blocks. The two branch outputs are concatenated and processed by r additional residual blocks and an output projection $H_{w,m}$.

Table 4: Forward and critic architectures. Each branch includes two affine projections in addition to its listed residual blocks. Each trunk includes an additional output projection. Each ensemble member is a complete, independently parameterized network.

Network	Width w	Branch blocks k	Trunk blocks r	Output m	Members
F	2048	4	6	256	2
Q_D	1024	2	4	1	2
Q_A	2048	2	4	1	2
Q_H	1024	1	3	1	2
Q_V	1024	1	3	1	2

Backward representation. The backward network receives the 527-dimensional concatenation of proprioceptive and privileged body-state features. Its architecture is

$$527 \xrightarrow{\text{Linear}} 256 \xrightarrow{\text{LN, tanh}} 256 \xrightarrow{\text{Linear}} 256 \xrightarrow{\ell_2 \text{ normalization}} B(x), \tag{12}$$

where the final normalization scales the output norm to 16. Thus, B has one hidden layer and two affine layers. It receives neither terrain observations nor global instructions.

Actor. The actor's low-dimensional observation contains 470 features. Appending the detached three-dimensional velocity estimate produces $u_t \in \mathbb{R}^{473}$. Two branches encode u_t and $[u_t, z_t]$ using

$$473 \to 2048 \to 1024, \qquad 729 \to 2048 \to 1024,$$

with a projection block P at each arrow. Their outputs are concatenated with the 256-dimensional visual feature and fused by $P_{2304,2048}$. The policy trunk contains six R_{2048} blocks, followed by $H_{2048,29}$ and $\tanh$. Training uses action sampling with standard deviation 0.05; evaluation uses the mean action.

Discriminator. The conditional discriminator receives $[x, z] \in \mathbb{R}^{783}$ and has layer widths

$$783 \to 1024 \to 1024 \to 1024 \to 1.$$

The first hidden layer uses layer normalization and $\tanh$; the next two hidden layers use ReLU. The output is a scalar logit, with a sigmoid applied when a probability is required.

Observation normalization. Proprioception, privileged body state, previous action, history, and privileged terrain features are standardized using running statistics with momentum 0.01. This normalization has no learned affine parameters. Heading and velocity instructions bypass this standardization. Depth normalization is described below.

A.2 Depth Processing and Visual Encoding

Simulated depth observations. The simulated sensor produces 36×64 images (height $\times$ width), with horizontal and vertical fields of view of 89.51° and 58.29°. Ray distances are converted to optical-axis depth using the corresponding unit-ray direction. The depth-processing profile maps invalid depth observations to the maximum distance and clips depths to $[0, 2.5]\,\mathrm{m}$.

We retain rows $18{:}36$ and columns $16{:}48$, using zero-based, end-exclusive indexing. This produces an 18×32 image without resizing. The crop is smoothed using a 3×3 Gaussian kernel with standard deviation 1 pixel and reflection padding. The filtered depth $\bar{d}_t$ is clipped and quantized as

$$u_t = \mathrm{uint8}\left[\mathrm{round}\left(255 \frac{\bar{d}_t}{2.5\,\mathrm{m}}\right)\right]. \tag{13}$$

The encoder converts the stored bytes to floating-point values by dividing by 255.

Temporal sampling and reset. At the 50 Hz control rate, the visual input contains eight frames sampled from a 37-frame buffer:

$$\mathcal{D}_t = [u_{t-\delta-35}, u_{t-\delta-30}, \ldots, u_{t-\delta-5}, u_{t-\delta}], \qquad \delta \sim \mathrm{Uniform}\{0, 1\}. \tag{14}$$

Frames are ordered from oldest to newest. Adjacent selected frames are separated by $0.1\,\mathrm{s}$, and the selected window spans $0.7\,\mathrm{s}$. The delay δ is sampled at reset and remains fixed within the episode. On reset, the image buffer is filled with the first image of the new episode, preventing visual history from carrying over from the preceding episode.

Camera randomization. At each reset, camera translation is perturbed independently within $\pm 0.01\,\mathrm{m}$ along each camera-link axis. Roll, pitch, and yaw perturbations are sampled within $\pm 2^\circ$, $\pm 10^\circ$, and $\pm 2^\circ$, respectively. These extrinsic perturbations remain fixed within an episode. The selected processing profile uses deterministic Gaussian smoothing without additional stochastic image-intensity augmentation.

Table 5: Shared single-frame depth encoder. Shapes are channels × height × width before flattening.

Layer	Parameters	Output shape
Input	Normalized depth	$1 \times 18 \times 32$
Conv + ReLU	$1 \to 16,\ k=3,\ s=1,\ p=1$	$16 \times 18 \times 32$
Conv + ReLU	$16 \to 32,\ k=3,\ s=2,\ p=1$	$32 \times 9 \times 16$
Conv + ReLU	$32 \to 64,\ k=3,\ s=1,\ p=1$	$64 \times 9 \times 16$
Adaptive average pooling	Output size 2×4	$64 \times 2 \times 4$
Flatten	—	512
Linear + ReLU	$512 \to 256$	256

CNN and temporal aggregation. A shared CNN processes each depth frame independently. Its architecture is given in Table 5. The eight resulting 256-dimensional frame features are passed to a single-layer, unidirectional GRU with hidden dimension 256. The final hidden state is the visual feature h_t^v. Each eight-frame window is encoded from a zero GRU initial state; hidden states are not carried across policy evaluations.

Velocity estimator. The estimator concatenates h_t^v with the current proprioceptive state, previous action, and proprioceptive history. Its input dimension is $256+64+29+372 = 721$. An MLP with widths

$$721 \to 512 \to 256 \to 3$$

uses ELU after its first two affine layers and a linear output. It predicts current root velocity in a yaw-aligned frame, whose vertical axis remains world-up. Behavioral latents, heading instructions, desired velocities, and privileged simulator features are excluded from its input.

The estimator is supervised by same-timestep simulator velocity:

$$\mathcal{L}_{\text{est}} = \frac{1}{3N} \sum_{i=1}^{N} \left\| \widehat{\mathbf{v}}_i^{\text{yaw}} - \mathbf{v}_i^{\text{yaw,sim}} \right\|_2^2 . \tag{15}$$

This loss has coefficient 1 and updates both the estimator and the shared visual encoder. The policy receives $\text{sg}(\widehat{\mathbf{v}}^{\text{yaw}})$, where sg denotes stop-gradient. Policy gradients still update the visual encoder through the direct visual-feature input.

A.3 Actor Objective and Critic Coefficients

For each ensemble, define the conservative estimate

$$Q_X^- = \min(Q_{X,1}, Q_{X,2}), \qquad Q_{\text{FB},k} = F_k(o, z, a)^\top z. \tag{16}$$

Let $\langle \cdot \rangle$ denote a minibatch average and $\langle f \rangle_m$ denote an average over samples with validity mask m. We use the detached FB scale

$$\alpha = \text{sg}\left(\left\langle |Q_{\text{FB}}^-| \right\rangle \right). \tag{17}$$

The actor and estimator are optimized with

$$\begin{aligned} \mathcal{L}_{\text{actor}} = & -\left\langle Q_{\text{FB}}^- \right\rangle \\ & - \alpha \Big[0.01 \left\langle Q_D^- \right\rangle + 0.02 \left\langle Q_A^- \right\rangle \\ & \qquad + 0.002 \left\langle Q_H^- \right\rangle_{m_H} + 0.002 \left\langle Q_V^- \right\rangle_{m_V} \Big] + \mathcal{L}_{\text{est}}. \end{aligned} \tag{18}$$

The coefficients in Table 6 weight critic values in the actor objective. They are distinct from the instantaneous reward coefficients used to train each critic. The heading and velocity terms use their collected instruction contexts and valid samples rather than relabeled instruction targets.

Conditional motion-prior reward. The discriminator is trained with binary cross-entropy, using expert samples as positives and policy samples as negatives. A gradient penalty with coefficient 10 is applied to the discriminator logit on interpolated expert–policy inputs. The reward for Q_D is

$$r_D(x, z) = \log p(x, z) - \log(1 - p(x, z)), \tag{19}$$

where p is the discriminator probability clipped to $[10^{-6}, 1 - 10^{-6}]$.

Table 6: Coefficients in the actor objective. The four auxiliary value terms are additionally multiplied by the detached scale α.

Value term	Role	Base coefficient
Q_{FB}	Behavioral intent	1
Q_D	Conditional motion prior	0.01
Q_A	Physical auxiliary objectives	0.02
Q_H	Heading tracking	0.002
Q_V	Horizontal-velocity tracking	0.002

Table 7: Physical costs and their coefficients inside r_A^{raw}. These coefficients are separate from the actor coefficient 0.02 for Q_A.

Cost	Coefficient
Action change	-0.1
Joint-position limit violation	-10
Body-impact severity	-1
Loaded-foot tangential speed	-4
Ankle-roll magnitude	-1
Foot stumbling	-2

Heading reward. On valid heading transitions, the reward is

$$r_{H,t} = -\left[1 - \cos\left(\psi_{t+1} - \psi^{\star}_{t+1}\right)\right]. \tag{20}$$

The executed and target headings are both evaluated at the next state. Invalid instruction transitions are masked from the corresponding critic and actor objectives.

Horizontal-velocity reward. On valid velocity transitions, the reward is

$$r_{V,t} = -\mathcal{H}_1\left(\frac{\left\|\mathbf{v}^{\mathrm{sim}}_{t+1,xy} - \mathbf{v}^{\star}_{t,xy}\right\|_2}{0.5\,\mathrm{m/s}}\right), \tag{21}$$

where

$$\mathcal{H}_1(u) = \begin{cases} \frac{1}{2}u^2, & 0 \le u \le 1, \\ u - \frac{1}{2}, & u > 1. \end{cases} \tag{22}$$

The reward compares post-action velocity with the current velocity instruction that generated the action. It applies Huber loss to the horizontal error-vector norm, rather than independently to its components. Valid zero-velocity commands remain supervised stopping targets. Heading and velocity bootstrapping is interrupted at instruction-context boundaries.

A.4 Physical Auxiliary Rewards

The auxiliary critic Q_A learns from six physical costs. Their raw weighted sum is

$$\begin{aligned} r_A^{\mathrm{raw}} = & -0.1c_{\mathrm{rate}} - 10c_{\mathrm{limit}} - c_{\mathrm{impact}} \\ & - 4c_{\mathrm{slip}} - c_{\mathrm{ankle}} - 2c_{\mathrm{stumble}}. \end{aligned} \tag{23}$$

Action and joint costs. Let $[x]_+ = \max(x, 0)$. The action-change and joint-limit costs are

$$c_{\mathrm{rate}} = \sum_{j=1}^{29}\left(a^{\mathrm{env}}_{t,j} - a^{\mathrm{env}}_{t-1,j}\right)^2, \tag{24}$$

$$c_{\mathrm{limit}} = \sum_{j=1}^{29}\left([q_j^- - q_j]_+ + [q_j - q_j^+]_+\right). \tag{25}$$

Here, a^{env} denotes the rescaled environment command before action delay. The normalized policy action is scaled by 5 and clipped to $[-5, 5]$ for this command. The soft joint limits (q_j^-, q_j^+) retain 95% of the hard joint range around its midpoint. The ankle cost is

$$c_{\mathrm{ankle}} = q^2_{\text{left ankle roll}} + q^2_{\text{right ankle roll}}. \tag{26}$$

Contact-dependent costs. Let W be robot weight and $\widehat{\mathbf{f}} = \mathbf{f}/(\|\mathbf{f}\|_2 + 10^{-6})$ the regularized contact-force direction. The training slippage cost is

$$c_{\text{slip}} = \sum_{f \in \{L,R\}} \mathbf{1}[\|\mathbf{f}_f\|_2/W > 0.05] \left\| \mathbf{v}_f - (\mathbf{v}_f^\top \widehat{\mathbf{f}}_f)\widehat{\mathbf{f}}_f \right\|_2. \tag{27}$$

It penalizes foot-link velocity tangent to the contact-force direction under load. This training cost differs from the displacement-based FS evaluation metric.

For body impact, let $\mathbf{f}_b^*$ be the contact-force vector with maximum magnitude during the control interval, and $\mathbf{v}_b^{\text{pre}}$ the pre-step body velocity. Define

$$S_\beta(x) = \begin{cases} x^2/(2\beta), & |x| < \beta, \\ |x| - \beta/2, & |x| \geq \beta. \end{cases} \tag{28}$$

The impact cost is

$$c_{\text{impact}} = \min \Bigg\{ 4,\ \sum_{b \in \mathcal{B}} S_{0.25}\left([\|\mathbf{f}_b^*\|_2/W - 0.05]_+\right) \times \left[1 + \min\left(\frac{[\mathbf{v}_b^{\text{pre}\top} \widehat{\mathbf{f}}_b^*]_+}{1\,\text{m/s}}, 2 \right)^2 \right] \Bigg\}. \tag{29}$$

The monitored set $\mathcal{B}$ contains the configured pelvis, shoulder, and hip links. The force direction follows the simulator's contact convention, so positive projected pre-step velocity represents incoming motion.

The stumbling cost uses foot-collision force histories:

$$c_{\text{stumble}} = \max_{g,h} \min \left\{ 1,\ \frac{[\|\mathbf{f}_{g,h}^{xy}\|_2 - |f_{g,h}^z| - 20\,\text{N}]_+}{100\,\text{N}} \right\}, \tag{30}$$

where g indexes foot-collision contact slots and h indexes physics substeps within the control interval.

Dynamic reward normalization. The combined auxiliary reward is divided by a running standard deviation:

$$\widetilde{r}_A = \frac{r_A^{\text{raw}}}{\sigma_A}. \tag{31}$$

The normalizer uses bias-corrected exponential moving averages of the first and second moments, with decay 0.99 and a variance floor of 10^{-8}. The statistics are updated during training. The output is scaled without subtracting the running mean, and normalization is applied to the combined reward rather than separately to each cost. The auxiliary reward supplied to Q_A has no additional control-timestep multiplier.

A.5 Optimization Hyperparameters

We use Adam optimization with the hyperparameters in Table 8. The forward and backward representations are trained with the FB successor-measure objective and backward-embedding orthonormality regularization. No additional implicit-reward scalar Q-regression loss is enabled.

Target parameters are updated as

$$\bar{\theta} \leftarrow (1 - \tau)\bar{\theta} + \tau\theta. \tag{32}$$

The scalar-critic bootstrap targets use the minimum of their two ensemble predictions. The FB successor-measure bootstrap target instead uses the mean of the two forward-network predictions. For actor optimization, both FB values and scalar-critic values use conservative minimum estimates.

A.6 Terrain Configuration

Training and evaluation use the 7 terrain families listed in Table 9. The terrain map contains 10 rows and 7 columns, with 1 column per terrain family. Each tile measures $5 \times 5\,\text{m}$, giving a $50 \times 35\,\text{m}$ main terrain grid. Each non-flat tile has a $0.5\,\text{m}$-wide flat border on all four sides, leaving a $4 \times 4\,\text{m}$ interior.

Table 8: Optimization settings of the evaluated full model.

Parameter	Value
Latent dimension	256
Discount factor	0.98
Minibatch size per GPU	1,024
Effective minibatch size across eight GPUs	8,192
Forward-network learning rate	3×10^{-4}
Backward-network learning rate	1×10^{-5}
Actor learning rate	3×10^{-4}
Scalar-critic learning rates	3×10^{-4}
Discriminator learning rate	5×10^{-6}
Backward orthonormality coefficient	100
Discriminator gradient-penalty coefficient	10
Velocity-estimation loss coefficient	1
Action sampling standard deviation	0.05
Action sampling noise clip	0.3
Forward/backward target-update rate	0.01
Scalar-critic target-update rate	0.005
Physics frequency	200 Hz
Policy frequency	50 Hz

Difficulty configuration. Rows are indexed by $r \in \{0, \ldots, 9\}$. The normalized difficulty of the tile in row r and column c is sampled as

$$d_{r,c} = \frac{r + u_{r,c}}{10}, \qquad u_{r,c} \sim \mathcal{U}(0, 1). \tag{33}$$

Thus, each row specifies a difficulty interval rather than one identical difficulty value for every terrain family. Terrain generation uses a fixed seed of 0, and all evaluated methods share the same generated evaluation map. The parameters below describe the configured generation ranges; the finite set of generated tiles need not attain their endpoints. For the flat terrain family, all ten rows remain physically flat.

Table 9: Terrain families and difficulty-dependent geometry. IDs follow the terrain generator's column order. Here d denotes normalized tile difficulty. Stair traversal directions are defined for motion from the central platform toward the tile boundary.

ID	**Terrain family**	**Geometry parameters**
0	Flat terrain	Planar surface; geometry is independent of difficulty.
1	Perlin-noise rough terrain	Target height range 0 to $H_{\mathrm{rough}}(d) = 0.02 + 0.04d\,\mathrm{m}$ before border resampling.
2	Pyramid stairs (outward descent)	Riser height $h(d) = 0.10 + 0.10d\,\mathrm{m}$; tread depth $0.30\,\mathrm{m}$; central platform $1.6 \times 1.6\,\mathrm{m}$.
3	Inverted pyramid stairs (outward ascent)	Same riser heights, tread depth, and central-platform dimensions as the pyramid stairs, with inverted elevation.
4	Low platforms	Grid-cell width $0.80\,\mathrm{m}$; cell heights sampled from $[-H_{\mathrm{plat}}(d), H_{\mathrm{plat}}(d)]$, where $H_{\mathrm{plat}}(d) = 0.03 + 0.07d\,\mathrm{m}$, before height quantization.
5	Inverted pyramid slope	Configured slope parameter $s(d) = 0.20d$; nominal angle envelope 0 to $\arctan(0.20) \approx 11.31^\circ$.
6	Box-obstacle terrain	Height bound $H_{\mathrm{box}}(d) = 0.05 + 0.10d\,\mathrm{m}$; sampled rectangle side lengths $\{0.4, 0.6, 0.8, 1.0\}\,\mathrm{m}$; 50 placement attempts per tile.

Terrain generation details. Rough terrain uses two Perlin-noise octaves, persistence 0.25, lacunarity 2, and a noise-scale parameter of 20. The sampled noise is normalized to $[0, 1]$ before multiplication by $H_{\mathrm{rough}}(d)$. Subsequent interpolation can reduce the realized height range.

Both stair families contain four concentric tread bands and a central platform, producing five vertical transitions between the tile border and the center. Their central elevations are $+5h(d)$ and $-5h(d)$,

respectively. The configured minimum central-platform width is $1.2\,\mathrm{m}$; fitting an integer number of treads into the tile yields an actual width of $1.6\,\mathrm{m}$.

Low-platform terrain uses a 5×5 grid within the tile interior. Sampled cell heights are rounded to increments of $0.025\,\mathrm{m}$. Adjacent cells with equal quantized heights may be merged, with each merged rectangle spanning at most three cells along either axis. A separate $1.2 \times 1.2\,\mathrm{m}$ central platform has surface height $H_{\mathrm{plat}}(d)$.

Box-obstacle terrain includes both positive and negative height offsets, using full- and half-height levels derived from $H_{\mathrm{box}}(d)$ with $0.005\,\mathrm{m}$ vertical discretization. Rectangle dimensions are sampled independently along the two horizontal axes. Placements may overlap and overwrite earlier placements; therefore, 50 placement attempts do not imply 50 distinct non-overlapping obstacles. The slope and box generators use a nominal central-platform or clearance-width parameter of $1.2\,\mathrm{m}$.

The rough, slope, and box heightfields use a nominal $0.2\,\mathrm{m}$ source sampling scale. They are subsequently resampled to 31×31 vertices per tile to enforce the flat-border geometry. Consequently, final surface extrema and local slopes can differ from their nominal generation parameters.

Map boundaries and evaluation labels. The main grid is surrounded by a $10\,\mathrm{m}$-wide region of additional rough and box terrain, followed by a $2\,\mathrm{m}$-wide flat outer band. The additional terrain interiors are elevated by $0.05\,\mathrm{m}$, while their tile borders remain flat. Perimeter walls are $5\,\mathrm{m}$ high and $0.05\,\mathrm{m}$ thick.

Evaluation covers all seven initial terrain types and all ten terrain rows. Terrain labels describe the initial tile of each rollout. The robot may subsequently enter neighboring tiles, so these labels do not necessarily describe the entire executed trajectory.

A.7 Reference Motions and Evaluation Coverage

All evaluations use fixed model weights without evaluation-time optimization. Reference motions and executed states are compared at 50 Hz. References are placed in the evaluation scene using the initial alignment, and the reference clock advances throughout each rollout. Errors are evaluated against the corresponding reference time without post-hoc temporal alignment.

LaFAN. The LaFAN evaluation uses 40 source motion sequences. For the four continuous metrics, the three-method comparison covers 902 segments, including short terminal segments, whereas the ablation evaluation covers 862 complete segments. Both recovery evaluations use the same 862 complete 10-second segments. A segment is an evaluation window and need not contain a complete skill, particularly when a falling-and-getting-up sequence spans multiple segments.

SOMA. The continuous-metric evaluation uses all 994 selected recordings, evaluated over their full durations. The original 120 Hz references are resampled to 50 Hz. For recovery evaluation, 180 recordings contain at least one complete 10-second window. These recordings yield 281 consecutive, non-overlapping 10-second segments after incomplete terminal windows are discarded. Thus, the continuous metrics and recovery rates use different evaluation sets.

The SOMA recordings provide an additional motion set beyond the 40 LaFAN sequences used in the reported training configuration.

Released PMT references. The additional PMT-reference evaluation uses 297 segments from 99 released files. These files contain only one unique joint-position sequence and therefore do not represent 99 independent motion skills. For recovery evaluation, the third segment starts at zero-based frame 999 rather than frame 1000. This produces 501 reference states spanning 500 control intervals, or exactly 10 seconds, without extrapolation. The third segment consequently overlaps the preceding segment by one control interval.

Evaluation counts. Each continuous-metric condition is evaluated with five paired trials. Recovery uses one trial per condition at each of four force magnitudes. All conditions cover seven initial terrain types and ten terrain rows. Table 10 reports the scheduled counts per model.

Table 10: Main evaluation coverage per model. Continuous metrics use five trials per motion–terrain–row condition. Recovery uses one trial at each of four force magnitudes.

Evaluation	Continuous metrics		Recovery	
	Motions/segments	Episodes	Segments	Episodes
LaFAN comparison	902	315,700	862	241,360
LaFAN ablation	862	301,700	862	241,360
SOMA	994	347,900	281	78,680
PMT references	297	103,950	297	83,160

A.8 Continuous Tracking Metrics

Let T_i denote the number of recorded post-action states in episode i. The following metrics first produce one scalar per episode. The reference and executed quantities are expressed using the same joint ordering and scene-coordinate convention.

Joint-angle error. For the 29 actuated joints, we compute

$$E_{\mathrm{jpos},i} = \frac{1}{29T_i} \sum_{t=1}^{T_i} \sum_{j=1}^{29} \left| q_{i,t,j} - q_{i,t,j}^{\mathrm{ref}} \right|. \tag{34}$$

The unit is radians. Despite the notation E_{jpos}, this metric measures joint-angle MAE, not Cartesian joint-position error.

Root horizontal-velocity error. We use simulator ground-truth root velocity rather than the policy's velocity estimate:

$$e_{i,t}^{v} = \left\| \mathbf{v}_{i,t,xy}^{\mathrm{sim}} - \mathbf{v}_{i,t,xy}^{\mathrm{ref}} \right\|_2, \qquad E_{\mathrm{vel},i}^{\mathrm{root}} = \frac{1}{T_i} \sum_{t=1}^{T_i} e_{i,t}^{v}. \tag{35}$$

Both vectors are expressed in world horizontal coordinates. The unit is $\mathrm{m/s}$. The metric compares velocity vectors, not their scalar speeds.

Root-heading error. Heading is obtained from the horizontal projection of the root's forward axis. Let $\psi_{i,t}$ and $\psi_{i,t}^{\mathrm{ref}}$ denote the executed and reference headings. The shortest absolute angular difference is

$$e_{i,t}^{h} = \frac{180}{\pi} \left| \mathrm{atan2}\left(\sin \Delta\psi_{i,t}, \cos \Delta\psi_{i,t} \right) \right|, \qquad \Delta\psi_{i,t} = \psi_{i,t} - \psi_{i,t}^{\mathrm{ref}}. \tag{36}$$

Thus, $e_{i,t}^{h} \in [0, 180]$ degrees. For the continuous metric, a frame is valid only when both forward-axis horizontal projections have norm greater than 10^{-6}. Writing $\mathcal{V}_i$ for this set,

$$E_{\mathrm{head},i}^{\mathrm{root}} = \frac{1}{|\mathcal{V}_i|} \sum_{t \in \mathcal{V}_i} e_{i,t}^{h}. \tag{37}$$

The valid-frame fraction is recorded separately. An episode with no valid heading frames has an undefined heading metric rather than zero error. The same validity rule is applied to every method, although the resulting masks may differ because executed orientations differ.

Foot sliding. FS measures horizontal displacement of the left and right ankle-roll link origins during consecutive contact frames. Let $\mathbf{p}_{i,f,t}^{xy}$ be the world horizontal position of foot-link origin $f \in \{L, R\}$, measured in meters. Contact is detected using the recorded vertical contact-force component:

$$c_{i,f,t} = \mathbf{1}\left[|F_{i,f,t}^{z}| > 1\,\mathrm{N} \right]. \tag{38}$$

We compute

$$\mathrm{FS}_i = \frac{100}{2(T_i - 1)} \sum_{f \in \{L,R\}} \sum_{t=2}^{T_i} c_{i,f,t-1} c_{i,f,t} \left\| \mathbf{p}_{i,f,t}^{xy} - \mathbf{p}_{i,f,t-1}^{xy} \right\|_2. \tag{39}$$

The factor 100 converts meters to centimeters. FS is reported in cm and represents contact-masked horizontal displacement averaged over both feet and all recorded control intervals at 50 Hz. The denominator includes non-contact intervals. Displacement magnitudes are summed before averaging, rather than using the net displacement between endpoints.

FS is a foot-link sliding proxy rather than a direct measurement of physical contact-point slip. Foot rotation around a stationary contact can move the measured link origin, and the score also depends on contact occupancy. If either foot has no valid consecutive-contact interval, the episode-level FS is undefined and is excluded from its metric-specific mean, with the valid episode count retained. Undefined values are not replaced by zero.

A.9 Perturbation and Recovery Protocol

Perturbations. Each recovery rollout spans 10 seconds. A horizontal force is applied to the pelvis from $t = 2.0$ to $t = 2.2$ seconds while the reference clock continues to advance. The force magnitude is selected from $\{50, 100, 150, 200\}$ N. For direction θ, the world-frame force is

$$\mathbf{F} = F \begin{bmatrix} \cos\theta & \sin\theta & 0 \end{bmatrix}^\top . \tag{40}$$

Directions are deterministically generated from the trial seed and condition identifiers. Matched conditions share the same force direction across methods and force magnitudes. One trial is evaluated for each motion–terrain–row–force condition.

Recovery criterion. At each recorded state, define

$$b_i(t) = \mathbf{1}\left[e_i^h(t) < 30^\circ \ \wedge \ e_i^v(t) \le 0.5\,\mathrm{m/s} \ \wedge \ \text{both errors are finite and valid}\right]. \tag{41}$$

The velocity threshold applies to the norm of the horizontal error vector at each time step. It is not applied separately to its components or to an error accumulated over time. Undefined heading frames do not satisfy the recovery criterion.

Let $t_{\mathrm{off}} = 2.2\,\mathrm{s}$ and $t_{\mathrm{end}} = 10\,\mathrm{s}$. For a recovery deadline $w \in \{3, 5\}\,\mathrm{s}$, a trial succeeds if the full observation window is completed without reset and there exists an entry time τ satisfying

$$\begin{aligned} &t_{\mathrm{off}} \le \tau \le t_{\mathrm{off}} + w, \\ &t_{\mathrm{end}} - \tau \ge 1\,\mathrm{s}, \\ &b_i(t) = 1 \quad \text{at every recorded state from } \tau \text{ through } t_{\mathrm{end}}. \end{aligned} \tag{42}$$

Therefore, a temporary return to tolerance followed by another violation is insufficient. Trials remaining within tolerance throughout the post-perturbation period also count as successful.

Writing $S_i^{(w)}$ for this binary outcome, the main-table rate is

$$\mathrm{RR@}w = 100\,\frac{1}{N_{\mathrm{scheduled}}} \sum_{i=1}^{N_{\mathrm{scheduled}}} S_i^{(w)}. \tag{43}$$

All scheduled trials remain in the denominator. Trials that reset or fail to complete the observation window cannot satisfy the success criterion. In the main text and main tables, RR denotes RR@5s. We retain explicit deadline labels when reporting both RR@3s and RR@5s in supplementary results.

Tracking violations during the one-second interval before force application are recorded separately. They do not automatically invalidate a subsequent recovery. This metric measures recovery or maintenance of heading and horizontal-velocity tracking. It does not require returning to the original horizontal path, satisfying a joint-angle threshold, or recovering an upright whole-body pose. Consequently, it should not be interpreted as a general fall-recovery success rate.

A.10 Aggregation and Incomplete Rollouts

For each continuous metric, let E_i be the episode-level value and N its number of valid episodes. We report

$$\mu = \frac{1}{N}\sum_{i=1}^{N} E_i, \qquad s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(E_i - \mu)^2}. \tag{44}$$

Table 11: Category-balanced SOMA recovery results. Rates are recording- and category-macro averages. Each condition uses one trial per force magnitude. Pre-push violation denotes failure to remain within the tracking tolerances throughout the one-second pre-push interval.

Method	RR@3s (%)	RR@5s (%)	Pre-push violation (%)
Three-method comparison			
UFO	22.50	28.38	31.70
PMT	0.87	1.07	98.98
P-BFM	56.99	69.50	20.30
Ablation			
w/o Heading Instr.	35.15	45.44	21.52
w/o Velocity Instr.	37.03	52.31	32.09
w/o Perception	54.64	66.04	21.54
P-BFM	57.03	69.56	20.31

Episodes are equally weighted across segments or recordings, initial terrains, terrain rows, and evaluation repeats. The reported standard deviation describes variation among episode-level means. It is neither a within-episode temporal standard deviation nor a standard deviation across five independently trained models.

The main tables do not use source-motion macro-averaging. Consequently, source motions contributing more segments receive greater weight in segment-based evaluations. For full-recording SOMA evaluation, each scheduled recording condition receives equal weight regardless of recording duration. Metric-specific validity counts are retained because undefined heading or FS values can change the effective sample size.

Ordinary tracking failures do not trigger automatic resets when simulation can continue. Continuous metrics therefore include subsequent execution rather than only the successful portion of a rollout. In the SOMA comparison, 8 PMT episodes and 219 UFO episodes ended early at the global environment boundary. Their recorded-time means remain included. These episodes have shorter observation windows, which limits their direct comparability with full-duration episodes. Program errors, non-finite states, and policy tracking failures are recorded separately.

A.11 Category-Balanced SOMA Recovery Supplement

We additionally evaluate a category-balanced SOMA subset to examine recovery under a different motion composition. The subset contains 30 recordings from each of six categories: walking, running, dancing, jumping, low-pose motions, and sports. The 180 recordings comprise 161 motion-name groups and yield 255 complete 10-second segments. Incomplete terminal windows are discarded. This evaluation uses the same perturbation and recovery criteria as Appendix A.9.

For this supplement, outcomes are first averaged over conditions and segments within each recording. Recording-level rates are then averaged within each category, followed by an equal-weight average over the six categories. This differs from the episode-weighted aggregation of the 281-segment SOMA recovery set used in the main tables.

The balanced results are reported separately and do not replace the original SOMA main-table values. In particular, PMT's high pre-push violation rate indicates that its recovery score also reflects tracking difficulty already present before perturbation. Differences in the aggregate rates therefore cannot be attributed solely to the applied forces.

A.12 Real-World Depth Acquisition and Preprocessing

The camera and depth-processing settings recorded for the final real-world deployment are summarized in Table 12. Automatic exposure was disabled, with exposure time fixed at $1600\,\mu$s. The depth stream was configured at 480×270 pixels and $60\,$Hz. The depth encoder ran on the CPU with a configured update frequency of $50\,$Hz. The camera acquisition rate and encoder update rate are distinct settings; the latter is configurable.

Table 12: Recorded camera and depth-processing settings for the final real-world deployment. Gain and infrared emitter power are reported as device setting values.

Parameter	**Setting**
Automatic exposure	Disabled
Exposure time	$1600\,\mu\mathrm{s}$
Gain	16
Infrared emitter power	240
Depth-stream resolution	480×270 pixels
Depth-stream frame rate	$60\,\mathrm{Hz}$
Depth-encoder device	CPU
Configured encoder rate	$50\,\mathrm{Hz}$
Vertical image flip	Disabled
Spatial filter	Disabled
Temporal filter	Disabled
Hole-filling filter	Disabled
Gaussian smoothing	Enabled, 3×3 kernel
Encoder input per frame	18×32 pixels
Maximum depth	$2.5\,\mathrm{m}$
Invalid-depth replacement	$2.5\,\mathrm{m}$
Normalized input range	$[0, 1]$

Depth preprocessing. Depth measurements were mapped to the training ray layout using nearest-neighbor sampling and then cropped to 18×32 pixels per frame. The maximum depth was set to $2.5\,\mathrm{m}$, and invalid measurements were replaced with this maximum value, consistent with the training preprocessing convention. Depth inputs were normalized to $[0, 1]$ before encoding. The final deployment enabled 3×3 Gaussian smoothing, with spatial, temporal, and hole-filling filters disabled.