BFMTRACK: BEHAVIOR FOUNDATION MODELS FOR PHYSICS-BASED MOTION TRACKING

Anonymous authors
Paper under double-blind review

ABSTRACT

Behavior Foundation Models (BFMs) offer a promising path toward universal physics-based control by learning a latent space of physically plausible behaviors from reward-free interactions, regularized toward large-scale motion data. Each latent specifies a stationary policy, and existing BFM inference procedures map a single task, such as goal-reaching or a state-based reward, to a single latent. Time-varying objectives such as motion tracking instead require a sequence of latents. Current BFM tracking heuristics obtain this sequence by averaging goal embeddings over a look-ahead window, which fails to capture the nuances of highly dynamic motions. We propose BFMTrack, a test-time optimization method that refines a sequence of latents with simulator feedback while keeping the pre-trained BFM frozen. BFMTrack substantially improves tracking accuracy over zero-shot BFM inference and surpasses a general motion tracker trained with a dedicated tracking reward. Under external disturbances, BFMTrack also outperforms a specialized tracker trained on a single motion, while preserving the BFM's natural recovery behavior. We further demonstrate BFMTrack on sparse references and deploy it on a real humanoid robot.

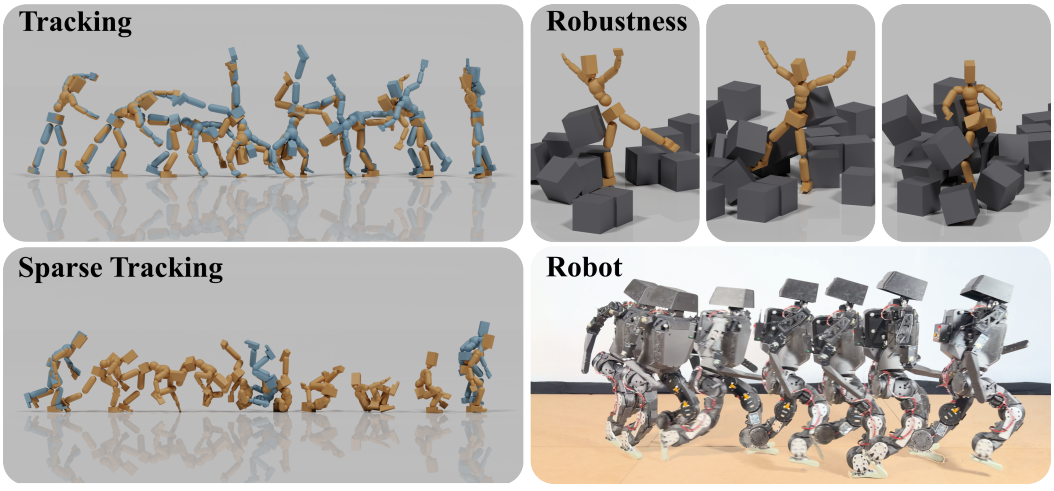


Figure 1: BFMTrack performs motion tracking (top left), recovers from repeated box impacts (top right), generates a roll from a sparse reference (bottom left), and executes motions on a real humanoid robot (bottom right). Reference (blue) and simulated (orange) poses are overlaid at several time steps.

1 INTRODUCTION

Unsupervised pre-training followed by task-specific adaptation has emerged as a dominant paradigm in language (Radford et al., 2019; Brown et al., 2020) and vision (Caron et al., 2021; He et al., 2022; Assran et al., 2023). Unsupervised reinforcement learning (RL) seeks to extend this paradigm to sequential decision-making (Touati & Ollivier, 2021; Touati et al., 2023; Frans et al., 2024).

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.

Behavior Foundation Models (BFMs) follow this approach, learning a latent-conditioned policy from reward-free interactions with the environment. After pre-training, BFMs map reward functions to policy latents, enabling stationary tasks, i.e., tasks specified by a single time-invariant reward, to be solved without further policy learning (Pirotta et al., 2024; Tirinzoni et al., 2025). Recent work extends BFMs to whole-body humanoid control, with a single shared policy solving tasks such as velocity control and goal reaching in simulation (Tirinzoni et al., 2025) and on hardware (Li et al., 2026). Beyond zero-shot task inference, these models exhibit two properties that are difficult to obtain with task-specific training: *naturalness* and *robustness*. Regularizing the policy toward a large motion-capture dataset during pre-training yields human-like behavior without specialized style rewards. Moreover, pre-training exposes BFMs to a diverse range of tasks and states, including those far from the reference motions, enabling them to recover from external disturbances (Li et al., 2026). Both properties are valuable for physics-based characters and robots, where deviations from the intended motion are unavoidable.

Motion tracking, i.e., following a time-varying reference trajectory with a physics-based character, is a long-standing problem in animation and robotics (Zordan & Hodgins, 2002; Nakaoka et al., 2003; Liu et al., 2010; Peng et al., 2018). Dedicated trackers, whether trained per motion or on large motion datasets, use the tracking error as their central learning signal. As a result, training primarily focuses on states close to the intended motion. While BFMs generate natural and robust behaviors, their inference procedures address stationary objectives, whereas motion tracking requires following a target pose that changes at every timestep. This raises the question of how to achieve accurate motion tracking with BFMs while retaining their benefits.

Since each BFM latent specifies a stationary policy, following a time-varying reference requires a sequence of latents, one per timestep. Current BFM trackers obtain this sequence with a moving-window heuristic (Tirinzoni et al., 2025; Li et al., 2026), which averages the upcoming reference states into a single latent at each timestep. Because averaging discards the temporal order within the window, the window length creates a trade-off between context and precision. Small windows target only the next pose and thus lack the context for anticipatory movements, such as a preparatory step before a kick, whereas large windows blur fast transitions and let the policy skip ahead of the reference. Moreover, the latents are inferred from the reference alone, without simulating the resulting motion.

In this work, we propose BFMTrack, which optimizes the latent sequence with a policy gradient. Starting from the zero-shot latents, it repeatedly simulates the resulting motion and refines the full sequence to reduce the tracking error, while keeping the pre-trained BFM frozen. This restricts the optimization to behaviors already provided by the BFM, preserving its natural style and closed-loop response to disturbances without an additional style reward.

We validate BFMTrack on three hours of held-out motions, consistently improving tracking over zero-shot BFM inference and outperforming a general motion tracker (Shi et al., 2026). Under external disturbances, BFMTrack outperforms a specialized tracker trained on a single motion (Peng et al., 2018), while preserving the BFM's natural recovery behavior, such as absorbing box impacts and standing up after falling (Fig. 1 top and supplementary video). Finally, we demonstrate BFMTrack on sparse references and deploy it on a real humanoid robot (Fig. 1 bottom).

2 Related Work

Behavior Foundation Models Behavior Foundation Models (BFMs) learn a family of policies from reward-free interaction with the environment and map rewards to policies after training in a zero-shot manner (Touati & Ollivier, 2021; Touati et al., 2023; Park et al., 2024; Bagatella et al., 2026a; Jajoo et al., 2026; Zheng et al., 2025). The zero-shot capabilities of BFMs have been applied to tasks such as imitation learning (Pirotta et al., 2024), active task inference (Rupf et al., 2026), and constrained RL (Hugessen et al., 2025). FB-CPR (Tirinzoni et al., 2025) scales a BFM to whole-body humanoid control, including motion tracking through a look-ahead heuristic, and BFM-Zero (Li et al., 2026) deploys it on hardware.

Test-time adaptation of a frozen BFM has been explored for stationary rewards, where LoLA (Sikchi et al., 2025) adapts a single latent with policy gradients. BFM-Zero (Li et al., 2026) additionally showcases robust single-leg standing under an added torso weight by adapting a single latent via

CEM (Rubinstein, 1999) and tracking of a single motion under altered ground friction via sampling-based trajectory adaptation of a latent sequence (Xue et al., 2025). BFMTrack extends this line of work to motion tracking, where existing approaches fix the latent sequence using a look-ahead heuristic, and evaluates performance against dedicated motion trackers.

Motion Tracking Motion tracking is typically addressed by policies trained with tracking rewards. Specialized trackers train one policy per motion and reach high accuracy, but require retraining for every new reference (Peng et al., 2018; He et al., 2025; Xie et al., 2025). General motion trackers instead train a single policy on large motion datasets and track unseen references without retraining (Luo et al., 2023; Serifi et al., 2024; Luo et al., 2026; Shi et al., 2026). Latent-space controllers (Yao et al., 2024; Peng et al., 2022; Tessler et al., 2023) also expose a latent interface, but rely on a trained high-level policy to use it, whereas BFMTrack optimizes a latent sequence for a frozen BFM without any further training. We provide a more in-depth review of motion tracking and control in Sec. A.

3 Preliminaries

Behavior Foundation Models We consider a reward-free Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, P, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, transition kernel $P(\mathrm{d}\mathbf{s}'|\mathbf{s}, \mathbf{a})$, and discount factor $\gamma \in [0, 1)$. A BFM pre-trained on this MDP learns a state embedding $B : \mathcal{S} \to \mathbb{R}^d$ and a family of latent-conditioned policies $\{\pi_{\mathbf{z}}\}_{\mathbf{z}\in\mathbb{R}^d}$ (Touati et al., 2023). Each policy $\pi_{\mathbf{z}}$ is trained to maximize the discounted return for the intrinsic reward

$$r_{\mathbf{z}}(\mathbf{s}) = \mathbf{z}^\top C_B^{-1} B(\mathbf{s}) \quad \text{where} \quad C_B = \mathbb{E}_{\mathbf{s}\sim\rho}[B(\mathbf{s})B(\mathbf{s})^\top] \tag{1}$$

is the uncentered covariance of embeddings under the pre-training state distribution ρ, assumed invertible. Each latent therefore specifies a reward, which favors states whose embedding correlates with $\mathbf{z}$, together with a policy that approximately maximizes it. Together, the latents parameterize a space of reward functions, which pre-training structures to cover as many relevant tasks as possible.[1] FB-CPR (Tirinzoni et al., 2025) trains each $\pi_{\mathbf{z}}$ on $r_{\mathbf{z}}$ plus a latent-conditioned adversarial reward $r_{\mathrm{disc},\mathbf{z}}$ (Ho & Ermon, 2016; Peng et al., 2021) that regularizes the policy toward natural motion from a motion-capture dataset. BFM-Zero (Li et al., 2026) further adds control regularization rewards r_{reg}, which penalize torques and action rates to keep the motion smooth and within the robot's limits, and trains with domain randomization for sim-to-real transfer, necessary for hardware deployment. These auxiliary rewards do not change which reward a latent represents, but they shape which policies are learned.

After pre-training, a new state-dependent reward $r : \mathcal{S} \to \mathbb{R}$ can be mapped to the latent whose reward $r_{\mathbf{z}}$ best approximates it in the least-squares sense:

$$\mathbf{z}_r = \arg\min_{\mathbf{z}\in\mathbb{R}^d} \mathbb{E}_{\mathbf{s}\sim\rho}\left[\left(r(\mathbf{s}) - \mathbf{z}^\top C_B^{-1} B(\mathbf{s})\right)^2\right] = \mathbb{E}_{\mathbf{s}\sim\rho}\left[r(\mathbf{s})B(\mathbf{s})\right]. \tag{2}$$

The inferred latent selects the policy $\pi_{\mathbf{z}_r}$ without further policy learning or planning. The resulting performance on r depends on both the reward approximation and the quality of the learned policy.

BFMTrack relies on two properties of this construction. First, a latent does not encode a pose or a short motion but an entire policy. The policy $\pi_{\mathbf{z}}$ defines a closed-loop action for every state in $\mathcal{S}$, including states far from any reference. Hence, even a fixed latent yields a feedback controller that reacts to disturbances. Second, the embedding of a single state is itself a meaningful latent. Consider a reward concentrated at a target pose $\mathbf{g}$ that rewards no other state. For this reward, Eq. (2) gives $\mathbf{z}_{\mathbf{g}} = B(\mathbf{g})$ up to a positive factor that leaves $\pi_{\mathbf{z}}$ unchanged, since scaling a reward does not change its optimal policy (Touati & Ollivier, 2021). The reward induced by this latent, $r_{\mathbf{z}_{\mathbf{g}}}(\mathbf{s}) = B(\mathbf{g})^\top C_B^{-1} B(\mathbf{s})$, is therefore the BFM's own goal-reaching reward, and $\pi_{B(\mathbf{g})}$ drives the character toward $\mathbf{g}$. At test time, BFMTrack uses only the latent-conditioned policy $\pi_{\mathbf{z}}$ and the state embedding B.

Motion Tracking with BFMs Given a reference motion, i.e., a sequence of target poses $\{\mathbf{g}_t\}_{t=0}^T$ with $\mathbf{g}_t \in \mathcal{S}$, we seek a controller that induces a state trajectory $\{\mathbf{s}_t\}_{t=0}^T$ that closely follows the reference. Following common practice on physical robots (Serifi et al., 2024; Grandia et al., 2024),

[1] Methods typically learn either B or the features $C_B^{-1}B$ directly.

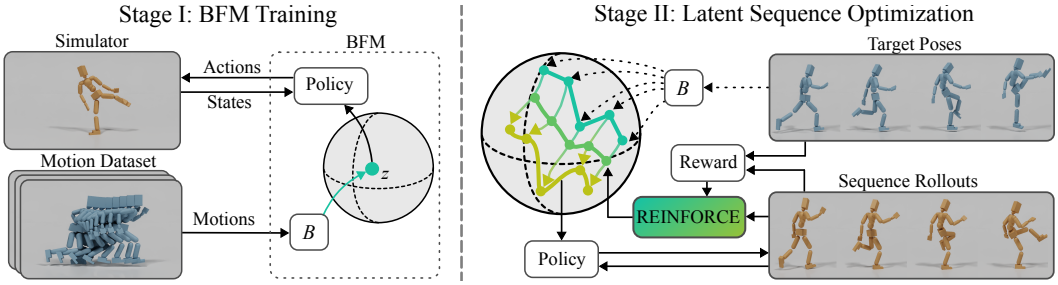


Figure 2: BFMTrack optimizes the latent sequence of a frozen BFM. Left: BFM pre-training from reward-free interaction, regularized toward a motion dataset. Right: the state embedding B maps the target poses to the zero-shot latent sequence (turquoise). BFMTrack uses REINFORCE to refine this sequence with the tracking reward of simulated rollouts.

we track the reference locally, i.e., in a path frame aligned with the character's root position and heading, so that drift in global position and heading is not penalized. The controller is initialized at $\mathbf{s}_0 = \mathbf{g}_0$ and has access to the full reference before execution, allowing it to use the entire horizon.

An ideal tracking reward favors being at $\mathbf{g}_t$ at time t. This reward is non-stationary, and since the BFM's state excludes time and the reference, it has no projection onto the reward family in Eq. (2). Existing BFM trackers (Tirinzoni et al., 2025; Li et al., 2026) instead build on the empirical-reward (ER) imitation latent of Pirotta et al. (2024), which is designed for a *stationary* behavior: it places a reward at every state of a demonstration and averages their embeddings. Applied to the next L reference poses, this yields

$$\mathbf{z}_t = \frac{1}{\min(L,\, T-t)} \sum_{\tau=t+1}^{\min(t+L,\, T)} B(\mathbf{g}_\tau), \qquad t = 0, \dots, T-1. \tag{3}$$

The ER latent is thus a moving average of the goal-reaching latents of the upcoming poses, and the resulting policy solves a stationary surrogate of the tracking problem. For $L = 1$, $\pi_{\mathbf{z}_t}$ maximizes the discounted goal reward of $\mathbf{g}_{t+1}$: it reaches and holds that pose, with no incentive to arrive on time or to anticipate what follows. For larger L, the poses lose their temporal order and blur into one target, so the policy may skip ahead or settle on an intermediate pose. The window length thus trades holding back against skipping ahead, and no fixed L resolves this for every part of a reference.

4 BFMTrack: Latent Sequence Optimization

The ER latents in Eq. (3) represent a reference motion as a sequence of latents $\{\mathbf{z}_t\}_{t=0}^{T-1}$, one policy per timestep, but they are inferred from the reference alone (Sec. 3). Even with an optimal choice of window, the latents are never corrected by the motion they produce, and each $\pi_{\mathbf{z}}$ is itself only approximately optimal for its own reward (Sikchi et al., 2025). BFMTrack therefore treats the heuristic sequence as an initialization and optimizes it against the tracking return measured in simulation (Fig. 2, right). The pre-trained BFM (Fig. 2, left) stays frozen, so that every latent remains a policy from the pre-trained family and retains its naturalness and robustness. Specifically, BFMTrack optimizes the latent sequence $\mathbf{z}_{0:T-1}$ to maximize the tracking return,

$$J(\mathbf{z}_{0:T-1}) = \mathbb{E}\left[\sum_{t=0}^{T-1} r_t \,\middle|\, \mathbf{a}_t \sim \pi_{\mathbf{z}_t}(\cdot \mid \mathbf{s}_t), \mathbf{s}_{t+1} \sim P(\cdot \mid \mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_0 = \mathbf{g}_0\right], \tag{4}$$

where r_t denotes the per-timestep tracking reward, and the expectation is taken over simulation rollouts of the latent-conditioned policy.

We formulate this latent optimization as a reinforcement learning problem, where the state is the timestep t with deterministic transitions $t \to t+1$, the action is the latent $\mathbf{z}_t$, and the reward $r_t = r(\mathbf{s}_{t+1}, \mathbf{g}_{t+1})$ is returned by the simulator after executing the frozen BFM policy $\pi_{\mathbf{z}_t}$. We parametrize the time-indexed latent policy with a learnable Gaussian mean $\boldsymbol{\mu}_t$ per timestep and a fixed standard deviation σ,

$$q_{\boldsymbol{\mu}}(\mathbf{z}_t \mid t) = \mathcal{N}(\boldsymbol{\mu}_t, \sigma^2 \mathbf{I}). \tag{5}$$

We optimize $q_{\boldsymbol{\mu}}$ with REINFORCE (Williams, 1992), similar to Sikchi et al. (2025) for single-latent adaptation. Optimizing with small gradient steps is further motivated by the finding that the mapping from a latent to its behavior is smooth under entropic regularization (Bagatella et al., 2026b) and by smooth interpolation between latents in BFM-Zero (Li et al., 2026), which suggests that the tracking objective varies smoothly with the latents.

We warm-start the latent sequence with the goal-reaching latents $\boldsymbol{\mu}_t = B(\mathbf{g}_{t+1})$. At each iteration, we sample $n = 1, \dots, N$ latent sequences $\mathbf{z}^{(n)}_{0:T-1}$ from $q_{\boldsymbol{\mu}}$, roll them out in N parallel simulation environments to obtain states $\mathbf{s}^{(n)}_{1:T}$ and rewards $r^{(n)}_{0:T-1}$, and compute the discounted returns $R^{(n)}_t = \sum_{j=t}^{T-1} \gamma^{j-t} r^{(n)}_j$. The discounting reflects that the influence of a latent decreases with temporal distance. Following Sikchi et al. (2025) and recent fine-tuning methods in language modeling (Ahmadian et al., 2024), we compute the advantage with a leave-one-out baseline (Kool et al., 2019) over the N rollouts and update each mean with Adam (Sec. D) along

$$\Delta\boldsymbol{\mu}_t = \frac{1}{N} \sum_{n=1}^{N} \overline{R}^{(n)}_t \nabla_{\boldsymbol{\mu}_t} \log \mathcal{N}\big(\mathbf{z}^{(n)}_t \mid \boldsymbol{\mu}_t, \sigma^2 \mathbf{I}\big) \quad \text{where} \quad \overline{R}^{(n)}_t = R^{(n)}_t - \frac{1}{N-1} \sum_{i \neq n} R^{(i)}_t. \tag{6}$$

The objective in Eq. (4) supports any per-timestep tracking reward, such as dedicated motion-tracking rewards or the BFM's own goal-reaching reward, $r_t = B(\mathbf{g}_{t+1})^\top C_B^{-1} B(\mathbf{s}_{t+1})$ from Sec. 3, as evaluated in Sec. 5.

Temporally Correlated Exploration To search efficiently over latent sequences, we coordinate exploration across neighboring timesteps by sampling the noise perturbation of each latent sequence $\boldsymbol{\varepsilon}^{(n)}_{0:T-1}$ from temporally correlated *colored noise* with power spectral density $S(f) \propto 1/f^\beta$ (Timmer & Koenig, 1995; Pinneri et al., 2021). We set $\beta = 1$ (*pink noise*), which has proven effective for exploration in continuous-control RL (Eberhard et al., 2023), and ablate β in Sec. 5.3. The correlation affects only the sampling distribution, while the update in Eq. (6) retains the marginal Gaussian score (Hollenstein et al., 2024; Raffin et al., 2022), which biases the updates toward temporally smooth changes of the latent sequence.

5 Experiments

We first evaluate how BFMTrack compares with zero-shot BFM inference and dedicated trackers for motion tracking under undisturbed conditions (Sec. 5.1) and external disturbances (Sec. 5.2). We then ablate the tracking reward, exploration noise, optimizer, and optimization budget (Sec. 5.3). Finally, we demonstrate applications to sparse references and a real humanoid (Sec. 5.4).

Setup All quantitative experiments use a 69-DoF simulated SMPL humanoid (Loper et al., 2015) (Sec. E.1) and the AMASS (Mahmood et al., 2019) training and test split of Tirinzoni et al. (2025), whose test split contains 990 clips with more than three hours of motion. We additionally deploy BFMTrack on Lima, a custom 20-DoF humanoid robot (Sec. 5.4, Tab. 2).

BFMTrack and ER (Pirotta et al., 2024) share a single frozen BFM. We pre-train it on the training split with FB-CPR (Tirinzoni et al., 2025) and control regularization r_{reg} (Li et al., 2026) (Sec. 3, Sec. F). ER infers the latents zero-shot with the best look-ahead horizon, $L = 5$ (Sec. B.2). BFMTrack instead optimizes them for each clip in 24 iterations with the PHC tracking reward r_{PHC} (Luo et al., 2023) (Sec. D).

We compare against two dedicated trackers (Sec. G). The Generalist Tracker and the Specialist Tracker follow the first-stage tracker of GPC (Shi et al., 2026) and DeepMimic (Peng et al., 2018), respectively. For a fair comparison, we implement both methods in our experimental setting, using the same observations, control regularization r_{reg}, and tracking reward r_{PHC}. The Generalist Tracker uses the same AMASS training split and compute budget as the BFM (4.5 days on a single NVIDIA RTX 4090), while the Specialist Tracker requires 4 hours of training per test clip. Given this per-clip cost, we evaluate the specialist on the *hardest subset*: the 30 test clips with the highest EMD under zero-shot ER, corresponding to the motions that are most challenging for the BFM.

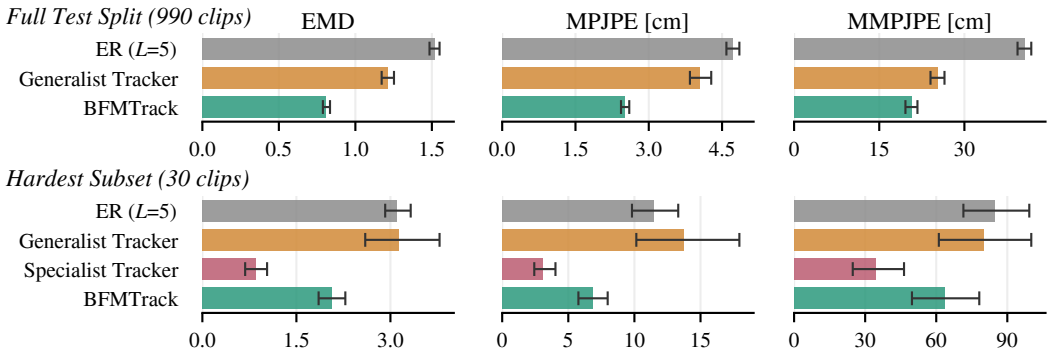


Figure 3: Mean metrics on the full AMASS test split (990 clips, top) and its hardest subset (30 clips, bottom); lower is better, and whiskers give 95% bootstrap confidence intervals of the mean over clips. ER is zero-shot BFM inference with the look-ahead latents of Eq. (3). BFMTrack outperforms ER and the generalist on all metrics, while the specialist remains more accurate on the hardest subset. See Tab. 7 for numerical results.

Metrics We report three metrics, defined in Sec. C. The Earth Mover's Distance (EMD) (Rubner et al., 2000) measures the distance between the distributions of visited and reference poses without requiring temporal alignment. It therefore captures how well the performed motion preserves the style and structure of the reference. The mean per-joint position error (MPJPE) and the mean max per-joint position error (MMPJPE) measure tracking accuracy in the local path frame, where MMPJPE averages the worst-case deviation of each clip and therefore serves as a threshold-free proxy for tracking success (Luo et al., 2023).

5.1 Motion Tracking

We first evaluate motion tracking accuracy on the full test split against zero-shot BFM inference with ER and the Generalist Tracker baseline (Fig. 3). BFMTrack outperforms both on all metrics, reducing the MPJPE to $2.51\,\mathrm{cm}$ compared to $4.05\,\mathrm{cm}$ for Generalist Tracker and $4.72\,\mathrm{cm}$ for ER. Since ER and BFMTrack use the same frozen BFM, the improved tracking shows that the latent space contains more accurate sequences than those selected by the fixed-window heuristic of prior BFM-based motion tracking (Li et al., 2026; Tirinzoni et al., 2025). This mirrors the observation of Sikchi et al. (2025) for single latents and stationary rewards. BFMTrack finds these sequences by optimizing the latent sequence for 24 iterations, requiring about $9\,\mathrm{s}$ of optimization time per second of motion on a single consumer GPU. The supplementary video also shows the improvement qualitatively in the cartwheel example.

On the hardest subset, BFMTrack reduces the MPJPE of ER by about 40% ($6.8\,\mathrm{cm}$ vs. $11.4\,\mathrm{cm}$), whereas the Generalist Tracker no longer improves on ER. We additionally compare against the Specialist Tracker, which is the most accurate method ($3.1\,\mathrm{cm}$) after 4 hours of training on each clip. However, as shown in Sec. 5.2, this advantage does not extend to tracking under external disturbances.

5.2 Robustness to External Disturbances

Although all methods are trained without external disturbances, we evaluate their robustness on the hardest subset under identical disturbance schedules: strong periodic pushes to the pelvis and weaker pushes to randomly selected body parts. More details are provided in Sec. C.1.

As shown in Fig. 4, BFMTrack achieves the lowest absolute error under both disturbance regimes, and its degradation from undisturbed tracking is small and comparable to that of ER. In contrast, both Generalist Tracker and Specialist Tracker degrade substantially, and the specialist goes from the most to the least accurate method. We attribute the robustness of BFMs to their pre-training, during which the policy is exposed to a diverse state distribution across many behaviors, including episodes that start from fallen states (Tirinzoni et al., 2025). Consistent with this, the Generalist Tracker, trained on a large dataset, is more robust than the Specialist Tracker.

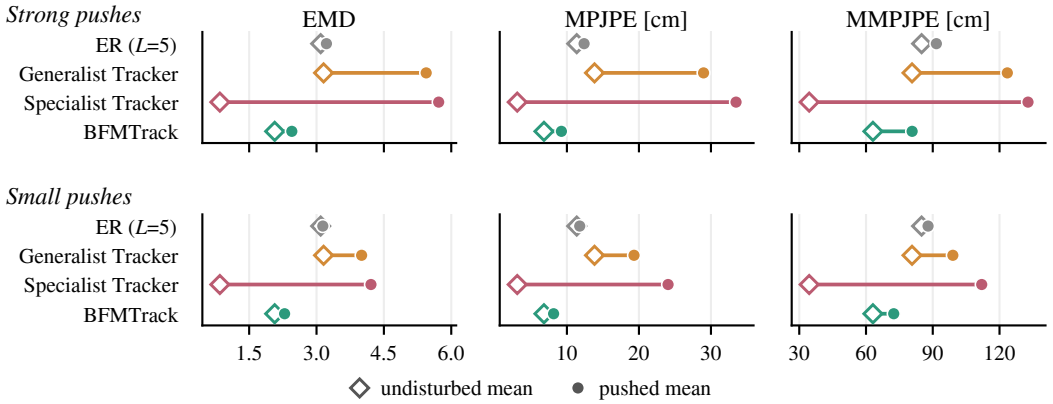


Figure 4: Results on the hardest subset with strong pelvis pushes (top) and small pushes to random body parts (bottom). Open markers denote undisturbed means and filled markers denote means under disturbances over 256 push schedules, each averaged across clips. BFMTrack achieves the lowest positional errors under both push regimes. See Tab. 8 for numerical results.

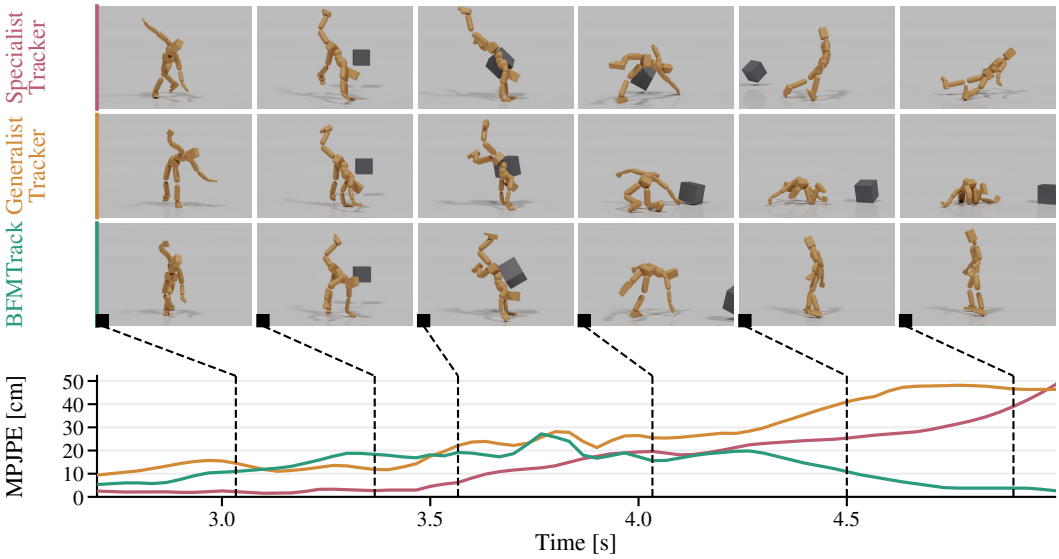


Figure 5: Snapshots at the same instants for each method during a cartwheel (top) and MPJPE over time (bottom; snapshot times dashed). Each character is struck at the same point of its motion by an $8\,\mathrm{kg}$ box thrown at $15\,\mathrm{m\,s^{-1}}$, which transfers $110\,\mathrm{N\,s}$ over $100\,\mathrm{ms}$, an impulse comparable to a strong push (Tab. 8). BFMTrack recovers from the impact, while both dedicated trackers fall.

A cartwheel example in which we throw a box at the character further illustrates this difference (Fig. 5 and supplementary video). While both dedicated trackers fall after the impact, BFMTrack recovers and resumes tracking.

5.3 Ablations

Tracking Objective We evaluate the effect of the reward on latent optimization by comparing the BFM's own goal-reaching reward $r_{B(\mathbf{g}_{t+1})}$ to the PHC tracking reward r_{PHC}. As shown in Fig. 6, r_{PHC} improves tracking accuracy across all noise colors. A plausible explanation is that the finite-dimensional state embedding favors spatially smooth, low-frequency rewards and blurs local reward details (Touati & Ollivier, 2021; Touati et al., 2023). The sharply localized tracking objective is therefore represented poorly. Directly optimizing r_{PHC} with simulator feedback avoids this representation and reaches the more accurate sequences of Sec. 5.1. This agrees with Sikchi

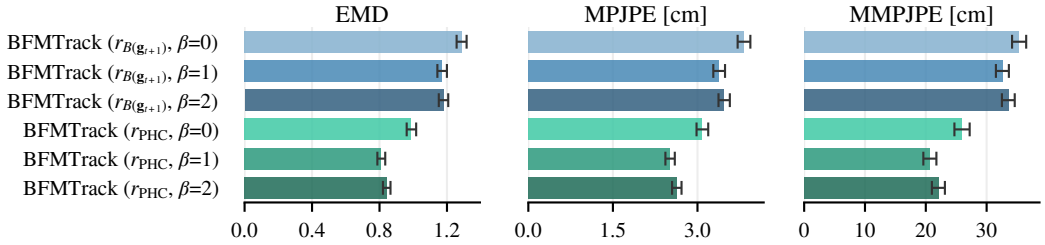


Figure 6: Mean results on the full test split for the BFM's goal-reaching reward $r_{B(\mathbf{g}_{t+1})}$ (blue) and r_{PHC} (green), varying the noise exponent β; lower is better, and whiskers give 95% bootstrap confidence intervals of the mean over clips. The PHC reward improves accuracy for all noise colors, and pink noise ($\beta = 1$) is best for both rewards. See Tab. 9 for numerical results.

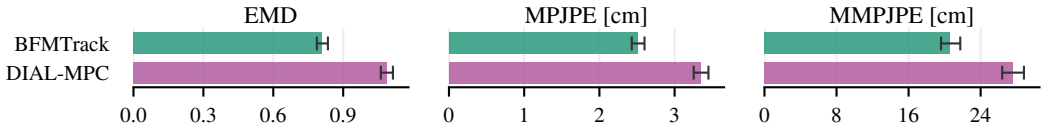


Figure 7: Mean results on the full test split for BFMTrack and the sampling-based update of DIAL-MPC (Xue et al., 2025), both with r_{PHC} and the same warm start; lower is better, and whiskers give 95% bootstrap confidence intervals of the mean over clips. BFMTrack is more accurate on all metrics. See Tab. 10 for numerical results.

et al. (2025), who find for stationary tasks that adapting to the true reward outperforms adapting to its reconstruction by the BFM.

Exploration Noise We evaluate temporally correlated *colored noise* for sampling each noise sequence, as illustrated in Fig. 6. For both rewards, pink noise ($\beta = 1$) achieves the best tracking accuracy, followed by red noise ($\beta = 2$) and uncorrelated white noise ($\beta = 0$). This result confirms the findings of Eberhard et al. (2023) in our setting.

Latent Optimizer The objective in Eq. (4) does not prescribe the policy-gradient update of Eq. (6). We therefore compare against the sampling-based update of DIAL-MPC (Xue et al., 2025), which moves each latent to the softmax-weighted average of the perturbed samples (Sec. G). BFM-Zero (Li et al., 2026) uses a similar update to adapt a single motion to altered ground friction. Warm start, reward r_{PHC}, and a budget of 24 iterations with 128 parallel rollouts remain identical. As shown in Fig. 7, BFMTrack is more accurate than DIAL-MPC on all metrics. We attribute this to the smoother landscape over BFM latents (Sec. 4), which favors a local gradient estimate over the global coverage that annealing provides for raw actions (Xue et al., 2025).

Optimization Budget We evaluate longer optimization with r_{PHC} on the hardest subset (Fig. 9 in Sec. B). Our default of 24 iterations captures about two-thirds of the improvement achieved at 128 iterations, providing a practical balance between accuracy and computation. A larger budget may pay off for particularly difficult motions, but it does not close the gap to the Specialist Tracker in the undisturbed setting.

5.4 Applications

Sparse References In sparse reference tracking, target poses are specified only at a subset of frames, leaving the motion between them unconstrained. We adapt BFMTrack by assigning a single latent to each interval between sparse frames. Since the PHC reward is designed for dense motion tracking, we instead optimize the latent sequence using the BFM's goal-reaching reward $r_{B(\mathbf{g}_{t+1})}$. The reward is defined only at the sparse reference frames, with zero reward elsewhere. Each latent is initialized with the zero-shot goal latent of the upcoming reference frame. We extract sparse references at increasing sparsity (Sec. C.2). As shown in Fig. 8, latent optimization improves the tracking accuracy at the sparse reference frames over zero-shot inference. We qualitatively show an example of a rolling sequence in Fig. 1 and the supplementary video.

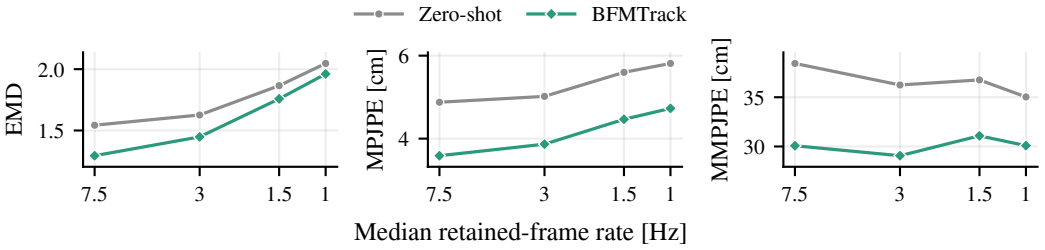


Figure 8: Tracking at increasing reference sparsity. Mean metrics over the AMASS test split for the zero-shot goal latents and BFMTrack optimized with the BFM's goal-reaching reward, at median retained-frame rates of 7.5 Hz, 3.0 Hz, 1.5 Hz and 1.0 Hz; lower is better. EMD is computed against the full dense reference, while MPJPE and MMPJPE are restricted to the retained frames, so they measure accuracy at fewer poses as sparsity increases. Latent optimization improves all metrics at every reference sparsity. See Tab. 11 for numerical results.

While latent optimization can compensate for missing reference frames, the BFM's motion prior becomes insufficient to consistently produce coherent motion between widely spaced targets as the references become sparser. In the supplementary video, we illustrate this with a walking motion at increasing sparsity, where the character produces repeated steps with the same foot.

Hardware Deployment We deploy BFMTrack on Lima, a custom 20-DoF humanoid robot, using an embodiment-specific BFM trained with retargeted motions (Müller et al., 2026) and domain randomization for sim-to-real transfer (Sec. E.1). Latent sequences are optimized in simulation and subsequently executed onboard by the policy at 50 Hz.

As shown in Fig. 1 and the supplementary video, the robot performs motions including running and dancing, demonstrating transfer of the optimized latent sequences to hardware. The qualitative results also show the recovery behavior observed in simulation. The robot adapts to external pushes and resumes tracking, and can recover from an unplanned fall even though the reference contains no recovery sequence.

6 Conclusion

We presented BFMTrack, which extends a frozen BFM from stationary tasks to motion tracking by optimizing a sequence of latents with simulator feedback. Our results show that zero-shot inference leaves much of a BFM's tracking ability unused: optimizing the latent sequence of the same frozen model improves all metrics over zero-shot inference and outperforms a generalist tracker trained with a dedicated tracking reward. Because the BFM stays frozen, BFMTrack also retains the robustness acquired during pre-training: under external disturbances, it achieves the lowest errors of all methods, whereas a per-clip specialist, the most accurate method without disturbances, degrades the most. The same optimization extends to sparse references and transfers to a humanoid robot.

Limitations and Future Work While BFMTrack provides an effective framework for precise motion tracking, several limitations offer exciting directions for future research. First, because the optimization runs a physics simulator in the loop, our approach currently operates offline and does not yet enable reactive, real-time motion tracking. Re-optimizing over a receding horizon with a learned world model (Li et al., 2025) could trade accuracy for reactivity and enable online adaptation of BFMTrack on the robot. Second, our evaluation covers a single BFM family, FB-CPR (Tirinzoni et al., 2025), which relies on large human motion datasets for training. Extending BFMTrack to other BFMs and to non-humanoid characters remains future work. Finally, in the sparse reference setting, gait naturalness can degrade, particularly in the stepping pattern. Looking ahead, targets that specify only part of the state, such as end-effector positions (Tessler et al., 2024), or targets in the world frame, combined with BFMs that observe objects or terrain, could extend BFMTrack to complex loco-manipulation (Luo et al., 2024a) and dynamic locomotion in challenging terrain (Zhang et al., 2026).

Reproducibility statement

Sec. 4 describes BFMTrack and Sec. D lists its hyperparameters. Sec. E specifies the embodiments, observations, and auxiliary rewards, and Sec. F the BFM pre-training. Sec. G details all baselines, and Sec. C defines the metrics, the protocol for external disturbances, and the extraction of sparse references. Sec. H reports the numerical results behind the figures of Sec. 5, with confidence intervals or standard deviations. Sec. B adds the optimization budget, the look-ahead horizon of ER, and results on the training split. All quantitative experiments use the public AMASS dataset (Mahmood et al., 2019) with the training and test split of Tirinzoni et al. (2025), which in turn uses filtering from Luo et al. (2023).

AI use statement

Generative AI tools were used for limited implementation and editorial assistance. Coding assistants were used to generate draft software code, and AI tools were used to assist with manuscript formatting, readability improvements, consistency checks, identification of potentially relevant literature, and reference formatting. All AI-assisted outputs were independently reviewed, verified, and, where appropriate, modified by the authors. Generative AI tools did not contribute to the conception or development of the research ideas, hypotheses, methods, experimental design, analyses, interpretation of results, conclusions, or other substantive scientific or technical contributions reported in this work. All such contributions were made by the human authors, who retain full responsibility for the content of the manuscript and all associated artifacts.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Jinseok Bae, Younghwan Lee, Donggeun Lim, and Young Min Kim. PLT: Part-wise latent tokens as adaptable motion priors for physically simulated characters. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–10, 2025.

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. TD-JEPA: Latent-predictive representations for zero-shot reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=SzXDuBN8M1.

Marco Bagatella, Thomas Rupf, Georg Martius, and Andreas Krause. Soft forward-backward representations for zero-shot reinforcement learning with general utilities, 2026b. URL https://arxiv.org/abs/2602.06769.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Rafael Cathomen, Mayank Mittal, Marin Vlastelica, and Marco Hutter. Divide, discover, deploy: Factorized skill learning with symmetry and style priors. In *Conference on Robot Learning*, 2025.

Zhiyang Dou, Xuelin Chen, Qingnan Fan, Taku Komura, and Wenping Wang. C·ASE: Learning conditional adversarial skill embeddings for physics-based characters. In *SIGGRAPH Asia 2023 Conference Papers*, pp. 1–11, 2023.

Onno Eberhard, Jakob Hollenstein, Cristina Pinneri, and Georg Martius. Pink noise is all you need: Colored noise exploration in deep reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Unsupervised zero-shot reinforcement learning via functional reward encodings. In *International Conference on Machine Learning (ICML)*, 2024.

Levi Fussell, Kevin Bergamin, and Daniel Holden. SuperTrack: Motion tracking for physically simulated characters using supervised learning. *ACM Transactions on Graphics (TOG)*, 40(6): 1–13, 2021.

Ruben Grandia, Espen Knoop, Michael A. Hopkins, Georg Wiedebach, Jared Bishop, Steven Pickles, David Müller, and Moritz Bächer. Design and Control of a Bipedal Robotic Character. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.103.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbabu, Chaoyi Pan, et al. ASAP: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. In *Proceedings of Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.066.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Jakob Hollenstein, Georg Martius, and Justus Piater. Colored noise in PPO: Improved exploration and performance through correlated action sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12466–12472, 2024. doi: 10.1609/aaai.v38i11.29139.

Xiaoyu Huang, Takara Truong, Yunbo Zhang, Fangzhou Yu, Jean Pierre Sleiman, Jessica Hodgins, Koushil Sreenath, and Farbod Farshidian. Diffuse-CLoC: Guided diffusion for physics-based character look-ahead control. *ACM Transactions on Graphics (TOG)*, 44(4), July 2025a. ISSN 0730-0301. doi: 10.1145/3731206. URL https://doi.org/10.1145/3731206.

Yiming Huang, Zhiyang Dou, and Lingjie Liu. ModSkill: Physical character skill modularization. In *IEEE/CVF International Conference on Computer Vision*, 2025b.

Adriana Hugessen, Harley Wiltzer, Cyrus Neary, Amy Zhang, and Glen Berseth. Zero-shot constraint satisfaction with forward-backward representations. In *Workshop on Reinforcement Learning Beyond Rewards@ Reinforcement Learning Conference 2025*, 2025.

Pranaya Jajoo, Harshit Sikchi, Siddhant Agarwal, Amy Zhang, Scott Niekum, and Martha White. Regularized latent dynamics prediction is a strong baseline for behavioral foundation models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! *Deep Reinforcement Learning Meets Structured Prediction Workshop at ICLR*, 2019.

Chenhao Li, Andreas Krause, and Marco Hutter. Robotic world model: A neural network simulator for robust policy optimization in robotics. *arXiv preprint arXiv:2501.10100*, 2025.

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. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jkhl2oI0g5.

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. *Science Robotics*, 11(117):eadx8924, 2026. doi: 10.1126/scirobotics.adx8924.

Hung Yu Ling, Fabio Zinno, George Cheng, and Michiel Van De Panne. Character controllers using motion VAEs. *ACM Trans. Graph.*, 39(4), August 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392422. URL https://doi.org/10.1145/3386569.3392422.

Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu. Sampling-based contact-rich motion control. *ACM Trans. Graph.*, 29(4), July 2010. ISSN 0730-0301. doi: 10.1145/1778765.1778865. URL https://doi.org/10.1145/1778765.1778865.

Nian Liu, Zilong Zhang, Zi Wang, Tengyu Liu, Hongzhao Xie, Xinyi Tong, Libin Liu, Yaodong Yang, and Zhaofeng He. Learning uniformly distributed embedding clusters of stylistic skills for physically simulated characters. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 10343–10351, 2025.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6), November 2015. ISSN 0730-0301. doi: 10.1145/2816795.2818013. URL https://doi.org/10.1145/2816795.2818013.

David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. Perpetual Humanoid Control for Real-time Simulated Avatars. In *2023 IEEE/CVF International Conference on Computer Vision*, 2023. doi: 10.1109/ICCV51070.2023.01000.

Zhengyi Luo, Jinkun Cao, Sammy Christen, Alexander Winkler, Kris Kitani, and Weipeng Xu. Omnigrasp: Grasping diverse objects with simulated humanoids. *Advances in Neural Information Processing Systems*, 37:2161–2184, 2024a.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. In *International Conference on Learning Representations*, 2024b.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Jinhyung Park, David Sami, Zi Wang, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, 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*, 11(117):eaed4592, 2026. doi: 10.1126/scirobotics.aed4592.

Li-Ke Ma, Zeshi Yang, Xin Tong, Baining Guo, and KangKang Yin. Learning and exploring motor skills with spacetime bounds. In *Computer Graphics Forum*, pp. 251–263. Wiley Online Library, 2021.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pp. 5442–5451, October 2019.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac Gym: High performance GPU-based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://openreview.net/forum?id=fgFBtYgJQX_.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *The Twelfth International Conference on Learning Representations*, 2024.

David Müller, Agon Serifi, Sammy Christen, Ruben Grandia, Espen Knoop, and Moritz Bächer. ReActor: Reinforcement learning for physics-aware motion retargeting. *ACM Trans. Graph.*, 45(4), July 2026. ISSN 0730-0301. doi: 10.1145/3811378. URL https://doi.org/10.1145/3811378.

Shinichiro Nakaoka, Atsushi Nakazawa, Kazuhito Yokoi, Hirohisa Hirukawa, and Katsushi Ikeuchi. Generating whole body motions for a biped humanoid robot from captured human dances. In *2003 IEEE International Conference on Robotics and Automation*, volume 3, pp. 3905–3910. IEEE, 2003.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with Hilbert representations. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, August 2018. ISSN 0730-0301, 1557-7368. doi: 10.1145/3197517.3201311.

Xue Bin Peng, Ze Ma, Pieter Abbeel, Sergey Levine, and Angjoo Kanazawa. AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. *ACM Transactions on Graphics (TOG)*, 40(4):1–20, August 2021. ISSN 0730-0301, 1557-7368. doi: 10.1145/3450626.3459670.

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. ASE: Large-Scale Reusable Adversarial Skill Embeddings for Physically Simulated Characters. *ACM Transactions on Graphics (TOG)*, 41(4):1–17, July 2022. ISSN 0730-0301, 1557-7368. doi: 10.1145/3528223.3530110.

Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning*, pp. 1049–1065, 2021.

Matteo Pirotta, Andrea Tirinzoni, Ahmed Touati, Alessandro Lazaric, and Yann Ollivier. Fast imitation via behavior foundation models. In *International Conference on Learning Representations*, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 2019.

Antonin Raffin, Jens Kober, and Freek Stulp. Smooth exploration for robotic reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1634–1644, 2022. URL https://proceedings.mlr.press/v164/raffin22a.html.

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

Yossi Rubner, Carlo Tomasi, and Leonidas Guibas. The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 2000. doi: 10.1023/A:1026543900054.

Thomas Rupf, Marco Bagatella, Marin Vlastelica, and Andreas Krause. Optimistic task inference for behavior foundation models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=m5byThUSNE.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *The Fourth International Conference on Learning Representations*, 2016.

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

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

Younggyo Seo, Carmelo Sferrazza, Haoran Geng, Michal Nauman, Zhao-Heng Yin, and Pieter Abbeel. FastTD3: Simple, fast, and capable reinforcement learning for humanoid control. *arXiv preprint arXiv:2505.22642*, 2025.

Agon Serifi, Ruben Grandia, Espen Knoop, Markus Gross, and Moritz Bächer. VMP: Versatile motion priors for robustly tracking motion on physical characters. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '24, pp. 1–11. Eurographics Association, 2024. doi: 10.1111/cgf.15175.

Yi Shi, Yifeng Jiang, Chen Tessler, and Xue Bin Peng. GPC: Large-scale generative pretraining for transferable motor control. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers (SIGGRAPH)*, 2026.

Harshit Sikchi, Andrea Tirinzoni, Ahmed Touati, Yingchen Xu, Anssi Kanervisto, Scott Niekum, Amy Zhang, Alessandro Lazaric, and Matteo Pirotta. Fast adaptation with behavioral foundation models. In *Reinforcement Learning Conference*, 2025. URL https://openreview.net/forum?id=soeW8RGo1N.

Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. CALM: Conditional Adversarial Latent Models for Directable Virtual Characters. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*, pp. 1–9, July 2023. doi: 10.1145/3588432.3591541.

Chen Tessler, Yunrong Guo, Ofir Nabati, Gal Chechik, and Xue Bin Peng. MaskedMimic: Unified physics-based character control through masked motion inpainting. *ACM Transactions on Graphics (TOG)*, 43(6):1–21, 2024.

Jens Timmer and Michel Koenig. On generating power law noise. *Astronomy and Astrophysics*, 300:707, 1995.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9sOR0nYLtz.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23, 2021.

Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MYEap_OcQI.

Takara Everest Truong, Michael Piseno, Zhaoming Xie, and Karen Liu. PDP: Physics-based character animation via diffusion policy. In *SIGGRAPH Asia 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. doi: 10.1145/3680528.3687683. URL https://doi.org/10.1145/3680528.3687683.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph.*, 39(4), August 2020. ISSN 0730-0301. doi: 10.1145/3386569.3392381. URL https://doi.org/10.1145/3386569.3392381.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Physics-based character controllers using conditional VAEs. *ACM Transactions on Graphics (TOG)*, 41(4), 2022.

Yan Wu, Korrawe Karunratanakul, Zhengyi Luo, and Siyu Tang. UniPhys: Unified planner and controller with diffusion for flexible physics-based character control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Weiji Xie, Jinrui Han, Jiakun Zheng, Huanyu Li, Xinzhe Liu, Jiyuan Shi, Weinan Zhang, Chenjia Bai, and Xuelong Li. KungfuBot: Physics-based humanoid whole-body control for learning highly-dynamic skills. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 62406–62433. Curran Associates, Inc., 2025. doi: 10.52202/085713-2089. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/5a0e51901cff2b42d379ec7869603e91-Paper-Conference.pdf.

Pei Xu, Kaixiang Xie, Sheldon Andrews, Paul G Kry, Michael Neff, Morgan McGuire, Ioannis Karamouzas, and Victor Zordan. AdaptNet: Policy adaptation for physics-based character control. *ACM Transactions on Graphics (TOG)*, 42(6):1–17, 2023.

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

Heyuan Yao, Zhenhua Song, Baoquan Chen, and Libin Liu. ControlVAE: Model-based learning of generative controllers for physics-based characters. *ACM Transactions on Graphics (TOG)*, 41 (6):1–16, 2022.

Heyuan Yao, Zhenhua Song, Yuyang Zhou, Tenglong Ao, Baoquan Chen, and Libin Liu. MoConVQ: Unified physics-based motion control via scalable discrete representations. *ACM Transactions on Graphics (TOG)*, 43(4):1–21, 2024.

Ye Yuan and Kris Kitani. Residual force control for agile human behavior imitation and extended motion synthesis. In *Advances in Neural Information Processing Systems*, 2020.

Fatemeh Zargarbashi, Jin Cheng, Dongho Kang, Robert Sumner, and Stelian Coros. RobotKeyframing: Learning locomotion with high-level objectives via mixture of dense and sparse rewards. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 916–932. PMLR, 2025. URL https://proceedings.mlr.press/v270/zargarbashi25a.html.

Zewei Zhang, Kehan Wen, Michael Xu, Junzhe He, Chenhao Li, Takahiro Miki, Clemens Schwarke, Chong Zhang, Xue Bin Peng, and Marco Hutter. Learning whole-body humanoid locomotion via motion generation and motion tracking. *IEEE Robotics and Automation Letters*, 11(9):10194–10201, 2026. doi: 10.1109/LRA.2026.3710365.

Chongyi Zheng, Jens Tuyls, Joanne Peng, and Benjamin Eysenbach. Can a MISL fly? Analysis and ingredients for mutual information skill learning. In *International Conference on Learning Representations*, 2025.

Victor Brian Zordan and Jessica K. Hodgins. Motion capture-driven simulations that hit and react. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '02, pp. 89–96, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581135734. doi: 10.1145/545261.545276. URL https://doi.org/10.1145/545261.545276.

A Extended Related Work

Physics-Based Motion Tracking Tracking reference motions on physics-based characters is a long-standing topic in animation and robotics. The canonical tracking specialist, DeepMimic (Peng et al., 2018), trains a phase-synchronized RL policy with hand-crafted tracking rewards; this tight clip-policy coupling implicitly captures reference dynamics for anticipatory control, but requires retraining per clip. Generalist trackers scale the number of motions a tracker can imitate via multiple experts (Won et al., 2020; Huang et al., 2025b), model-based components (Fussell et al., 2021; Yao et al., 2024; Li et al., 2025), learned motion priors (Luo et al., 2023; Won et al., 2022; Serifi et al., 2024), diffusion-based policies (Liao et al., 2026; Huang et al., 2025a; Truong et al., 2024; Wu et al., 2025), or large-scale training (Luo et al., 2026; Tessler et al., 2024; Shi et al., 2026). In all of these, the tracking reward is the training signal, and the policy is evaluated on how closely it follows the reference.

Latent-Space Motion Controllers Several controllers expose a latent interface over which downstream tasks are solved, and they differ mainly in how the latent space is learned. One approach is to encode reference motions into latents, with a kinematic motion model (Ling et al., 2020), a conditional adversarial objective (Tessler et al., 2023; Dou et al., 2023), distillation from a general tracker (Luo et al., 2024b), part-wise tokens (Bae et al., 2025), or a tracking objective over a look-ahead window (Serifi et al., 2024; Luo et al., 2026; Shi et al., 2026). ControlVAE (Yao et al., 2022) and MoConVQ (Yao et al., 2024) learn the latent jointly with a world model. Skill-discovery methods do not encode the reference and instead maximize mutual information between a latent and the states it induces (Eysenbach et al., 2019; Peng et al., 2022; Cathomen et al., 2025), with later work structuring that skill space (Liu et al., 2025). AdaptNet (Xu et al., 2023) instead adapts a pre-trained controller to new dynamics. In all of these, a downstream task is solved by training a closed-loop high-level policy that observes the character state and emits latents, guided by a task reward and usually by an additional style reward. The latent space itself is trained to reconstruct motion or to be diverse. BFMTrack differs in both respects: it optimizes a fixed, state-independent latent sequence without training any high-level network, and it does so over a BFM whose latent space is trained to cover a family of rewards, so that each latent is an approximately optimal policy for its own reward and no style term is needed.

Sparse References Synthesizing physically valid motion from sparse constraints has been addressed by spacetime optimization and its learned successors (Ma et al., 2021), and more recently by policies trained to reach sparse targets: MaskedMimic (Tessler et al., 2024), which trains a masked-conditioning policy to inbetween arbitrary subsets of joints and frames. RobotKeyframing (Zargarbashi et al., 2025) trains a legged-robot policy to satisfy a variable number of timed pose targets with a mixture of dense and sparse rewards. Both train the inbetweening behavior explicitly, with the sparse objective as part of the training signal. BFMTrack instead obtains the motion between retained frames from the frozen BFM's prior, by masking the tracking reward to the retained frames (Sec. 5.4).

B Additional Results

B.1 Optimization Budget

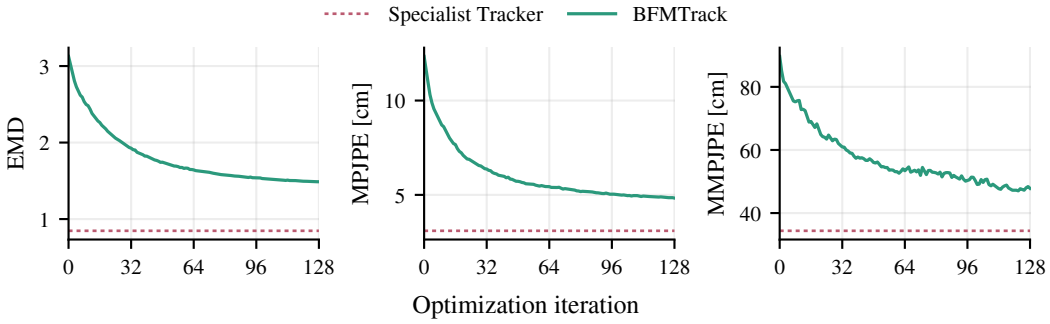
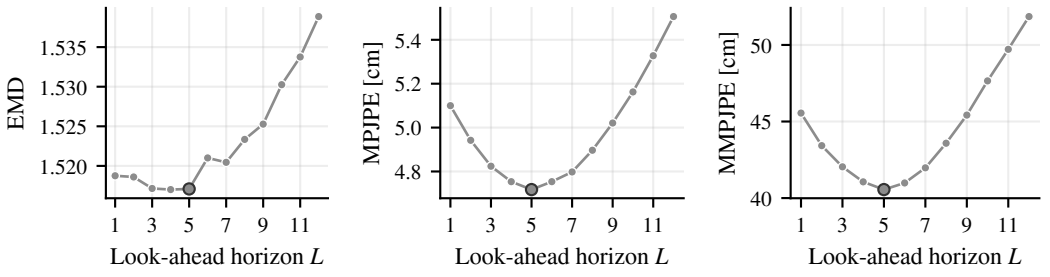


Figure 9: Additional optimization improves tracking with diminishing returns after 64 iterations. We extend latent optimization to 128 iterations on the hardest subset.

B.2 Zero-Shot Tracking Horizon

The ER baseline introduces the look-ahead horizon L as a hyperparameter. Fig. 10 sweeps L on the full test split: a short horizon lacks the anticipatory context needed to prepare for fast transitions, whereas a long horizon averages too far ahead. $L = 5$ gives the best positional accuracy and is used as the comparator throughout the main text. Fig. 11 illustrates the effect of the look-ahead horizon on a kicking motion.



(a) EMD (b) MPJPE (c) MMPJPE

Figure 10: A look-ahead horizon of $L = 5$ gives the best zero-shot positional accuracy, as short horizons lack anticipation and long ones average too far ahead. We report mean EMD, MPJPE and MMPJPE over the 990 AMASS test clips as a function of the horizon L in control frames, with the BFM fixed.

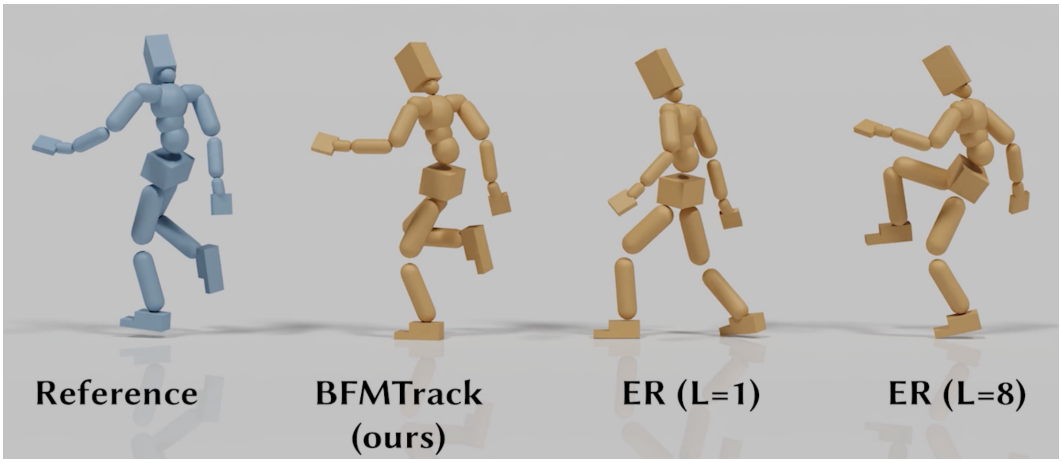


Figure 11: Comparison of BFMTrack against ER with $L = 1$ and $L = 8$ for tracking a kicking reference motion. A short horizon lacks the necessary context to prepare for the kick, exhibiting a lag, whereas the long horizon averages too far ahead, leading to a premature kicking motion.

B.3 Motion Tracking

In addition, we evaluate the frozen Generalist Tracker checkpoint and ER ($L = 5$) on the full AMASS training split (Tab. 1).

We train the Generalist Tracker for over 60,000 PPO iterations, with reward and success curves shown in Fig. 12. Reward is measured on adaptively sampled training motions, while success is averaged equally over all training chunks in the periodic evaluations used to update their sampling probabilities.

Table 1: The Generalist Tracker also outperforms ER on the training split. We report mean metrics over all 8,901 AMASS training clips for the generalist and zero-shot ER with $L = 5$. Best scores are bold; second-best scores are underlined.

Method	EMD ↓	MPJPE [cm] ↓	MMPJPE [cm] ↓
ER ($L = 5$)	<u>1.51</u>	<u>4.7</u>	<u>41.1</u>
Generalist Tracker	**1.16**	**3.8**	**23.8**

C Experimental Protocols and Metrics

Metric Definitions For body positions $\mathbf{p}_{tj}$ and references $\hat{\mathbf{p}}_{tj}$ in the local path frame, let $e_{tj} = \|\mathbf{p}_{tj} - \hat{\mathbf{p}}_{tj}\|_2$. Each clip contributes $\frac{1}{TJ}\sum_{t,j} e_{tj}$ to MPJPE and $\max_{t,j} e_{tj}$ to MMPJPE, and tables average these contributions equally across clips; in the sparse setting, t is restricted to the retained frames. EMD is the optimal-transport cost between uniform distributions over concatenated body positions and 6D orientations in the local path frame, with root height retained, and is reported in native feature units.

C.1 Robustness to External Disturbances

The strong regime applies horizontal pelvis pushes of $500\,\mathrm{N}$ for $200\,\mathrm{ms}$ every second, delivering $100\,\mathrm{N\,s}$ per push in a random direction. The small regime applies pushes of fixed magnitude $50\,\mathrm{N}$ and fixed duration $100\,\mathrm{ms}$ (three $30\,\mathrm{Hz}$ control steps) in uniformly sampled three-dimensional directions. At every control step, each of the 24 body links independently starts such a push with probability $1/30$, so pushes are unrestricted: any number may be active at once, and simultaneous pushes on the same link add their forces. The small regime excludes the first and last 5 frames, and the same sampled force schedule is replayed for every method. The undisturbed reference values

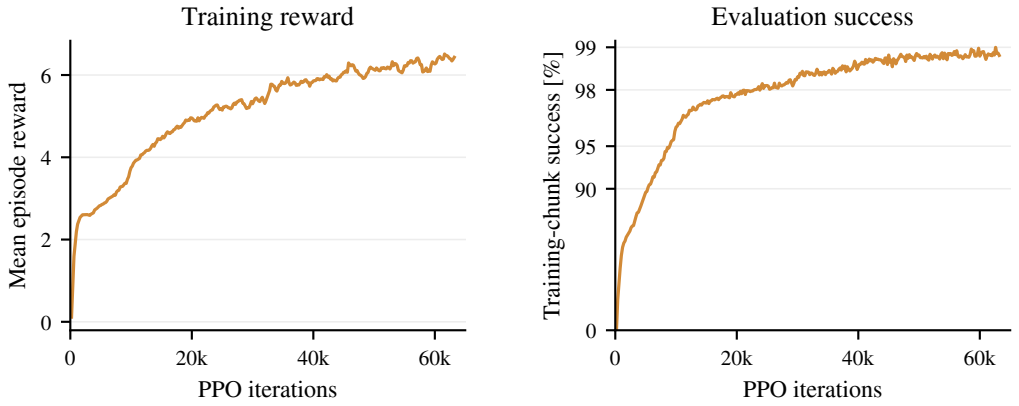


Figure 12: Generalist training progress over 60,000 PPO iterations. Left: mean episode reward. Right: success on training chunks, evaluated deterministically every 200 iterations. Each chunk (up to 10 s) starts at its first reference frame and succeeds if no body's local-frame position error exceeds 0.5 m.

for the increases in Tab. 8 come from a matched zero-force evaluation with the same rollout harness, reference arrays, and initial states; they agree with the hardest-subset rows of Tab. 7 (single rollouts) to within the rollout noise. All policies act deterministically in this harness, so the small spread of the undisturbed runs stems from nondeterminism of the parallel physics simulation, which is negligible on most clips and visible only where a rollout is close to falling. The reported standard deviation is taken across the 256 trial averages, each averaged over the 30 clips, and therefore reflects disturbance sampling for fixed models and optimized sequences.

C.2 Extraction of Sparse References

Given a kinematic reference motion of T frames, we extract the retained frames using a difference-of-Gaussians (DoG) prominence heuristic (Lowe, 2004) on the joint kinetic energy signal.

Let $E(t) = \sum_j \dot{q}_j(t)^2$ denote the sum of squared joint velocities at frame t. We define the prominence signal as

$$p(t) = |E(t) - (G_\sigma * E)(t)|\,, \tag{7}$$

where G_σ denotes a 1D Gaussian filter with standard deviation σ, and $*$ is convolution along the time axis. We use $\sigma = 15$ frames, i.e., 0.5 s at 30 fps, which produces a smoothed baseline. The prominence $p(t)$ is therefore large at frames where the instantaneous energy deviates most from its local average, corresponding to rapid movements or brief pauses. We select frames by applying non-maximum suppression (NMS) to the prominence signal based on a minimum temporal spacing parameter Δt. Specifically, we retain a frame t if $p(t)$ is the maximum within a temporal window of size $2\Delta t$; the first and last frame of the reference are always retained. By increasing $\Delta t \in \{0.1, 0.2, 0.3, 0.5\}$ s, we generate sparse references of decreasing density. Evaluating these sparse references across all AMASS test clips yields median spacings between retained frames of $\{0.13, 0.33, 0.67, 1.0\}$ s, as shown in Fig. 13. These spacings correspond to median retained-frame rates of 7.5 Hz, 3.0 Hz, 1.5 Hz and 1.0 Hz, respectively.

D BFMTrack Details

D.1 Latent Sequence Optimization

The BFM's latents are normalized during pre-training and lie on the hypersphere of radius $\sqrt{d}$ in $\mathbb{R}^d$ with $d = 256$; the policy was only trained on latents of this norm. The latent optimization is performed over 24 iterations using $N = 128$ parallel environments, with the r_{PHC} reward unless stated otherwise. The latent means are initialized with the goal-reaching latents $\boldsymbol{\mu}_t = B(\mathbf{g}_{t+1})$.

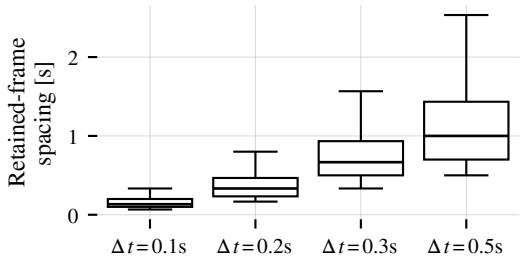


Figure 13: Temporal spacing of the retained frames in sparse references. We apply the extraction heuristic to the 990 AMASS test clips with minimum spacing parameters $\Delta t \in \{0.1, 0.2, 0.3, 0.5\}$ s. Box plots display the median and interquartile range (IQR), with whiskers extending to $1.5\times$ the IQR.

For sparse references, we initialize each segment with the zero-shot latent of its retained frame (Sec. 5.4). In each iteration, every environment is reset to the first reference pose, executes the deterministic mean action of the frozen policy for the whole clip with its sampled latent sequence, and returns the per-timestep rewards; there is no early termination. The exploration noise has standard deviation $\sigma = 0.2$ per latent coordinate and is temporally correlated as described in Sec. D.2, and the discounted returns use $\gamma = 0.97$. We use the Adam optimizer (Kingma & Ba, 2015) with a learning rate of 0.1 and modified momentum parameters $\beta = (0.8, 0.99)$. Before each Adam step, we project the gradient of every latent mean onto the tangent plane of the sphere at $\boldsymbol{\mu}_t$, removing the radial component, and after the step we rescale $\boldsymbol{\mu}_t$ back to the norm $\sqrt{d}$. The optimization therefore moves the latent sequence along the sphere on which the BFM was trained.

D.2 Colored Exploration Noise

We generate the temporally correlated noise sequences $\boldsymbol{\epsilon}^{(n)}_{0:T-1}$ following Pinneri et al. (2021) using the spectral method of Timmer & Koenig (1995). White noise corresponds to $\beta = 0$, pink noise to $\beta = 1$, and red noise to $\beta = 2$; we ablate these in Sec. 5.3.

E Humanoids and Shared Environment

E.1 Embodiments and State Representations

We evaluate our framework on two embodiments using Isaac Sim (Makoviychuk et al., 2021) as the simulator: a 69-DoF SMPL humanoid (Yuan & Kitani, 2020; Loper et al., 2015) and Lima, a custom 20-DoF humanoid robot that we use for real-world deployment. For both platforms, the actions of the control policy are position targets for proportional-derivative (PD) controllers. Because off-policy training methods require bounded action spaces for improved training convergence, we restrict the policy outputs to $[-1, 1]$ and rescale them affinely to the joint ranges enlarged by a margin (0.1 rad for SMPL), enabling the PD controller to deliberately overshoot to exert torque near the joint limits; close to a hard stop, the admissible setpoint offset is ramped down to limit impact torques.

While this general control scheme applies to both embodiments, the simulation, actuator, network, and observation settings differ. The SMPL humanoid utilizes an idealized, torque-limited actuator model (Serifi et al., 2024) running at 30 Hz with a 6-layer residual network. Lima uses actuator models of its Unitree and Dynamixel motors (Grandia et al., 2024), runs at 50 Hz, and uses a smaller 4-layer MLP to meet the real-time constraints of its onboard Intel Core i7 CPU. We summarize key parameters in Tab. 2 and describe the observations of each embodiment below.

Full state Both embodiments share a full kinematic state representation, which the state embedding B and the discriminator consume as the motion state $\mathbf{s}^{\text{mot}}_t$,

$$\mathbf{s}^{\text{full}}_t := (h_t,\, \hat{\mathbf{g}}_t,\, \mathbf{v}_t,\, \boldsymbol{\omega}_t,\, \mathbf{q}_t,\, \dot{\mathbf{q}}_t,\, \ldots, \mathbf{p}^i_t, \mathbf{R}^i_t, \mathbf{v}^i_t, \boldsymbol{\omega}^i_t, \ldots), \quad i = 1, \ldots, M,$$

Table 2: Implementation settings for the SMPL humanoid and the Lima robot. We list simulation, control, and network parameters for both embodiments. While both share the same FB-CPR pre-training recipe, their architectures and simulation parameters differ to accommodate Lima's real-world compute and hardware constraints.

Parameter	**SMPL Humanoid**	**Lima Robot**
Hardware & Simulation		
Degrees of Freedom (DoF)	69	20
Simulation rate	150 Hz	250 Hz
Control rate	30 Hz	50 Hz
PD gains k_p / k_d	100 / 1.0	15 / 0.6 (legs), 1.0 / 0.04 (arms, neck)
Domain randomization	—	Mass, CoM, friction, forces, obs. noise
Actor observation history	2 actions	3 actions, 4 joint states
Network Architecture		
Latent dimension d	256	256
Actor-Critic setup	Symmetric	Asymmetric (privileged critic)
Actor π	Residual, 6×2048	MLP, $[1024, 1024, 512, 512]$
Forward map F / critic Q	Residual, 6×2048	Residual, 6×1024
State embedding B / disc.	MLP, 3×1024	MLP, 3×1024

where h_t is the root height, $\hat{\mathbf{g}}_t$ the gravity direction in the root frame, which encodes roll and pitch, $\mathbf{v}_t$ and $\boldsymbol{\omega}_t$ the root linear and angular velocities, and $\mathbf{q}_t$ and $\dot{\mathbf{q}}_t$ the joint positions and velocities. The kinematic states $(\mathbf{p}_t^i, \mathbf{R}_t^i, \mathbf{v}_t^i, \boldsymbol{\omega}_t^i)$ of the M rigid bodies are given relative to the root position, with rotations in a 6D representation. To ensure global pose invariance, all vector quantities are expressed in a local path frame with the origin at the root, the x-axis pointing along the root heading, and the z-axis upward.

SMPL humanoid For the SMPL humanoid, the actor, the critic, and the forward map F all observe the full state, so the setup is symmetric. The actor state additionally contains the two previous actions, $\mathbf{s}_t^{\text{act}} := (\mathbf{s}_t^{\text{full}}, \mathbf{a}_{t-1}, \mathbf{a}_{t-2})$, whereas the motion state used by B and the discriminator is $\mathbf{s}_t^{\text{mot}} := \mathbf{s}_t^{\text{full}}$ without actions. No observation history and no domain randomization are used for the SMPL humanoid.

Lima For Lima, the actor is restricted to quantities measurable on the hardware,

$$\mathbf{s}_t^{\text{act}} := (\hat{\mathbf{g}}_t,\, \boldsymbol{\omega}_t^{\text{root}},\, \mathbf{a}_{t-1},\, \mathbf{a}_{t-2},\, \mathbf{a}_{t-3},\, \mathbf{q}_{t-3:t},\, \dot{\mathbf{q}}_{t-3:t}),$$

where the angular velocity $\boldsymbol{\omega}_t^{\text{root}}$ is expressed in the root frame, the three previous actions and the joint state of the four most recent control steps provide a short history, and the velocities of the arm and neck joints are masked out because their encoders provide no reliable velocity estimate. The root linear velocity and height are not observed. Uniform observation noise is added to the measured roll and pitch ($0.02\,\text{rad}$) and angular velocity ($0.1\,\text{rad}\,\text{s}^{-1}$). The motion state for B and the discriminator is the full state, $\mathbf{s}_t^{\text{mot}} := \mathbf{s}_t^{\text{full}}$.

To improve sim-to-real transfer, we apply domain randomization to the foot friction coefficient ($[0.5, 1.0]$), the link masses ($\pm5\%$), the pelvis center of mass (offsets of up to $1.5\,\text{cm}$), and continuous small external disturbance forces and torques on the head, elbows, pelvis, hips, and feet, in addition to the observation noise above. Since these quantities are not observable on the physical robot, we adopt an asymmetric actor-critic architecture. The critic and the forward map F receive a privileged state $\mathbf{s}_t^{\text{pri}}$ consisting of the full state, the sampled friction coefficients, and the currently applied disturbance forces and torques, enabling more accurate value estimation while keeping the actor restricted to observations available on hardware.

E.2 Auxiliary Regularization Rewards

Following BFM-Zero (Li et al., 2026), we apply the auxiliary regularization rewards for humanoid robot control listed in Tab. 3 during BFM training.

Table 3: Auxiliary reward terms used during BFM pre-training and tracker training. The alive reward is used only for tracker training. Entries give the reward coefficients for SMPL and Lima; dashes indicate unused terms.

Term	SMPL	Lima
Torques (legs)	-2.5×10^{-5}	-1.0×10^{-3}
Torques (arms and neck)		-1.0×10^{-2}
Acceleration (legs)	-2.5×10^{-6}	-2.5×10^{-6}
Acceleration (arms and neck)		-2.5×10^{-5}
Action rate (legs)	-1.0×10^{-2}	-30
Action rate (arms and neck)		-30
Action 2nd derivative (legs)	-1.0×10^{-2}	-20
Action 2nd derivative (arms and neck)		-20
Self-contact	—	-1.0
Hard-stop impact (CBF)	—	-1.0
Alive (tracking only)	$+1.0$	—

F BFM PRE-TRAINING

This section describes the pre-training of the frozen BFMs used throughout the paper for the two embodiments of Sec. E.1.

F.1 TRAINING

We train the BFM using FB-CPR (Tirinzoni et al., 2025) within the parallel training framework of BFM-Zero (Li et al., 2026), using 1024 simulation environments in parallel. To handle the resulting data throughput, we adopt the fast off-policy learning techniques of Seo et al. (2025), performing 16 gradient updates per step of the 1024 parallel environments, i.e., an update-to-data (UTD) ratio of $1/64$. We use a 256-dimensional latent space and train the model for 3M gradient steps. Pre-training the SMPL BFM takes about 4.5 days on a single NVIDIA RTX 4090 GPU. Tab. 4 reports the parameters that differ from Tirinzoni et al. (2025) and Li et al. (2026); all other settings are identical.

Table 4: BFM pre-training settings for SMPL and Lima. We list parameters that differ from Tirinzoni et al. (2025) and Li et al. (2026).

Parameter	SMPL	Lima
Episode length T	300	500
Seeding steps (random actions)	8	500
Fall initialization probability	0.1	0.2
Replay buffer size	$\approx$3M	$\approx$5M
Discriminator reg. coef. α_D	0.01	0.05
Auxiliary reward reg. coef. α_R	0.001	0.5
Rollout noise $\sigma \sim \mathcal{U}(\sigma_{\min}, \sigma_{\max})$ (Seo et al., 2025)	$(0.001, 0.3)$	
Actor training std	0.005	
Std clip	0.015	
Learning rates (F, π, Q_R)	3×10^{-4}	

Following Seo et al. (2025), we sample the rollout exploration noise σ independently per environment from a uniform distribution $\mathcal{U}(\sigma_{\min}, \sigma_{\max})$, refreshing on episode reset. This diversity of exploration scales improves off-policy learning without requiring a fixed noise schedule. With probability given by the fall initialization probability, an episode starts from a dataset pose with a uniformly random orientation, placed just above the ground with zero velocities, so that the BFM also covers states far from any reference motion.

F.2 Motion Data and Prioritized Sampling

The adversarial regularization in FB-CPR is driven by a filtered version (Luo et al., 2023) of the AMASS dataset (Mahmood et al., 2019), with height corrections applied to account for discrepancies in the mocap data. For the Lima robot, the dataset is retargeted via an RL-based technique (Müller et al., 2026).

Following Tirinzoni et al. (2025) and Li et al. (2026), we address the imbalance in motion difficulty within AMASS by partitioning all sequences into $10\,\mathrm{s}$ chunks and dynamically re-weighting their sampling frequencies during training. We periodically evaluate the policy's tracking performance on these chunks to prioritize the exploration of more difficult clips. For each chunk, we roll out the policy and compute the average tracking error in the root body frame by stacking all rigid-body positions

$$e = \frac{1}{T}\sum_{t=1}^{T}\left\| \begin{bmatrix} \mathbf{p}_t^1 \\ \vdots \\ \mathbf{p}_t^{|\mathcal{J}|} \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{p}}_t^1 \\ \vdots \\ \hat{\mathbf{p}}_t^{|\mathcal{J}|} \end{bmatrix} \right\|_2, \tag{8}$$

where $\mathbf{p}_t^i, \hat{\mathbf{p}}_t^i \in \mathbb{R}^3$ denote the i-th rigid-body position of the simulated and reference trajectories, respectively, for a total of $|\mathcal{J}|$ bodies.

The resulting error e dictates the updated sampling probability for each chunk, with distinct schedules and functions applied depending on the target character. For the SMPL humanoid, we re-weight the dataset every 100k gradient steps using histogram equalization. Specifically, chunk errors are sorted into 8 uniform bins between $20\,\mathrm{cm}$ and $100\,\mathrm{cm}$, and each chunk is assigned a weight inversely proportional to its bin's total count. The agent thus samples all difficulty levels about equally often. For the Lima robot, we instead re-weight the dataset every 400k gradient steps using an exponential upweighting scheme

$$w = 2^{\,6\,\cdot\,\mathrm{clamp}(e,\,200/3,\,200)\,/\,200}. \tag{9}$$

This formulation assigns exponentially higher sampling probabilities to chunks with larger tracking errors, aggressively directing the optimization toward the most challenging motions.

G Baseline Details

All methods observe the reference through the motion state $\mathbf{s}_t^{\text{mot}}$ of Sec. E.1, which B and the discriminator also consume. They differ only in the look-ahead offsets of the reference frames they read. Both trackers moreover observe the same actor state $\mathbf{s}_t^{\text{act}}$ as the BFM policy.

Zero-Shot Baseline The ER baseline uses the frozen BFM with the look-ahead latent of Eq. (3); the horizon $L = 5$ used in the main text is selected by the sweep in Sec. B.2.

Generalist Tracker To compare BFMTrack to a state-of-the-art generalist tracker, we implement the first-stage tracker of GPC (Shi et al., 2026) matching our SMPL environment, observations, regularization and tracking rewards, and compute, but otherwise using their network architectures, training curricula and hyperparameters. A target-only encoder reads the motion states of seven future reference frames at offsets of $(1, 2, 5, 7, 12, 18, 25)$ control steps, i.e., a look-ahead of $0.83\,\mathrm{s}$. We evaluate the policy after about 4.5 days of training on a single NVIDIA RTX 4090, matching the budget and hardware used for BFM pre-training (Sec. F.1). Training uses 4,096 environments and 32 steps per environment per PPO iteration. Fig. 12 shows the full training history. We follow Shi et al. (2026) and update motion sampling prioritization by evaluating the current checkpoint every 200 PPO iterations. We do so on the same $10\,\mathrm{s}$ motion chunks as used for motion sampling prioritization in BFM pre-training (Sec. F.2). Successful chunks are downweighted by 0.999^{200} with a minimum weight of $1/N_{\text{chunks}}$, while failed chunks receive weight 1. The generalist's training episodes are capped at 300 control steps, as for the BFM.

Specialist Tracker The Specialist Tracker trains a PPO policy from scratch for each evaluated clip, following DeepMimic (Peng et al., 2018). Its observations include the current actor state, the

motion state of the next reference frame, and the normalized motion phase. The actor and critic are MLPs with hidden dimensions $(512, 512, 512)$. Each policy is trained with a fixed wall-clock budget of 4 hours on a single NVIDIA RTX 4090, using 4,096 environments and 32 rollout steps per PPO iteration. This amounted to at least 5,800 PPO iterations, i.e., 7.6×10^8 environment transitions. Reference-state initialization follows the Generalist Tracker: 20% of the episodes start at the first frame and the rest at a phase sampled uniformly within the clip; a maximum body position error above $0.5\,\mathrm{m}$ terminates a training episode after a five-step grace period. Specialist episodes otherwise continue to the end of the reference clip, with no additional time limit.

Training Objectives Both trackers use the PPO (Schulman et al., 2017) implementation and adaptive learning-rate schedule of RSL-RL (Schwarke et al., 2025), with Adam's default momentum and epsilon settings. We train a single generalist policy and one specialist policy per evaluated clip, each with a single training run. Tab. 5 summarizes the training settings of both trackers. The tracking reward is computed as (Luo et al., 2023)

$$r_{\mathrm{PHC}} = 0.5\exp(-100e_p) + 0.3\exp(-10e_R) + 0.1\exp(-0.1e_v) + 0.1\exp(-0.1e_\omega), \tag{10}$$

where e_p, e_v, e_ω average squared Cartesian errors of body positions, linear and angular velocities, and e_R averages squared body rotation errors, each state expressed in its own heading frame with the horizontal root position removed. We clip the trackers' training reward so that it does not incentivize termination. The final reward is $\max(0,\, 7r_{\mathrm{PHC}} + r_{\mathrm{env}})$, where r_{env} contains the environment's control regularizers including the $+1$ alive reward. In comparison, BFMTrack uses the tracking term r_{PHC} alone.

Table 5: Training settings of the generalist and specialist trackers. Both use the same SMPL environment and tracking objective described in this section. Reference offsets are measured in control steps.

Parameter	**Generalist Tracker**	**Specialist Tracker**
Training motions	AMASS training split	One evaluated clip
Training budget	≈4.5 days	4 hours per clip
Hardware	One NVIDIA RTX 4090	
Parallel environments	4,096	
Rollout steps per environment	32	
Episode time limit	300 control steps	Reference clip end
Action distribution	Gaussian	
Initial action standard deviation	0.333	
Optimizer	Adam	
PPO epochs per iteration	5	
Minibatches per epoch	4	
PPO clipping parameter	0.2	
Value-loss coefficient	1.0	
Clipped value loss	Enabled	
Entropy coefficient	0.0	
Discount γ	0.99	
GAE (Schulman et al., 2016) parameter λ	0.95	
Initial learning rate	10^{-3}	
Learning-rate schedule	Adaptive	
Target KL divergence	0.01	
Gradient clipping norm	1.0	
First-frame initialization probability	0.2	
Otherwise initialize at	Uniformly sampled reference phase	
Termination body-position error	$> 0.5\,\mathrm{m}$ (maximum over bodies)	
Termination grace period	5 steps	
Future reference offsets	$(1, 2, 5, 7, 12, 18, 25)$	1
Extra observation	—	Phase (clip timestep t)
Target-code dimension	40	—
FSQ (Mentzer et al., 2024) levels per coordinate	9	—

DIAL-MPC Update For the optimizer ablation (Sec. 5.3) we adapt the DIAL-MPC update (Xue et al., 2025) to our setting in two ways. First, we run it open-loop over the whole clip, so that it

optimizes the same latent sequence as BFMTrack instead of re-planning a short horizon at every step. Second, DIAL-MPC scores every particle by the mean reward over its horizon, which would assign a single weight to all latents of a sequence, so we compute the discounted return per timestep ($\gamma = 0.97$, as for BFMTrack) and use it as the per-timestep score for the update of the corresponding latent. Each iteration then perturbs the incumbent sequence with Gaussian noise under the dual-loop annealing schedule of DIAL-MPC, with $\beta_1 = 0.85$ and $\beta_2 = 0.9$ and our budget of 128 particles over 24 iterations, and replaces each latent by the softmax-weighted average of the population. Samples and the updated mean are projected back to the hypersphere. The warm start and reward match BFMTrack, and Tab. 6 lists the hyperparameters.

Table 6: Hyperparameters of the DIAL-MPC update used in the optimizer ablation. The annealing parameters β_1 and β_2 follow BFM-Zero (Li et al., 2026).

Parameter	**Value**
Particles per iteration	128
Iterations	24
β_1	0.85 (Li et al., 2026)
β_2	0.9 (Li et al., 2026)
Initial noise standard deviation	1.0 (Xue et al., 2025)
Temperature λ	0.5

H Detailed Results

This section collects the numerical results behind the figures of Sec. 5, in the order of the main text.

H.1 Motion Tracking

Tab. 7 lists the means and confidence intervals plotted in Fig. 3.

Table 7: BFMTrack improves positional tracking over ER and the generalist, while the specialist remains more accurate on the hardest subset. We report mean metrics on the full AMASS test split (990 clips) and on its hardest subset (30 clips); small numbers give 95% bootstrap confidence intervals of the mean over clips. Best scores are bold; second-best scores are underlined. Fig. 3 plots these values.

Method	EMD ↓	MPJPE [cm] ↓	MMPJPE [cm] ↓
Full test split (990 clips)			
ER ($L = 5$)	1.52 (1.48, 1.55)	4.7 (4.6, 4.9)	40.6 (39.4, 41.8)
Generalist Tracker	<u>1.21</u> (1.17, 1.25)	<u>4.1</u> (3.8, 4.3)	<u>25.2</u> (24.0, 26.5)
BFMTrack (ours)	**0.81** (0.79, 0.83)	**2.5** (2.4, 2.6)	**20.6** (19.6, 21.7)
Hardest subset (30 clips)			
ER ($L = 5$)	3.11 (2.92, 3.32)	11.4 (9.8, 13.3)	84.8 (71.5, 99.3)
Generalist Tracker	3.14 (2.60, 3.78)	13.7 (10.1, 17.9)	79.8 (61.1, 100.1)
Specialist Tracker	**0.85** (0.68, 1.03)	**3.1** (2.4, 4.0)	**34.4** (24.8, 46.4)
BFMTrack (ours)	<u>2.07</u> (1.85, 2.28)	<u>6.8</u> (5.8, 8.0)	<u>63.4</u> (49.8, 78.2)

H.2 Robustness to External Disturbances

Table 8: BFMTrack achieves the best positional accuracy under external pushes. Values are mean ± sample standard deviation over 256 push schedules, each averaged across the 30 hardest clips. Parentheses give the change from undisturbed tracking. Best absolute scores within each block are bold; second-best scores are underlined.

Method	EMD ↓	MPJPE [cm] ↓	MMPJPE [cm] ↓
Strong: 100 N s pelvis pushes over 200 ms, every 1 s			
ER ($L = 5$)	<u>3.22</u> ± 0.04 (+0.13)	<u>12.4</u> ± 0.2 (+1.0)	<u>91.6</u> ± 2.0 (+6.6)
Generalist Tracker	5.44 ± 0.19 (+2.28)	29.0 ± 1.2 (+15.1)	123.5 ± 2.9 (+42.8)
Specialist Tracker	5.72 ± 0.10 (+4.87)	33.5 ± 0.6 (+30.4)	132.8 ± 2.5 (+98.3)
BFMTrack (ours)	**2.45** ± 0.02 (+0.38)	**9.2** ± 0.2 (+2.4)	**80.7** ± 2.3 (+17.6)
Small: 50 N random-link pushes, per-link probability $1/30$ *per frame*			
ER ($L = 5$)	<u>3.14</u> ± 0.04 (+0.05)	<u>11.8</u> ± 0.2 (+0.4)	<u>87.9</u> ± 1.9 (+2.9)
Generalist Tracker	4.00 ± 0.19 (+0.85)	19.3 ± 1.2 (+5.5)	99.0 ± 4.4 (+18.3)
Specialist Tracker	4.21 ± 0.23 (+3.36)	24.0 ± 1.4 (+20.9)	112.0 ± 5.7 (+77.5)
BFMTrack (ours)	**2.29** ± 0.02 (+0.22)	**8.2** ± 0.2 (+1.3)	**72.5** ± 1.9 (+9.3)

H.3 Ablations

Table 9: The PHC reward improves positional accuracy, and pink noise achieves the best accuracy among the tested noise colors. Mean results on the full test split for the goal-reaching reward (top) and r_{PHC} (bottom), varying the noise exponent β; small numbers give 95% bootstrap confidence intervals of the mean over clips. Best scores within each block are bold; second-best scores are underlined. Fig. 6 plots these values.

Method	EMD ↓	MPJPE [cm] ↓	MMPJPE [cm] ↓
BFM goal-reaching reward			
ER ($L = 5$)	1.52 (1.48, 1.55)	4.7 (4.6, 4.9)	40.6 (39.4, 41.8)
BFMTrack (ours) ($\beta = 0$)	1.29 (1.26, 1.32)	3.8 (3.7, 3.9)	35.3 (34.2, 36.5)
BFMTrack (ours) ($\beta = 1$)	**1.17** (1.14, 1.20)	**3.4** (3.3, 3.5)	**32.6** (31.5, 33.7)
BFMTrack (ours) ($\beta = 2$)	<u>1.18</u> (1.15, 1.21)	<u>3.5</u> (3.4, 3.6)	<u>33.6</u> (32.5, 34.6)
PHC reward r_{PHC}			
BFMTrack (ours) ($\beta = 0$)	0.99 (0.96, 1.02)	3.1 (3.0, 3.2)	25.9 (24.7, 27.2)
BFMTrack (ours) ($\beta = 1$)	**0.81** (0.79, 0.83)	**2.5** (2.4, 2.6)	**20.6** (19.6, 21.7)
BFMTrack (ours) ($\beta = 2$)	<u>0.84</u> (0.82, 0.87)	<u>2.6</u> (2.5, 2.7)	<u>22.0</u> (21.0, 23.1)

Table 10: BFMTrack outperforms the sampling-based update of DIAL-MPC at the same budget on the full test split. Small numbers give 95% bootstrap confidence intervals of the mean over clips. Best scores are bold; second-best scores are underlined. Fig. 7 plots these values.

Method	EMD ↓	MPJPE [cm] ↓	MMPJPE [cm] ↓
BFMTrack (ours)	**0.81** (0.79, 0.83)	**2.5** (2.4, 2.6)	**20.6** (19.6, 21.7)
DIAL-MPC	<u>1.09</u> (1.06, 1.11)	<u>3.4</u> (3.3, 3.4)	<u>27.6</u> (26.4, 28.8)

H.4 SPARSE REFERENCES

Table 11: Latent optimization improves positional tracking at every tested retained-frame rate. We compare BFMTrack with its zero-shot initialization over the 990 AMASS test clips at median retained-frame rates of 7.5 Hz, 3.0 Hz, 1.5 Hz and 1.0 Hz, using the BFM goal-reaching reward for 24 optimization iterations. EMD is computed against the full dense reference; MPJPE and MMPJPE are restricted to the retained frames; small numbers give 95% bootstrap confidence intervals of the mean over clips. Best scores within each block are bold; second-best scores are underlined.

Method	EMD ↓	MPJPE [cm] ↓	MMPJPE [cm] ↓
Retained-frame rate 7.5 Hz			
Zero-shot	<u>1.54</u> (1.51, 1.58)	<u>4.9</u> (4.7, 5.0)	<u>38.4</u> (37.3, 39.6)
BFMTrack (ours)	**1.29** (1.26, 1.32)	**3.6** (3.5, 3.7)	**30.1** (29.1, 31.1)
Retained-frame rate 3.0 Hz			
Zero-shot	<u>1.63</u> (1.59, 1.66)	<u>5.0</u> (4.9, 5.2)	<u>36.2</u> (35.1, 37.3)
BFMTrack (ours)	**1.45** (1.42, 1.48)	**3.9** (3.8, 4.0)	**29.1** (28.1, 30.0)
Retained-frame rate 1.5 Hz			
Zero-shot	<u>1.87</u> (1.83, 1.91)	<u>5.6</u> (5.4, 5.7)	<u>36.8</u> (35.7, 37.8)
BFMTrack (ours)	**1.76** (1.72, 1.80)	**4.5** (4.3, 4.6)	**31.1** (30.1, 32.1)
Retained-frame rate 1.0 Hz			
Zero-shot	<u>2.05</u> (2.00, 2.09)	<u>5.8</u> (5.7, 6.0)	<u>35.0</u> (33.9, 36.1)
BFMTrack (ours)	**1.96** (1.92, 2.00)	**4.7** (4.6, 4.9)	**30.1** (29.1, 31.1)