BEJEPA: A GOAL-CONDITIONED HUMANOID BEHAVIOR WORLD MODEL VIA UNSUPERVISED RL

Anonymous authors
Paper under double-blind review

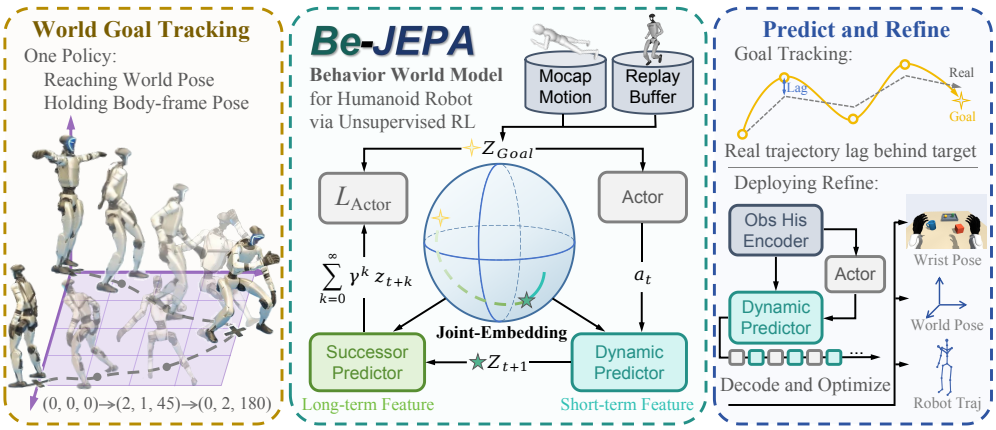

Figure 1: **BeJEPA: a goal-conditioned Behavior World Model for humanoid control.** Left: our goal is to jointly control world position, heading, and whole-body posture. Center: Dynamic and Successor learn physical consequences and long-term behavior in a shared latent space through unsupervised RL. Right: the Dynamic Predictor refines actions online.

ABSTRACT

Humanoid control requires coordinating whole-body posture with position and heading in the world frame. Mainstream controllers imitate trajectories in the root frame, but struggle to regain the target world pose after displacement. We introduce **BeJEPA**, a goal-conditioned Behavior World Model learned through unsupervised reinforcement learning from human motion priors and robot interaction. BeJEPA learns a shared representation of body configuration and world placement through a joint-embedding predictive architecture (JEPA). A Dynamic Predictor predicts the next state embedding under an action, while a Successor Predictor summarizes a goal-conditioned policy's expected discounted future state embeddings in a single vector. During joint training, Dynamic grounds the initial representation in observed physical transitions, while Successor increasingly organizes it around long-term behavior. Their composition guides policy improvement through one physical prediction and one behavioral summary, without long imagined rollouts. In 20-second motion-tracking experiments, BeJEPA with online action refinement using Dynamic outperforms all evaluated baselines in global body-position error, reducing it on AMASS by 61.7% relative to BFM-Zero with world-pose inputs and 58.7% relative to SONIC. Physical deployment on a Unitree G1 demonstrates joint attainment of posture, world position, and heading goals, recovery from the ground, and closed-loop return after disturbances and goal changes. Code: https://anonymous.4open.science/r/BeJEPA-08F8.

1 INTRODUCTION

Whole-body control (WBC) is the physical foundation of humanoid applications, enabling manipulation, expressive motion, and the execution of higher-level commands (Ben et al., 2025; Wei et al.,

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.

2026). It translates desired robot states into coordinated actuator commands. For world-coordinate tasks, the desired body posture must be reached together with the requested base position and heading. Existing motion-centered controllers can reproduce a posture while remaining displaced from its intended location. Correcting this displacement requires the robot to step, temporarily departing from the requested posture before reaching it at the right place. Whole-body control must therefore judge a movement by the goal it helps achieve, even when that movement initially worsens posture matching.

Human motion references show coordinated movement, making imitation a practical starting point for learning WBC (Peng et al., 2018; He et al., 2025; Liao et al., 2026). However, tracking a given trajectory does not explain how to reach a new goal when no trajectory is provided. This motivates reinforcement learning (RL), through which the robot can discover intermediate movements and evaluate how they help it reach the goal. Meta Motivo and BFM-Zero use unsupervised RL to learn reusable behaviors (Tirinzoni et al., 2025; Li et al., 2026). Given a reward that is linear in the learned state features, these methods can select behaviors to maximize return without specifying intermediate states. BFM-Zero uses body-relative features, so its learned objective focuses on posture and excludes absolute position and heading. To control posture and world placement together, the robot must learn how actions change its body and how the resulting behavior brings it closer to the goal.

We introduce **BeJEPA**, a *goal-conditioned Behavior World Model* for humanoid control, learned through unsupervised RL from human motion priors and robot interaction. Its shared representation encodes body configuration, world position, and heading. A goal-independent *Dynamic Predictor* models one-step action consequences, while a goal-conditioned *Successor Predictor* summarizes the policy's expected future state features. Early in joint training, Dynamic provides physical supervision that helps preserve useful features despite the unskilled policy's noisy long-term targets. As the policy improves, Successor organizes these features around long-term behavior. Their composition trains the actor to choose movements by their future goal alignment, including steps that temporarily worsen posture matching, without long imagined rollouts. Dynamic also supports online action refinement for time-indexed tracking.

Our contributions are threefold:

- **A goal-conditioned Behavior World Model for humanoid control.** BeJEPA learns a shared representation of posture and world placement through unsupervised RL, supporting goal reaching and motion tracking.
- **A Dynamic–Successor predictor decomposition.** We compose immediate dynamics with a goal-conditioned prediction of future state features, jointly learning the representation and guiding long-term policy improvement without long imagined rollouts.
- **World-anchored control in simulation and on hardware.** The learned actor improves global body alignment on AMASS and LAFAN. G1 demonstrates joint posture and world-pose control, ground recovery, and closed-loop return after disturbances and goal changes.

2 Related Work

Humanoid behavioral foundation models. Motion imitation is a dominant route to humanoid behavioral foundation models (Peng et al., 2018; He et al., 2025; Liao et al., 2026). SONIC and Humanoid-GPT scale tracking accuracy and generalization across motions while retaining a reference-sequence interface (Luo et al., 2026; Qi et al., 2026). Isolated or abruptly changed goals thus need an additional mechanism for intermediate motion. Hierarchical systems supply this motion with a separate planner (Tevet et al., 2025). Generated motions must remain within the tracker's capabilities, while replanning frequency trades motion consistency against responsiveness (Huang et al., 2025). Meta Motivo and BFM-Zero instead infer behaviors directly through unsupervised RL (Tirinzoni et al., 2025; Li et al., 2026). However, BFM-Zero's body-relative formulation leaves world position and heading outside its learned goals. BeJEPA learns world placement through both physical and behavioral prediction, outperforming BFM-Zero retrained with world-pose inputs (Section 4.2).

World models and latent-space control. Reconstruction-based models such as PlaNet and Dreamer allocate representation capacity to observation detail, including variation irrelevant to control (Hafner et al., 2019b;a; 2025). Dreamer and the decoder-free TD-MPC family limit rollout horizons and estimate the remaining return with learned values (Hafner et al., 2019a; Hansen et al., 2022; 2024).

These values depend on training rewards, so their long-term predictions do not automatically transfer to newly specified objectives. Joint-embedding predictive architectures (JEPAs) learn by predicting state embeddings, allowing abstraction from unpredictable observation detail (Assran et al., 2023; 2025). Yet planning with latent dynamics, as in DINO-WM and LeWorldModel, composes transitions and risks accumulating prediction error (Zhou et al., 2024; Maes et al., 2026). BeJEPA combines feature prediction with a directly learned, goal-conditioned summary of future state features, enabling long-term policy improvement without extended dynamics rollouts or downstream reward supervision.

Successor representations and reward-free reinforcement learning. Successor features summarize a policy's expected discounted future state features, separating behavioral prediction from reward readout (Barreto et al., 2017). Forward–Backward representations learn this structure to infer policies for new rewards and goals (Touati & Ollivier, 2021; Touati et al., 2022). TD-JEPA introduces joint-embedding learning of these long-term predictions from offline transitions (Bagatella et al., 2026). Acquiring informative transitions online remains a separate problem, motivating dedicated FB exploration strategies (Urpí et al., 2025). In online humanoid learning, the policy and its long-term prediction targets evolve together. Early movements may reveal little about how to reach a distant goal, although each executed action already provides an observed physical transition. BeJEPA learns these two timescales jointly: observed one-step transitions directly supervise the encoder while goal-conditioned continuation organizes the same representation over longer horizons.

3 BeJEPA: A Behavior World Model

BeJEPA learns a controller for goal reaching and motion tracking without task-specific retraining. It composes Dynamic's one-step physical prediction with Successor's goal-conditioned future-state feature summary in a shared representation of body configuration and world placement to guide policy improvement (Figure 2). Dynamic also refines actions online for spatial and temporal alignment.

3.1 Problem formulation

We formulate humanoid control as a partially observable Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, P, \Omega, p_0, \gamma)$. Here $s_t \in \mathcal{S}$ is the full state, $o_t \in \mathcal{O}$ is the controller's observation, and $a_t \in \mathcal{A} \subset \mathbb{R}^{29}$ specifies joint targets for a low-level proportional–derivative controller. Transitions follow $s_{t+1} \sim P(\cdot \mid s_t, a_t)$, observations follow $\Omega(\cdot \mid s_t)$, initial states follow p_0, and $\gamma \in (0, 1)$ discounts future outcomes. The observation contains proprioception, estimated root pose, and recent sensor and action history; the full state additionally exposes simulated body kinematics. Root pose is represented as $\rho_t = (x_t, y_t, \sin\psi_t, \cos\psi_t)$ relative to a fixed reference anchor shared by current and target states. A fixed target state s_g specifies goal reaching; a time-indexed reference s_t^{ref} specifies motion tracking. We jointly train the world model and a goal-conditioned, off-policy actor–critic from a replay buffer $\mathcal{B}$ of online robot transitions. A retargeted human motion dataset $\mathcal{D}_{\mathrm{ref}}$ supplies goals and a conditional motion prior (Peng et al., 2021); this prior and task-independent physical penalties guide executable control without downstream task rewards.

3.2 Goal-conditioned control with successor features

Goal reaching and motion tracking both require matching a target body configuration and its world placement. Let $E_\phi : \mathcal{S} \to \mathbb{R}^d$ encode a state as $z_t = E_\phi(s_t)$ and a target as $z_g = E_\phi(s_g)$. We normalize both embeddings to radius $\sqrt{d}$ and define the goal reward by their similarity:

$$r_g(s) = \langle E_\phi(s), z_g \rangle = d - \tfrac{1}{2}\|E_\phi(s) - z_g\|_2^2. \tag{1}$$

Thus maximizing reward is exactly minimizing distance to the target in the learned representation. The goal reward is linear in state features: z_g selects the target, and E_ϕ learns the similarity geometry. A fixed z_g specifies goal reaching; successive reference embeddings specify tracking goals.

Successor features turn instantaneous similarity into a future-behavior objective (Barreto et al., 2017), providing a basis for zero-shot reward transfer (Touati et al., 2022). For policy π_g conditioned on z_g and a state region $X \subseteq \mathcal{S}$, the normalized *successor measure* (Touati & Ollivier, 2021) is

$$M^{\pi_g}(X \mid s) = (1-\gamma)\,\mathbb{E}_{\pi_g}\left[\sum_{k=0}^{\infty} \gamma^k \mathbf{1}\{s_{t+k} \in X\} \,\middle|\, s_t = s\right]. \tag{2}$$

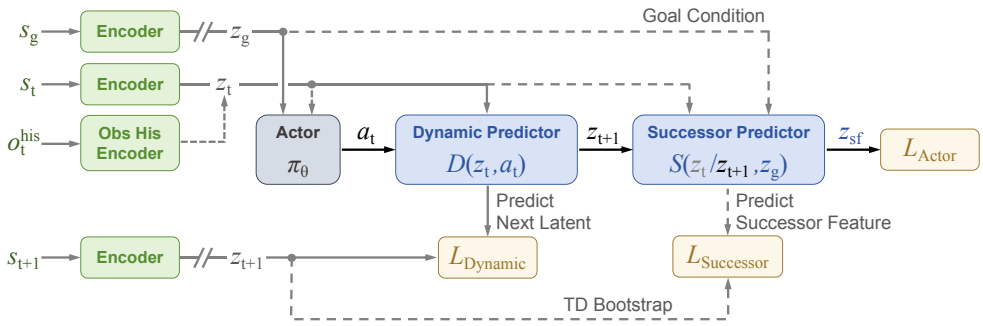

Figure 2: **Overview of the BeJEPA architecture.** Dynamic predicts the observed next latent, while Successor learns discounted continuation under the collection goal. Both objectives update the shared encoder. The actor learns through their composition, tracing its objective back to the current action. Slashed paths denote stop gradient; z_{sf} denotes a predicted successor feature.

This quantity counts discounted visits to X, including the current state; the factor $1-\gamma$ makes its total mass one. Generative temporal-difference models also learn discounted future-state distributions directly, avoiding recursive one-step prediction over the full horizon (Janner et al., 2020). Instead of representing a distribution over all future states, we retain its expected embedding:

$$S^{\pi_g}(s) := \int_{\mathcal{S}} E_\phi(s')\, M^{\pi_g}(\mathrm{d}s' \mid s) = (1-\gamma)\, \mathbb{E}_{\pi_g}\left[\sum_{k=0}^{\infty} \gamma^k z_{t+k} \,\middle|\, s_t = s\right]. \tag{3}$$

This *successor feature* summarizes the states the policy is expected to occupy. Because the goal reward is linear in this embedding, its normalized return follows directly:

$$V_g(s) := \int_{\mathcal{S}} r_g(s')\, M^{\pi_g}(\mathrm{d}s' \mid s) = \langle S^{\pi_g}(s), z_g \rangle\,. \tag{4}$$

The successor predicts the future states visited while pursuing the goal; its inner product with the goal evaluates their alignment. An action can therefore be useful even if the next posture is less similar to the target, if subsequent behavior brings the robot closer to it. Contrastive goal-conditioned RL shows a related connection between representation inner products and goal values (Eysenbach et al., 2022).

To learn this long-term prediction from individual transitions, separate the current feature:

$$S^{\pi_g}(s_t) = (1-\gamma) z_t + \gamma\, \mathbb{E}_{a_t \sim \pi_g,\, s_{t+1} \sim P}\left[S^{\pi_g}(s_{t+1})\right]. \tag{5}$$

This vector Bellman equation replaces an entire future trajectory with one observed transition and a prediction of its continuation. We use this identity to learn a goal-conditioned summary of future state features alongside one-step physical prediction.

3.3 Learning with Dynamics and Successor Prediction

BeJEPA separates learning what an action changes from learning what a goal-conditioned policy subsequently achieves (Figure 2). An immediate physical consequence can be learned from every attempted movement, even before the robot can reliably reach the intended goal. We therefore introduce a goal-independent *Dynamic Predictor* $D_\nu(z_t, a_t) \approx z_{t+1}$ and a goal-conditioned *Successor Predictor* $S_\xi(z_t, z_g) \approx S^{\pi_g}(s_t)$, both operating in the shared representation. Dynamic reuses physical transitions across goals; Successor shares a state-based continuation across actions with the same latent consequence, reducing the action dependence that must be learned through bootstrapping.

These two predictions connect through the Bellman equation. Let $Q_g(s_t, a_t)$ be the normalized return obtained by taking a_t now and following π_g thereafter. Substituting Equation 4 gives

$$\begin{aligned} Q_g(s_t, a_t) &= (1-\gamma) r_g(s_t) + \gamma\, \mathbb{E}_{s_{t+1} \sim P(\cdot \mid s_t, a_t)} V_g(s_{t+1}) \\ &= \left\langle (1-\gamma) z_t + \gamma\, \mathbb{E}_{s_{t+1} \sim P(\cdot \mid s_t, a_t)} S^{\pi_g}(s_{t+1}),\, z_g \right\rangle. \end{aligned} \tag{6}$$

This identity separates the present action from subsequent policy decisions; using a single predicted next latent requires a suitable representation. Following the JEPA principle of I-JEPA and V-JEPA 2 (Assran et al., 2023), prediction in a learned embedding space encourages retaining predictable structure and abstracting away unpredictable detail. Here, action conditioning makes a control input's repeatable consequences the prediction target. The aim is a next-state representation concentrated around a predictable outcome, so that the continuation can be evaluated at $D_\nu(z_t, a_t)$. This motivates the approximation $\mathbb{E}[S_\xi(z_{t+1}, z_g) \mid s_t, a_t] \approx S_\xi(D_\nu(z_t, a_t), z_g)$ and the composed action value:

$$\widehat{Q}_g(z_t, a_t) = \big\langle (1-\gamma) z_t + \gamma S_\xi(\underbrace{D_\nu(z_t, a_t)}_{\text{action consequence}}, z_g),\, z_g \big\rangle. \tag{7}$$

The actor can now evaluate an action through one physical prediction followed by one prediction of the remaining behavior. The first model is supervised by executed transitions; the second evaluates their longer-term consequences without recursively rolling out that horizon.

Both predictors learn from the robot's own replay transitions. Let z_b be the goal stored at collection, $d_t \in \{0, 1\}$ indicate termination, and $\gamma_t = \gamma(1 - d_t)$. With sg denoting stop-gradient and $S_{\bar{\xi}}$ the target network whose parameters $\bar{\xi}$ track ξ by an exponential moving average, the losses are

$$\mathcal{L}_D = \mathbb{E}_{\mathcal{B}}\Big[\|D_\nu(z_t, a_t) - \mathrm{sg}[z_{t+1}]\|_2^2\Big], \qquad \mathcal{L}_S = \mathbb{E}_{\mathcal{B}}\Big[\big\|S_\xi(z_t, z_b) - y_t^S\big\|_2^2\Big], \tag{8}$$

$$y_t^S = \mathrm{sg}\big[(1-\gamma_t) z_t + \gamma_t S_{\bar{\xi}}(z_{t+1}, z_b)\big]. \tag{9}$$

The Dynamic loss predicts a detached next-state embedding and updates E_ϕ through z_t, giving action-predictable information a direct representation-learning signal. Successor organizes the same coordinates around the future states visited under the stored goal z_b. Its TD target uses the *observed* next state, so training the continuation requires neither a predicted transition nor a next-action query to the actor. We combine these losses with representation regularization:

$$\mathcal{L}_{\text{world}} = \lambda_D \mathcal{L}_D + \mathcal{L}_S + \mathcal{L}_{\text{repr}}, \qquad \mathcal{L}_{\text{repr}} = \lambda_{\text{orth}} \mathcal{L}_{\text{orth}} + \lambda_w \mathbb{E}_{\mathcal{B}}\big[\|H_\zeta^w(z_t) - w(s_t)\|_{W_w}^2\big]. \tag{10}$$

Here $\mathcal{L}_{\text{orth}}$ discourages redundant features, while a lightweight wrist readout H_ζ^w preserves fine control detail by reconstructing wrist pose and motion $w(s_t)$; $\|v\|_W^2 = v^\top W v$ denotes a weighted squared error. The encoder and predictors learn jointly, with λ_D decreasing over training to emphasize local physical prediction early and behavioral continuation as the repertoire develops.

An observation-history encoder distills the shared representation into $\widetilde{z}_t = E_\eta^{\text{obs}}(o_t)$, where o_t includes recent observation history, allowing the actor to operate on control observations and action history h_t. This transfer from privileged training information to deployment observations is also used in teacher–student learning for legged locomotion (Miki et al., 2022). For goals drawn from p_g, which mixes collection goals with replay and motion-reference states, we optimize the composed value in Equation 7. Omitting its action-independent current-state term, the actor loss is

$$\mathcal{L}_{\text{actor}} = -\gamma\, \mathbb{E}_{\mathcal{B}, p_g}[\langle S_\xi(D_\nu(z_t, \pi_\theta(\widetilde{z}_t, h_t, z_g)), z_g), z_g \rangle] + \mathcal{R}_{\text{act}}, \tag{11}$$

where $\mathcal{R}_{\text{act}}$ combines the motion prior and physical execution penalties. Only the actor is updated: gradients pass through the fixed Successor and Dynamic predictors, translating a desirable future state change into a change of the current action. Algorithm 1 summarizes training; observation distillation, ensemble aggregation, and regularization details appear in Appendix C.

3.4 Aligning Behavior in Space and Time

Motion tracking requires the right body configuration *at the right place and time*. Successive goal embeddings guide the actor, but its objective evaluates discounted future similarity to each goal, leaving room for spatial offset and temporal lag. Finite-dimensional task features can also blur spatial precision, as analyzed in FB-AWARE (Cetin et al., 2024). We use Dynamic to compare a short predicted trajectory with time-indexed references and refine the actor's actions online, using gradient-based latent MPC similar to Dream-MPC (Spieler & Behnke, 2026).

The actor and planner start from the same observation-history embedding $\widehat{z}_0 = \widetilde{z}_t = E_\eta^{\text{obs}}(o_t)$, without privileged current-state inputs. Reference embeddings $z_{t+1:t+H}^{\text{ref}}$ come from the commanded motion. For bounded action residuals δ_k, the rollout is

$$\bar{a}_k = \pi_\theta(\widehat{z}_k, h_k, z_{t+k+1}^{\text{ref}}), \quad a_k = \mathrm{clip}_{\mathcal{A}}(\bar{a}_k + \delta_k), \quad \widehat{z}_{k+1} = \mathcal{D}_{t,k}(\widehat{z}_k, a_k), \tag{12}$$

Algorithm 1 BeJEPA training

Require: Motion data $\mathcal{D}_{\text{ref}}$, replay $\mathcal{B}$, batch size n, goal sampler p_g.
Initialize $E_\phi, D_\nu, S_\xi, H_\zeta^w, E_\eta^{\text{obs}}, \pi_\theta$ and target network $S_{\bar{\xi}}$ with $\bar{\xi} = \xi$.
while interaction budget remains **do**
  Collect goal-conditioned transitions into $\mathcal{B}$.
  for each replay update **do**
    Sample $\{(s_i, o_i, h_i, a_i, s_i^+, z_{b,i}, d_i)\}_{i=1}^n \sim \mathcal{B}$ and encode:
    $z_i = E_\phi(s_i)$; $z_i^+ = E_\phi(s_i^+)$; $\widetilde{z}_i = E_\eta^{\text{obs}}(o_i)$; $\gamma_i = \gamma(1 - d_i)$.
    $\mathcal{L}_D = \frac{1}{n}\sum_i \|D_\nu(z_i, a_i) - \text{sg}[z_i^+]\|_2^2$.
    $\mathcal{L}_S = \frac{1}{n}\sum_i \left\|S_\xi(z_i, z_{b,i}) - \text{sg}[(1-\gamma_i)z_i + \gamma_i S_{\bar{\xi}}(z_i^+, z_{b,i})]\right\|_2^2$.
    $\mathcal{L}_{\text{world}} = \lambda_D \mathcal{L}_D + \mathcal{L}_S + \mathcal{L}_{\text{repr}}$, with $\mathcal{L}_{\text{repr}}$ from Equation 10.
    $\mathcal{L}_{\text{obs}} = \frac{1}{n}\sum_i \|\widetilde{z}_i - \text{sg}[z_i]\|_2^2$.
    $\{z_{g,i}\}_{i=1}^n \sim p_g$; $\widehat{a}_i = \pi_\theta(\widetilde{z}_i, h_i, z_{g,i})$.
    $\mathcal{L}_{\text{actor}} = -\frac{\gamma}{n}\sum_i \langle S_\xi(D_\nu(z_i, \widehat{a}_i), z_{g,i}), z_{g,i}\rangle + \mathcal{R}_{\text{act}}$.
    Minimize $\mathcal{L}_{\text{world}}$ over $(E_\phi, D_\nu, S_\xi, H_\zeta^w)$ and $\mathcal{L}_{\text{obs}}$ over E_η^{obs}.
    Update only π_θ to minimize $\mathcal{L}_{\text{actor}}$, keeping encoders and predictors fixed.
    Update target parameters $\bar{\xi} \leftarrow (1-\tau)\bar{\xi} + \tau\xi$.
  end for
end while

where h_k advances the action history and $\mathcal{D}_{t,k}$ uses the frozen Dynamic ensemble, optionally corrected by recent observation–prediction discrepancies (Appendix B). Only the first $R \leq H$ residuals are optimized; the remaining steps follow the actor.

A root-pose readout trained on detached latents and wrist readout expose geometric errors alongside latent similarity. Writing $e_{\text{geom},k}(z)$ for root-position, wrapped-heading, and wrist-pose reference errors at step k, define $c_k(z) = \lambda_z \|z - z_{t+k+1}^{\text{ref}}\|_2^2/d + \|e_{\text{geom},k}(z)\|_W^2$. All execution priorities use:

$$\mathcal{J}_t(\delta) = \sum_{k=0}^{H-1} \gamma^k \left[c_k(\widehat{z}_{k+1}) + \lambda_a \|a_k - \bar{a}_k\|_2^2 + \lambda_\Delta \|a_k - a_{k-1}\|_2^2\right], \qquad \|\delta_k\|_\infty \leq \epsilon, \tag{13}$$

where a_{-1} is the last executed action. Cost weights, planning horizons, and adjustable joints specify desired precision. A bounded gradient search compares corrections with the actor rollout, executes the first action of the lowest-cost sequence, and replans from the next observation. Reference-dependent activation can retain the actor outside the refinement regime. All network weights remain fixed: the physical model learned for behavior acquisition also supports online spatial and temporal alignment.

4 Experiments

Our experiments examine whether BeJEPA can coordinate body posture with world placement, and how physical and behavioral prediction support this ability. We ask four questions:

Q1. How well does BeJEPA perform on standard tracking metrics?

Q2. Does the Dynamic–Successor split improve world-goal control?

Q3. How are position, motion, and physical state organized in the latent space?

Q4. How does BeJEPA perform on the physical robot?

4.1 Q1: How well does BeJEPA track whole-body motion?

We pretrain with 4,096 parallel Isaac Sim environments, following massively parallel legged RL (Rudin et al., 2022). We evaluate the 29-actuator G1 policy in MuJoCo at 50 Hz on selected non-interactive AMASS segments (Mahmood et al., 2019) and LAFAN segments (Harvey et al., 2020), each lasting 20 s to assess long-distance tracking. Unlike prior protocols that exclude trajectories with large spatial deviations and mark them as failures, we retain these trajectories to fairly evaluate the error in world coordinates. We measure pelvis-centered and global MPJPE, root translation and heading errors, and local wrist position and orientation errors. We compare with SONIC, Humanoid-GPT, and BFM-Zero retrained with world-pose inputs.

Table 1: **Whole-body tracking.** MPJPE is in mm. Root err. is $e_{xy} + 0.01e_{\text{yaw}}$ (m, degrees); wrist err. is $e_{\text{pos},L} + 0.5e_{\text{rot},L}$ (cm, degrees). Bold and underline indicate the best and second-best results.

	AMASS				LAFAN			
Method	$\text{MPJPE}_G\downarrow$	$\text{MPJPE}_L\downarrow$	Root err.↓	Wrist err.↓	$\text{MPJPE}_G\downarrow$	$\text{MPJPE}_L\downarrow$	Root err.↓	Wrist err.↓
BFM-Zero (world)	573.4	97.1	0.801	32.4	797.1	89.3	1.020	25.8
SONIC	531.5	**49.7**	0.571	**15.3**	887.5	<u>77.8</u>	0.971	<u>23.6</u>
Humanoid-GPT	832.4	115.7	1.107	27.5	1433.1	127.9	1.720	29.7
BeJEPA	<u>278.0</u>	82.3	<u>0.377</u>	25.9	<u>584.5</u>	81.9	<u>0.716</u>	24.3
BeJEPA-MPC (Balanced)	**219.5**	<u>63.9</u>	**0.255**	<u>16.3</u>	**543.8**	**75.3**	**0.639**	**20.1**

World placement is part of the behavior goal. Local posture accuracy can remain high despite drift from the reference path, making global alignment essential to world-directed control. BeJEPA has the lowest global MPJPE and root error among the learned policy baselines on both datasets (Table 1). The shared goal specifies desired body pose and its world placement. Compared with SONIC, BeJEPA improves world placement despite higher MPJPE_L. Correcting world position can require an extra step that sacrifices local pose matching, making MPJPE_G better suited to world-coordinate evaluation, especially as targets derive from human trajectories infeasible for robots.

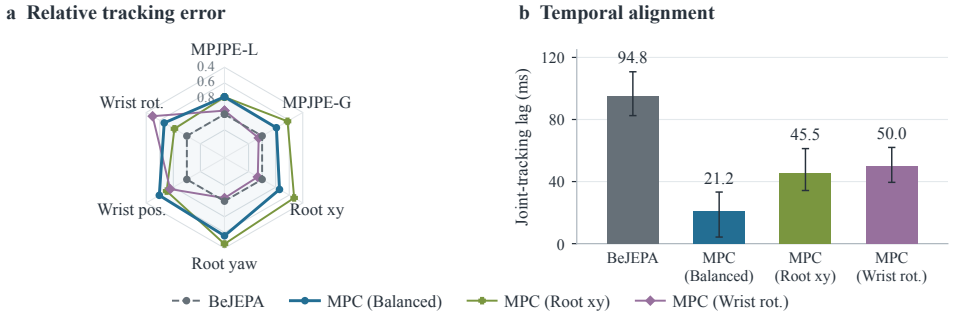

Figure 3: **Tracking trade-offs across execution priorities and temporal alignment on AMASS.** (a) Six errors normalized by BeJEPA (1.0); outward is better on a common linear scale. (b) Mean joint-tracking lag over ± 500 ms shifts (20 ms steps).

Physical prediction aligns execution with the reference clock. BeJEPA's successor objective values continued proximity to a goal; tracking additionally requires arriving at each reference state at its assigned time. BeJEPA-MPC (Balanced) reduces error across all six metrics, with mean relative reductions of 31.2% on AMASS and 13.4% on LAFAN using the same pretrained weights. We examine temporal alignment by shifting each trajectory's reference to minimize joint-angle error. BeJEPA-MPC (Balanced) reduces the mean signed lag from 94.8 to 21.2 ms on AMASS (Figure 3b) and from 97.2 to 51.0 ms on LAFAN. This connects Dynamic's two roles: transition supervision learns control-relevant consequences, and online prediction uses them to align behavior with time-indexed targets. This addresses the temporal precision required beyond future goal similarity (Section 3.4).

Execution priorities can be specified at inference. At deployment, Dynamic supports task-specific precision through the MPC cost and rollout settings. BeJEPA-MPC (Root xy) and BeJEPA-MPC (Wrist rot.) emphasize root translation and wrist orientation, respectively, improving the corresponding metric with trade-offs elsewhere (Figure 3a). The same frozen model can thus adapt execution to a task's priorities without retraining (Appendix B).

4.2 Q2: Does the Dynamic–Successor split improve world control?

We compare BeJEPA with world-input BFM-Zero, a merged predictor of comparable capacity, split variants with either predictor's encoder gradient stopped, and Dynamic-only MPC. The gradient

ablations retain both predictors and the actor objective, isolating each predictor's contribution to the representation. Under fixed goals, minimum position and heading errors measure closest approach, while mean errors measure alignment throughout execution.

Table 2: **Dynamic–Successor ablation on goal inference.** Lower errors are better.

	Root xy (m)		Root yaw (°)	
Model	Min.	Mean	Min.	Mean
BFM-Zero (world)	0.450	0.984	8.28	46.26
Merged predictor	0.666	1.348	6.41	35.81
Split, no Dynamic encoder gradient	0.645	1.231	4.76	34.10
Split, no Successor encoder gradient	0.221	0.885	3.19	25.41
Dynamic-only MPC	0.675	1.004	11.30	82.23
BeJEPA	**0.159**	**0.642**	**2.47**	**24.02**

Both predictors contribute to world control. The merged successor predictor has large world-goal errors (Table 2), consistent with the difficulty of learning from noisy long-term targets. Splitting improves alignment even with Dynamic's encoder gradient stopped, supporting the structural benefit in unsupervised RL, although errors remain large. Training the encoder through Dynamic while stopping Successor's gradient improves alignment further, but joint training performs best. Dynamic grounds the features; Successor organizes them for long-term control. MPC following LeWorldModel (Maes et al., 2026) without the actor also struggles with distant goals: motion exceeds Dynamic's reliable local horizon (Appendix E.2), motivating Successor's long-term summary without extended rollouts.

4.3 Q3: What structure does BeJEPA give the latent space?

Body content and world placement jointly define a goal. We randomly sample body states from the buffer and cross them with randomized x, y, and yaw values. To expose how world placement changes a goal, we subtract each body's mean embedding across world poses and fit a shared principal component analysis (PCA) to these changes. The resulting coordinates vary systematically with translation and heading (Figure 4). World x varies mainly along PC_1^ρ and world y along PC_2^ρ, while PC_3^ρ partially separates heading, with substantial overlap. The representation retains body content while distinguishing its possible world placements, supplying the latent goal reward's geometry.

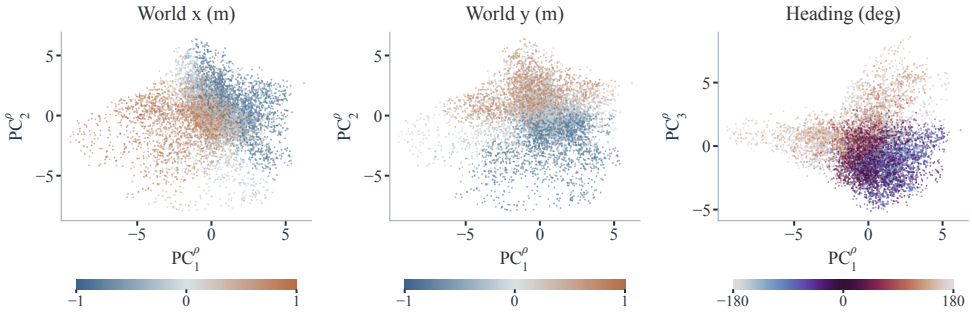

Figure 4: **World geometry within a shared body-state representation.** Eighty body states are crossed with random world poses: $x, y \in [-1, 1]$ m and yaw in $[-\pi, \pi]$. Within-body PCA (PC^ρ) subtracts body means before projection; colors show x, y, and heading.

A common world pose reveals the whole-body repertoire. We examine trajectories in the motion prior, placing their body states at a common root position and heading. The t-SNE map organizes crawling, kneeling, upright gaits, and expressive movement into characteristic posture neighborhoods (Figure 5a). Similar body poses tend to occupy nearby regions in this projection, suggesting that the latent space captures meaningful posture similarity across motions.

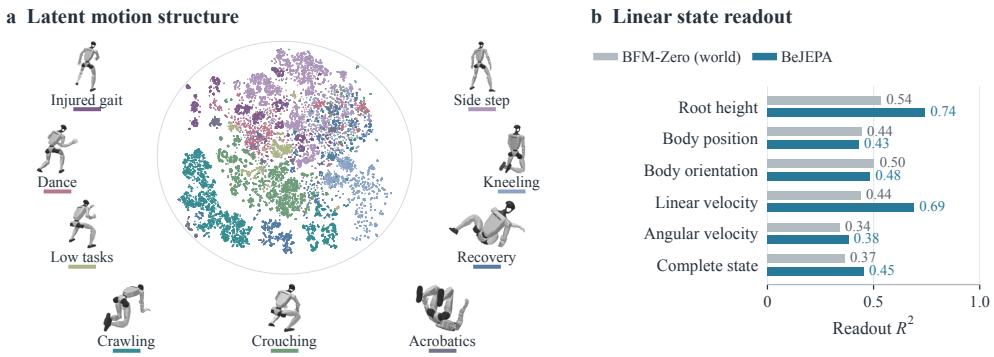

Figure 5: **Motion organization and linear state readouts in the latent space.** (a) t-SNE with root $x = y = \psi = 0$; colors match the illustrated motion families. (b) Linear state readouts at the original world poses.

Physical quantities are more linearly accessible. Linear readouts measure accessible physical information (Figure 5b). Relative to world-input BFM-Zero, BeJEPA improves linear readout of body velocity and root height, while position and orientation remain comparable. Although lower scores do not rule out nonlinear recovery, these gains, together with the posture neighborhoods and encoder-gradient ablation, suggest that BeJEPA learns a more structured latent representation, with physical information more readily accessible to linear readouts.

4.4 Q4: How does BeJEPA perform on the physical robot?

We deploy the pretrained BeJEPA policy on a physical Unitree G1. The onboard LiDAR, coupled with FAST-LIO (Xu & Zhang, 2021; Xu et al., 2022), provides the position and heading estimates used by the world-anchored observation interface. Together with proprioception, these estimates allow the policy to relate the robot's current body configuration to a goal placed in the world.

Recovering balance and posture. A world-anchored goal remains meaningful when an external disturbance moves the robot away from it. Following a kick, the G1 first steps toward its original standing location, then makes a smaller second step to complete the return (Figure 6a). Starting from the ground presents a larger change in body configuration: the robot pushes up, rises, reorients, and reaches the commanded standing pose (Figure 6b). The two sequences demonstrate complementary aspects of recovery: correcting displacement and restoring an upright body configuration.

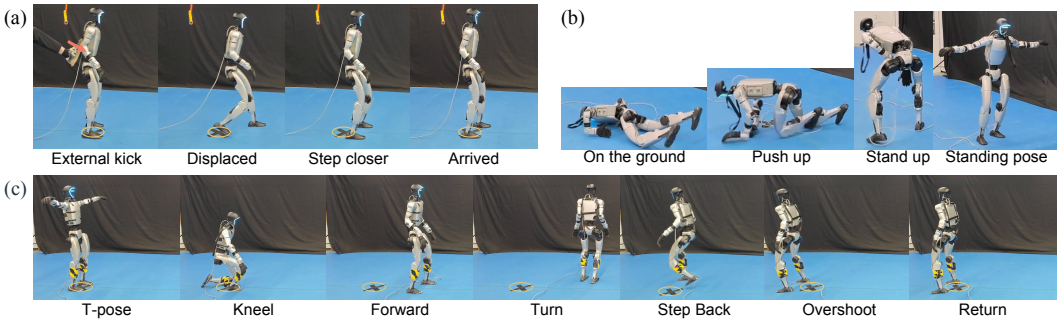

Figure 6: **World-anchored control on the physical G1.** (a) Recovery after a kick. (b) Standing up from the ground. (c) Sequential goal switching. Gold rings identify the floor reference marks in (a,c).

Changing behavior preserves the spatial reference. Sequential goals take the robot from a T-pose through kneeling, advancing, and turning before returning to its starting location (Figure 6c). Returning to the same floor mark despite these posture and heading changes demonstrates that the learned behaviors retain a persistent spatial reference.

AI USE STATEMENT

Generative AI tools assisted with manuscript drafting, polishing, code analysis, and preparation of analysis and plotting scripts. We have reviewed all AI-assisted work, including code, text, data, and claims. We take responsibility for the final content of this work.

REFERENCES

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15619–15629. IEEE, 2023.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. Td-jepa: Latent-predictive representations for zero-shot reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 36211–36244, 2026.

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

Qingwei Ben, Feiyu Jia, Jia Zeng, Junting Dong, Dahua Lin, and Jiangmiao Pang. Homie: Humanoid loco-manipulation with isomorphic exoskeleton cockpit. *arXiv preprint arXiv:2502.13013*, 2025.

Edoardo Cetin, Ahmed Touati, and Yann Ollivier. Finer behavioral foundation models via auto-regressive features and advantage weighting. *arXiv preprint arXiv:2412.04368*, 2024.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 35603–35620, 2022.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Nick Hansen, Hao Su, and Xiaolong Wang. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, volume 2024, pp. 47376–47405, 2024.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022.

Félix G Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust motion in-betweening. *ACM Transactions on Graphics (TOG)*, 39(4):60–1, 2020.

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

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):1–12, 2025.

Michael Janner, Igor Mordatch, and Sergey Levine. Generative temporal difference learning for infinite-horizon prediction. *arXiv preprint arXiv:2010.14496*, 2020.

Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, et al. Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 79697–79725, 2026.

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.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. Sonic: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117):eaed4592, 2026.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5442–5451, 2019.

Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science robotics*, 7(62): eabk2822, 2022.

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, 2018.

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

Zekun Qi, Xuchuan Chen, Dairu Liu, Chenghuai Lin, Yunrui Lian, Sikai Liang, Zhikai Zhang, Yu Guan, Jilong Wang, Wenyao Zhang, et al. Humanoid-gpt: Scaling data and structure for zero-shot motion tracking. *arXiv preprint arXiv:2606.03985*, 2026.

Nikita Rudin, David Hoeller, Philipp Reist, and Marco Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on robot learning*, pp. 91–100. PMLR, 2022.

Jonathan Spieler and Sven Behnke. Dream-mpc: Gradient-based model predictive control with latent imagination. *arXiv preprint arXiv:2605.04568*, 2026.

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit Bermano, and Michiel Van de Panne. Closd: Closing the loop between simulation and diffusion for multi-task character control. In *International Conference on Learning Representations*, volume 2025, pp. 46506–46520, 2025.

Andrea Tirinzoni, Ahmed Touati, Jesse Farebrother, Mateusz Guzek, Anssi Kanervisto, Yingchen Xu, Alessandro Lazaric, and Matteo Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models. In *International Conference on Learning Representations*, volume 2025, pp. 21693–21748, 2025.

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

Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? *arXiv preprint arXiv:2209.14935*, 2022.

Núria Armengol Urpí, Marin Vlastelica, Georg Martius, and Stelian Coros. Epistemically-guided forward-backward exploration. *arXiv preprint arXiv:2507.05477*, 2025.

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Jie Liu, Xiawei Liu, Kaidi Kang, et al. Ψ_0: An open foundation model towards universal humanoid loco-manipulation. *arXiv preprint arXiv:2603.12263*, 2026.

Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.

Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

APPENDIX

A SUCCESSOR MEASURES AND TEMPORAL-DIFFERENCE LATENT PREDICTION

This section develops the successor-feature interpretation used in Section 3.2 and connects it to TD-JEPA (Bagatella et al., 2026). Throughout, π_g denotes a fixed goal-conditioned policy, E_ϕ a fixed feature map when deriving prediction identities, and s the Markov state including the controller history. We use the normalized, current-state convention of Equation 2.

A.1 A DISTRIBUTION OVER FUTURE STATES

Let K be independent of the trajectory with $\Pr(K = k) = (1-\gamma)\gamma^k$, $k \geq 0$. The successor measure has the probabilistic interpretation

$$M^{\pi_g}(X \mid s) = \Pr(s_{t+K} \in X \mid s_t = s, \pi_g). \tag{14}$$

It is therefore a probability measure over states visited at a geometrically distributed future time. Writing $P_g(\mathrm{d}s^+ \mid s) = \int_{\mathcal{A}} P(\mathrm{d}s^+ \mid s, a)\pi_g(\mathrm{d}a \mid s)$, separating the event $K = 0$ gives

$$M^{\pi_g}(\mathrm{d}s' \mid s) = (1-\gamma)\delta_s(\mathrm{d}s') + \gamma \int_{\mathcal{S}} M^{\pi_g}(\mathrm{d}s' \mid s^+) P_g(\mathrm{d}s^+ \mid s). \tag{15}$$

Integrating $E_\phi(s')$ yields the vector Bellman equation

$$S^{\pi_g}(s) = (1-\gamma)E_\phi(s) + \gamma \mathbb{E}_{s^+ \sim P_g(\cdot|s)} S^{\pi_g}(s^+). \tag{16}$$

For a finite state space, let Φ contain the state features as rows and S_g their successor features. Then

$$\mathsf{S}_g = (1-\gamma)(I - \gamma P_g)^{-1}\Phi = (1-\gamma)\sum_{k \geq 0} \gamma^k P_g^k \Phi. \tag{17}$$

The predictor thus compresses repeated application of the policy's transition operator into a single feature vector.

A.2 FROM FUTURE-FEATURE REGRESSION TO TD-JEPA

Consider predicting $Y = E_\phi(s_{t+K})$ from the current state and goal. Its conditional mean is exactly $S^{\pi_g}(s)$. For any prediction $f(s, z_g)$, expanding the square around this mean gives

$$\begin{aligned}\mathbb{E}\left[\|f(s,z_g) - Y\|_2^2 \mid s, g\right] &= \|f(s,z_g) - S^{\pi_g}(s)\|_2^2 \\ &\quad + \mathbb{E}\left[\|Y - S^{\pi_g}(s)\|_2^2 \mid s, g\right].\end{aligned} \tag{18}$$

The cross term vanishes because $\mathbb{E}[Y - S^{\pi_g}(s) \mid s, g] = 0$. The second term is independent of f: regression onto sampled future embeddings learns their discounted barycenter. This is the latent-prediction interpretation of successor features in TD-JEPA.

Temporal-difference learning replaces the sampled future by the one-step decomposition in Equation 16. Define $\mathcal{T}_g F = (1-\gamma)E_\phi + \gamma P_g F$. For bounded vector-valued functions,

$$\|\mathcal{T}_g F - \mathcal{T}_g G\|_\infty \leq \gamma \|F - G\|_\infty. \tag{19}$$

Hence S^{π_g} is its unique fixed point, and repeated Bellman updates recover the same expected feature as future-state regression. In BeJEPA, the sampled regression target is

$$y_t^S = \mathrm{sg}\left[(1-\gamma_t)E_\phi(s_t) + \gamma_t S_{\bar{\xi}}(E_\phi(s_{t+1}), z_b)\right], \qquad \gamma_t = \gamma(1-d_t), \tag{20}$$

where z_b is the collection goal and d_t indicates termination. Both features are computed by the online encoder, and the target predictor follows $\bar{\xi} \leftarrow (1-\tau)\bar{\xi} + \tau\xi$. The stored goal associates the observed transition with its behavioral continuation; goal relabeling for policy improvement is described in Appendix C.1. At termination, the target reduces to the current feature.

The action-conditioned predictor in TD-JEPA uses future features starting at the next state. In normalized notation, write this quantity as $F_g^+(s,a) = (1-\gamma)\mathbb{E}[\sum_{k \geq 0} \gamma^k E_\phi(s_{t+k+1}) \mid s_t = s, a_t = a, \pi_g]$. The two conventions are related by

$$F_g^+(s,a) = \mathbb{E}_{s^+|s,a} S^{\pi_g}(s^+), \qquad S^{\pi_g}(s) = (1-\gamma)E_\phi(s) + \gamma \mathbb{E}_{a \sim \pi_g} F_g^+(s,a). \tag{21}$$

BeJEPA represents the first expression through a physical transition followed by a goal-conditioned continuation, $S_\xi(D_\nu(z,a), z_g)$.

A.3 WHY THE PREDICTION SUPPORTS CONTROL

For a linear feature reward $r_w(s) = E_\phi(s)^\top w$, integration against the successor measure gives

$$(1-\gamma)\mathbb{E}_{\pi_g}\left[\sum_{k\geq 0}\gamma^k r_w(s_{t+k}) \mid s_t = s\right] = S^{\pi_g}(s)^\top w. \tag{22}$$

Here g determines the behavior and w determines the reward readout. BeJEPA chooses $w = z_g = E_\phi(s_g)$. Because $\|E_\phi(s)\|_2^2 = \|z_g\|_2^2 = d$, the resulting goal value also satisfies

$$V_g(s) = d - \frac{1-\gamma}{2}\mathbb{E}_{\pi_g}\left[\sum_{k\geq 0}\gamma^k \|E_\phi(s_{t+k}) - z_g\|_2^2 \mid s_t = s\right]. \tag{23}$$

Policy improvement therefore minimizes distance to the goal across the discounted future. To connect this objective to the present action, the composition in Equation 7 gives

$$\nabla_\theta \widehat{Q}_g = \gamma \left(\frac{\partial \pi_\theta}{\partial \theta}\right)^\top \left(\frac{\partial D_\nu}{\partial a}\right)^\top \left(\frac{\partial S_\xi}{\partial z^+}\right)^\top z_g, \qquad z^+ = D_\nu(z_t, a_t). \tag{24}$$

The current-state reward has zero action derivative. Successor supplies a direction of improvement in the next-state representation, and Dynamic translates that direction into an action update. The encoders and predictors are held fixed while this gradient updates the actor.

A.4 WHY SEPARATING DYNAMIC AND SUCCESSOR IMPROVES LEARNING

We show how separating action consequences from behavioral continuation improves replay reuse while preserving the goal-conditioned control objective.

Following JEPA's emphasis on predictable features (Assran et al., 2023; 2025), assume that $z = E_\phi(s)$ removes unpredictable nuisance variation while retaining all information needed for goal rewards and policy decisions. The resulting latent is Markov, and its next state is determined by the current latent and action: $z^+ = D^\star(z, a)$. Fix the encoder and a goal-conditioned policy π_g for the following argument.

Proof. Let $S_g(z)$ denote the policy's successor feature in this representation. By Equation 21, the action-conditioned future feature satisfies

$$F_g^+(z, a) = \mathbb{E}[S_g(z^+) \mid z, a] = S_g(D^\star(z, a)). \tag{25}$$

Substituting this identity into Equation 6 recovers the composed value in Equation 7 with exact predictors. The decomposition therefore preserves action values and their maximizing actions.

The same identity exposes the sharing that makes learning more efficient. A merged predictor learns continuation as a function of (z, a, z_g). The split assigns immediate action dependence to Dynamic, whose observed next-latent targets can be learned from transitions collected under any goal. Each improvement in this physical prediction is consequently reused across goals. Successor learns continuation as a function of (z, z_g): all predecessor state–action pairs reaching the same latent share its future-feature prediction. An update to that continuation therefore benefits every action leading to it, reducing duplicated long-term prediction across actions. Moreover, Equation 9 learns this continuation directly from observed state transitions, without querying a next action in the bootstrap. Thus action effects receive direct one-step supervision, while TD learning estimates a shared state-based continuation; the long-term predictor no longer needs to separately learn the initiating action's effect on future value. □

Appendix D supports this efficiency gain empirically: even with Dynamic's encoder gradient stopped, the split model achieves lower early global-tracking and fixed-goal heading errors than the merged predictor. Full BeJEPA improves further when Dynamic also shapes the encoder, linking the predictable-representation premise to the observed learning advantage.

B Model-Predictive Action Refinement

B.1 A shared prediction and refinement objective

All MPC settings use the same frozen actor, History Encoder, Dynamic ensemble, and geometric readouts. The History Encoder initializes the current latent from deployable observations; the state encoder supplies reference-goal embeddings. The actor provides coordinated nominal behavior, while Dynamic predicts how small action changes alter its execution. We average the two predictors' costs and propagate their mean latent, reducing sensitivity to an individual predictor's errors. The stage cost in Equation 13 is

$$c_k(z) = \frac{\lambda_z}{d}\|z - z^{\mathrm{ref}}_{t+k+1}\|_2^2 + \lambda_{xy}\|e_{xy,k}\|_2^2 + \lambda_\psi e_{\psi,k}^2 + \frac{\lambda_{wp}}{6}\|e_{wp,k}\|_2^2 + \frac{\lambda_{wr}}{12}\|e_{wr,k}\|_2^2. \tag{26}$$

Root and wrist positions are in meters, heading errors are wrapped to $[-\pi, \pi]$, and each wrist orientation uses a six-dimensional rotation representation. The latent term preserves whole-body agreement, while geometric readouts make task-specific precision explicit. Scoring every predicted step with $\gamma = 0.98$ constrains the intermediate motion as well as its destination; residual and smoothness penalties retain continuity with the actor.

B.2 Execution priorities within the same controller

Table 3 gives the settings reported in Section 4.1. Balanced refinement combines latent, root, and wrist terms over a horizon long enough to anticipate coordinated motion. Root-xy refinement increases the translation weight and prediction horizon, allowing earlier actions to account for their later effect on world placement. Wrist-orientation refinement instead concentrates the cost on the hands and restricts corrections to the arm and wrist joints over a short horizon. This directs the action adjustment toward local precision while retaining the actor's lower-body behavior. These priorities change how the same physical predictions are evaluated, without learning a new policy for each task.

Table 3: MPC settings for the reported execution priorities. H: prediction horizon; R: residual horizon; N: gradient updates. Bounds ϵ and step sizes η are in normalized action coordinates. All settings use per-step costs with $\gamma = 0.98$.

Setting	H	R	N	ϵ	η	λ_a	λ_Δ
Balanced	16	8	8	0.20	0.080	0.10	0.030
Root xy	24	8	8	0.24	0.080	0.03	0.010
Wrist rotation	3	3	6	0.30	0.150	0.01	0.005

Setting	λ_z	λ_{xy}	λ_ψ	λ_{wp}	λ_{wr}	Refined joints
Balanced	20	80	12	300	30	All
Root xy	20	120	12	0	0	All
Wrist rotation	0.2	0.5	0.1	600	100	Arms and wrists

Balanced and root-xy refinement apply to reference segments with mean planar speed at most 0.8 m/s, retaining the actor on faster segments; wrist refinement applies throughout. This confines whole-body intervention to slower reference motions while allowing local hand corrections across the motion set. The activation rule uses the supplied reference, so a task's motion requirements determine when refinement is applied.

Filtering prediction bias for world placement. Root-xy refinement uses the supplied reference root poses directly and anchors predicted displacement to the currently observed root pose. This stabilizes world coordinates, a low-frequency component that is prone to overshoot. To reduce persistent transition bias, we also filter the discrepancy between observed and predicted latents:

$$\begin{aligned} b_t &= (1-\alpha)b_{t-1} + \alpha\big[\widetilde{z}_t - \overline{D}_\nu(\widetilde{z}_{t-1}, a_{t-1})\big], \\ \widehat{z}^{(i)}_{k+1} &= \Pi\Big(D^{(i)}_\nu(\widehat{z}_k, a_k) + \beta^k b_t\Big), \end{aligned} \tag{27}$$

where $\overline{D}_\nu$ is the ensemble mean, Π projects onto the latent sphere, $\alpha = 0.5$, and $\beta = 0.9$. Exponential averaging suppresses rapid fluctuations in the discrepancy while retaining a persistent offset; through a locally linear root readout, it acts approximately as filtering the position innovation. The horizon decay limits how far the current correction is extrapolated. This causal adjustment uses past actions and observations to improve the rollout estimate while preserving the shared tracking objective and action search. Balanced and wrist refinement use the uncorrected predictions and geometric targets decoded from reference embeddings.

C Training and Network Architecture

C.1 Motion data and goal relabeling

Our motion prior contains 2,000 G1 motion clips: 862 ten-second LAFAN clips (Harvey et al., 2020) and 1,138 motions from the SONIC/Bones collection (Luo et al., 2026). Together they provide approximately 4.78 hours of motion at 30 Hz, covering upright locomotion, crouching, crawling, kneeling, recovery, and expressive whole-body movement. Retargeted body states provide reference goals and expert state–goal pairs for the conditional prior; robot interaction supplies the state–action transitions used to learn Dynamic and Successor.

Half of the rollout environments follow moving reference goals in 250-step windows, with randomized initial position and heading. The remaining environments collect fixed-goal behavior, including translations of up to 3 m and heading changes of up to π. Ground initializations occur with probability 0.3, providing experience for posture recovery.

Replay retains the collection goal z_b for Successor TD learning. For actor and CPR/AUX critic updates, we replace this goal with probability 0.8. The replacement is an encoded replay state with probability 0.4 and an encoded motion-reference state with probability 0.6. Thus the marginal training-goal distribution is

$$p_g = 0.20\, p_{\text{behavior}} + 0.32\, p_{\text{replay}} + 0.48\, p_{\text{reference}}. \tag{28}$$

Replay candidates are independently reassigned by a batch permutation. This exposes policy improvement to attained states and reference behaviors while retaining goals encountered during collection.

C.2 Shared representation and core parameters

The shared latent has dimension $d = 256$ and radius $\sqrt{d}$. Table 4 lists the network architecture and parameter counts. The state encoder, History Encoder, Dynamic, Successor, and actor use residual multilayer perceptrons; Dynamic and Successor each have two ensemble members. The History Encoder learns the shared coordinates by distillation:

$$\mathcal{L}_{\text{obs}} = \mathbb{E}_{\mathcal{B}}\left[\|E_\eta^{\text{obs}}(o_t) - \text{sg}[E_\phi(s_t)]\|_2^2\right]. \tag{29}$$

The actor combines this embedding, the goal, and recent action history to produce 29 joint commands. The training encoder receives proprioception, body kinematics, and root pose; the History Encoder receives proprioception, estimated root pose, and recent observation history, supplying the current latent for both the actor and Dynamic at deployment.

The feature regularizer in Equation 10 is

$$\mathcal{L}_{\text{orth}} = -\frac{1}{B}\sum_i \|z_i\|_2^2 + \frac{1}{2B(B-1)}\sum_{i\neq j}(z_i^\top z_j)^2, \tag{30}$$

which encourages distinct feature directions under sphere normalization. Since $\|z_i\|_2^2 = d$ after projection onto the sphere, the first term is the constant $-d$ and is omitted in computation. A wrist head reconstructs root-relative wrist position, orientation, and velocity with coefficient 0.05, preserving fine motion information in the shared representation. A root-pose head is fitted to detached latents with coefficient 0.05 and supplies position and heading readouts for MPC.

Each collection step adds 4,096 transitions, followed by 16 replay updates. Dynamic uses one-step prediction throughout training. Its loss weight is 0.1 for the first 30 million transitions and follows a cosine decay to 0.001 at 100 million transitions. The encoder learning rate similarly decays from

Table 4: Network architecture and size. Depth denotes the main-network hidden-layer count. The CPR/AUX critics use a two-branch design similar to BFM-Zero (Li et al., 2026); each input branch adds four residual blocks, explaining their larger parameter count than Successor. Parameter counts include all ensemble members and exclude target-network copies and normalization buffers. M denotes one million parameters.

Network	Width	Depth	Members	Parameters (M)
State / goal encoder E_ϕ	1,024	6	1	7.12
History Encoder E_η^{obs}	512	4	1	1.44
Dynamic D_ν	2,048	3	2	27.44
Successor S_ξ	2,048	6	2	53.57
Actor π_θ	2,048	6	1	31.64
Wrist readout H_ζ^w	256	4	1	0.27
Root-pose readout H_ζ^ρ	128	2	1	0.05
CPR critic C	2,048	6	2	134.98
AUX critic A	2,048	6	2	134.98
Motion-prior discriminator	1,024	3	1	2.91
Core model (encoders, predictors, actor, and readouts)				121.54
All online networks, including training guidance				394.41
Actor and History Encoder only				33.08

Table 5: Core pretraining parameters.

Parameter	Value	Parameter	Value
Parallel environments	4,096	Replay capacity	5.12×10^6
Batch size	4,096	Updates / collection step	16
Discount γ	0.98	Target EMA τ	0.005
Encoder learning rate	10^{-5}	Predictor learning rate	3×10^{-4}
Actor learning rate	3×10^{-4}	Observation encoder LR	3×10^{-4}
Critic learning rate	10^{-4}	Discriminator LR	10^{-5}
Orthogonality weight	10^{-4}	Wrist reconstruction weight	0.05
CPR / AUX coefficients	0.005 each	Actor exploration std.	0.1
Optimizer	Adam	Gradient norm limit	10

10^{-5} to 10^{-7} between 50 and 150 million transitions. These schedules emphasize learning physical consequences early, then give behavioral continuation increasing influence as the representation stabilizes.

C.3 CPR AND AUX GUIDANCE

Conditional policy regularization (CPR) follows the motion-prior approach of BFM-Zero (Li et al., 2026). A discriminator learns to distinguish reference state–goal pairs from collected state–goal pairs using logistic classification with a gradient penalty. Its reward trains a goal-conditioned critic C that favors behavior compatible with the motion prior. An auxiliary critic A learns a physical-execution reward combining torque, action smoothness, joint-limit, contact, foot-orientation, and slippage penalties. Both critics use the Bellman target

$$y_t^j = r_t^j + \gamma_t \overline{Q}^{\,j}(s_{t+1}, z_g, a_{t+1}), \qquad j \in \{\text{CPR}, \text{AUX}\}, \quad a_{t+1} \sim \pi_\theta(\cdot \mid o_{t+1}, z_g), \tag{31}$$

where $Q^{\text{CPR}} = C$ and $Q^{\text{AUX}} = A$; the physical reward is scale-normalized.

For scalar ensemble predictions $q_1, \ldots, q_m$, define

$$\text{Pess}_\kappa(q) = \frac{1}{m}\sum_i q_i - \frac{\kappa}{m(m-1)}\sum_{i \neq j} |q_i - q_j|. \tag{32}$$

The two Dynamic and two Successor members give four composed values $q_{ij} = \langle (1-\gamma) z_t + \gamma S_j(D_i(z_t, a_t), z_g), z_g \rangle$. Using $\kappa = 0.25$ for this ensemble and for the two-member critics, the

complete actor objective is

$$\mathcal{L}_\pi = -\frac{1}{d}\mathbb{E}\left[q^- + w\left(0.005C^- + 0.005A^-\right)\right], \qquad w = \mathrm{sg}\left[\mathbb{E}|q^-|\right], \tag{33}$$

where $^-$ denotes pessimistic aggregation. The detached scale w keeps the prior guidance proportional to the learned goal value. CPR promotes coherent movement and AUX shapes physical execution, complementing the goal objective supplied by Dynamic and Successor.

D Learning Dynamics

D.1 Convergence of posture and world alignment

Q2 (Section 4.2) compares the final learned behaviors; Figure 7 follows their acquisition throughout training. We compare BeJEPA with the merged predictor and two split models that stop either Dynamic's or Successor's gradient to the encoder, evaluating their actors directly under matched tracking and fixed-goal protocols.

Local posture tracking improves quickly across models, while global tracking and fixed-goal errors expose differences in learning world alignment. The merged predictor must learn immediate action effects and goal-conditioned continuation together, without a separately supervised physical transition model. The split model with Dynamic's encoder gradient stopped benefits from separating these predictions, but its representation receives no direct physical-prediction gradient; world alignment remains slower and less accurate than in full BeJEPA.

When only Dynamic among the two predictors updates the encoder, convergence is generally better than with the merged predictor or stopped Dynamic gradient. This ablation retains a trained Successor for policy improvement, but removes its influence on the representation. Its remaining fixed-goal position and heading errors indicate that physical predictability alone is insufficient for accurate longer-term goal reaching. Full BeJEPA combines physical supervision with goal-conditioned continuation learning in the shared encoder, yielding faster world alignment and lower residual goal errors.

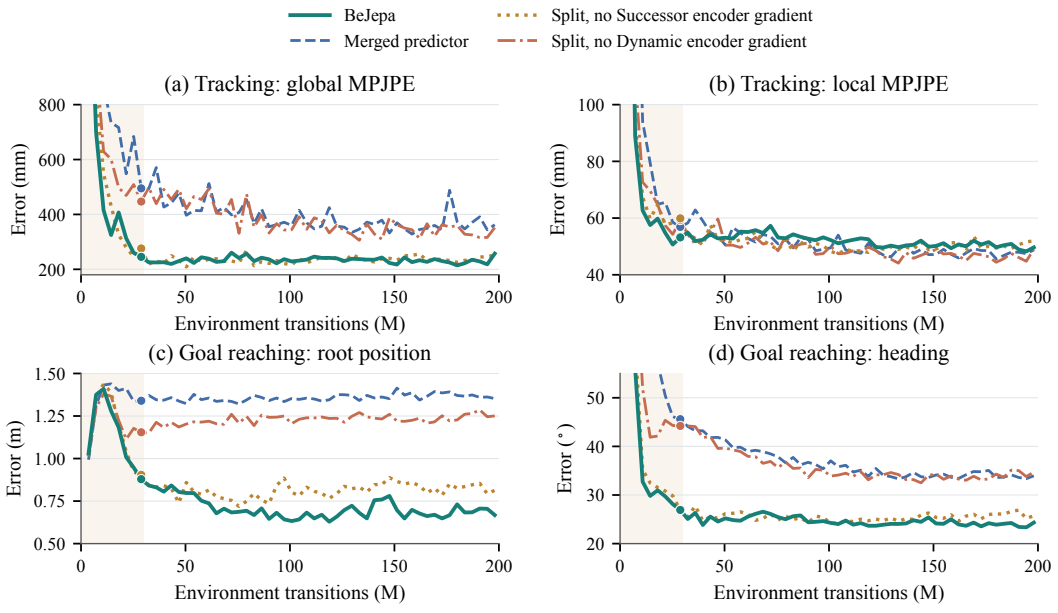

Figure 7: **Tracking and fixed-goal errors over training.** (a,b) Global and local tracking MPJPE. (c,d) Mean position and heading errors from the separate fixed-goal evaluation. The two split ablations stop the indicated predictor's gradient to the encoder while retaining both predictors; shading and markers highlight early learning.

D.2 How Dynamic and Successor shape the encoder

Let $g_D = \nabla_\phi \mathcal{L}_D$ and $g_S = \nabla_\phi \mathcal{L}_S$ denote the encoder gradients of the unweighted prediction losses. We examine their directional alignment and their contributions after applying the training weights:

$$c_{DS} = \frac{\langle g_D, g_S \rangle}{\|g_D\|_2 \|g_S\|_2}, \qquad G_D = \lambda_D \|g_D\|_2, \qquad G_S = \|g_S\|_2. \tag{34}$$

These quantities are computed on the shared encoder, where physical prediction and behavioral continuation influence the same coordinates.

Early learning: physical prediction establishes the latent coordinates. During the first 5 million transitions, the median weighted gradient ratio G_D/G_S is 2.63; during 5–10 million it is 1.02, as Successor catches up (Figure 8a–c). Dynamic therefore supplies the dominant initial prediction signal, with the two contributions becoming comparable within the first 10 million transitions. Its prediction error drops rapidly over this interval. Observed action consequences can thus shape the encoder before reliable goal-conditioned behavior has formed, providing the physical coordinates in which longer-term continuation is subsequently learned.

Intermediate learning: behavioral continuation builds on physical structure. After roughly 15 million transitions, Successor becomes the larger prediction gradient: the median G_D/G_S is 0.39 during 15–30 million and 0.20 during 30–100 million. Meanwhile, mean gradient cosine similarity rises from 0.119 during 10–15 million to 0.160 during 15–30 million and 0.226 during 30–100 million (Figure 8a,b). The change in magnitude therefore accompanies greater directional agreement: Successor increasingly organizes the representation around long-term behavior, while Dynamic continues to constrain it through predictable physical change. For a small step along the Dynamic gradient, holding prediction targets fixed,

$$\mathcal{L}_S(\phi - \alpha g_D) - \mathcal{L}_S(\phi) = -\alpha \langle g_S, g_D \rangle + O(\alpha^2). \tag{35}$$

Positive alignment identifies a shared local descent direction for the two objectives. Its increase during this stage supports the interpretation that predictable action consequences provide useful representational structure for learning behavioral continuation.

Late learning: the objectives settle with unequal influence. After 100 million transitions, when λ_D has decayed to 0.001 and the control curves approach a plateau, the median G_D/G_S falls to 0.0031 and mean cosine similarity falls to 0.069 (Figure 8a–c). All logged interval-averaged cosines remain positive, and both prediction errors settle at low values. Together, these observations are consistent with physical prediction establishing useful coordinates early, the two objectives jointly shaping long-term representations in the intermediate stage, and Successor dominating later refinement as behavior stabilizes.

E Additional Experiments

E.1 Acquiring position, heading, and a simple posture together

Can the policy jointly reach a world position and a requested body posture? We command standing, arms-out, and shallow-squat goals at displaced positions and headings, and track root-position, yaw, and joint-angle errors throughout the approach and subsequent holding period. Figure 9 resolves the initial adjustment densely and follows the same executions after arrival.

The initial joint-error response differs across postures: it rises from zero for standing, but falls for arms out and shallow squat. This is consistent with a shared need to adopt a moving configuration before settling into the final pose. In particular, remaining exactly in the standing target would prevent the leg motion needed to walk. As the robot approaches the destination, position and heading errors decrease together, while joint errors leave the initial locomotion transient and enter a lower, bounded range. The resulting behavior is a gradual transition from traveling toward the goal to adjusting the requested posture near it. It does not require a manually specified switch between walking and posing.

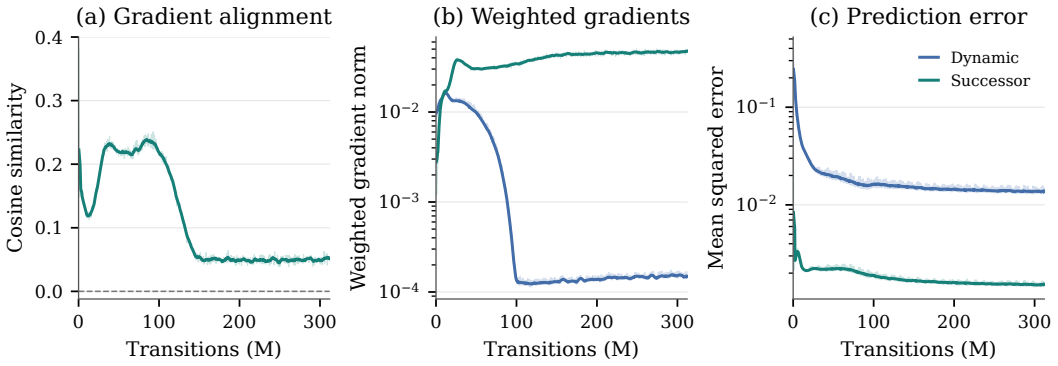

Figure 8: **Encoder-gradient alignment, weighted norms, and prediction losses.** (a) Gradient cosine similarity on the shared encoder. (b) Gradient norms after applying loss coefficients. (c) Unweighted prediction losses averaged over latent coordinates and ensemble members. Faint lines show averages over each logging interval; solid lines show a nine-record moving average. Panels (b,c) use logarithmic vertical axes and share the legend in (c).

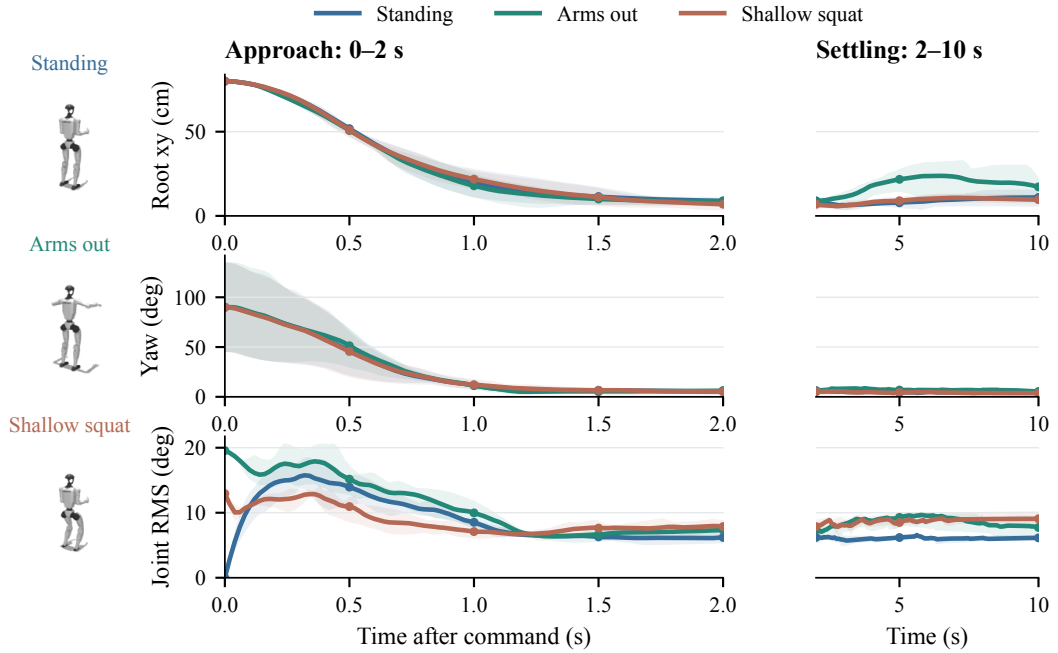

Figure 9: **Root-position, heading, and joint-angle errors during fixed-goal execution.** Lines show mean errors; shading shows the interquartile range. The early approach is expanded horizontally, with markers at 0, 0.5, 1, 1.5, 2, 5, and 10 s. The illustrated postures receive the same position and heading commands.

Table 6: **Simultaneous goal acquisition (%).** Cumulative fraction satisfying root-position error ≤ 0.2 m, yaw error $\leq 15°$, and joint RMS $\leq 10°$ together for 0.5 s. This measures acquisition; the later curves in Figure 9 show holding behavior.

Time (s)	0	0.5	1	1.5	2	5	10
Standing	0.0	0.0	0.0	12.5	87.5	100.0	100.0
Arms out	0.0	0.0	0.0	0.0	87.5	100.0	100.0
Shallow squat	0.0	0.0	0.0	25.0	75.0	100.0	100.0

Table 6 confirms that the three requirements are attained simultaneously, rather than in separate executions. After acquisition, the errors remain bounded near the goal, although residual motion persists, especially for arms out. Together, the approach and holding curves support coordinated world alignment and posture control; they do not imply exact, motionless convergence.

E.2 How far can Dynamic predict without a new observation?

How long does Dynamic remain useful when no new state observation is available? Starting from an encoded hardware state, we recursively propagate the executed actions,

$$\widehat{z}_0 = E_\phi(s_t), \qquad \widehat{z}_{k+1} = D_\nu(\widehat{z}_k, a_{t+k}), \tag{36}$$

and compare the predicted features and their root-pose readouts with the measured future states. No intermediate observation corrects the rollout.

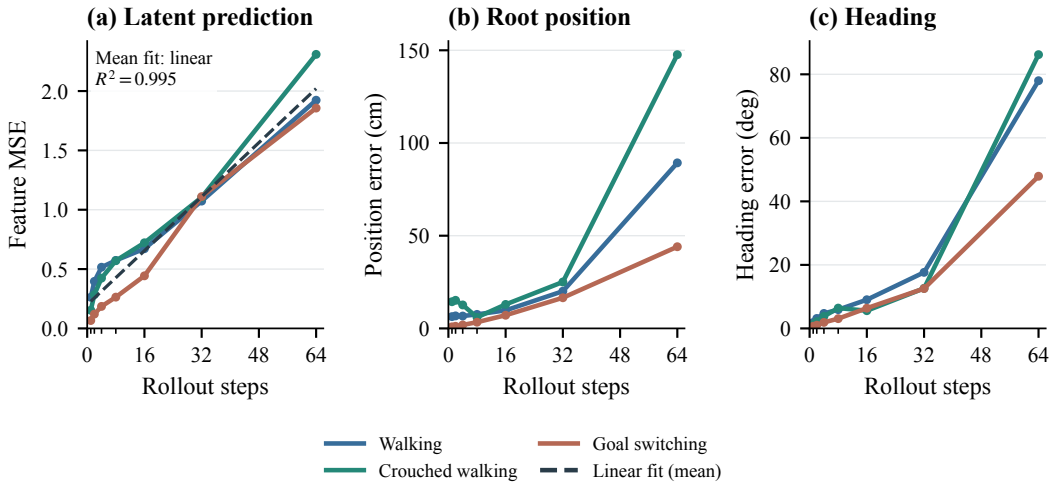

Figure 10: **Feature and root-pose prediction errors versus Dynamic rollout horizon.** Markers evaluate $1, 2, 4, 8, 16, 32$, and 64 steps, with 20 ms per step. (a) Feature MSE; the dashed line fits the mean across executions as a linear function of elapsed time. (b,c) Root-position and heading errors, including readout error. Each colored curve averages over starting points within an execution.

The average feature error grows approximately linearly over the measured range; a quadratic fit offers little additional explanatory power. The physical readouts remain local over roughly the first eight steps (160 ms), then show more pronounced error growth at longer horizons: the forecasts eventually drift substantially in both position and heading. Thus the one-step predictor should not be interpreted as a uniformly accurate long-range simulator.

These results support treating Dynamic as a local model of action consequences. In BeJEPA, the subsequent behavior is represented by Successor, which learns discounted future features directly from observed transitions.

E.3 Separating destination from arrival heading

Can the robot reach destinations in different directions while independently controlling its arrival heading? We place stationary posture goals on a circle around the initial base position. In one condition, target headings point radially outward; in the other, every destination shares the same world heading. The resulting paths and heading vectors distinguish control of world position from simply walking forward along the body's facing direction.

With radial targets, heading vectors turn toward the requested outward orientation as the paths approach the circle. With a fixed target heading, the robot reaches destinations to its sides and behind it while arriving with a common orientation. Travel direction and arrival heading therefore need not coincide. This supports the role of the world-position and yaw components in goal conditioning:

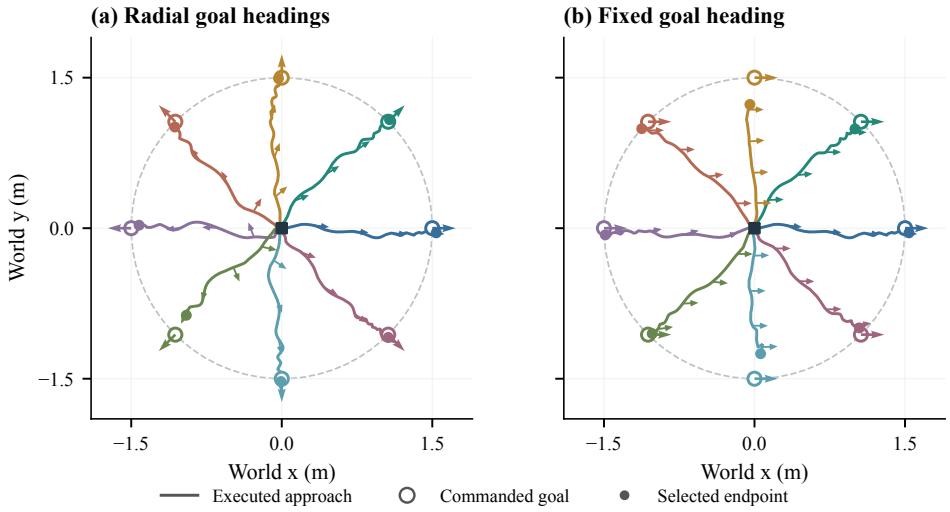

Figure 11: **Planar root trajectories and base headings for circularly arranged goals.** Targets lie on a 1.5 m circle. Open markers and arrows denote target position and yaw; filled markers denote the displayed endpoints. Short, equal-length arrows show base heading along the path, not velocity. We show the arms-out posture with the lowest mean combined alignment error across conditions, tracing each path to its best simultaneous position, heading, and joint alignment within the evaluation window.

the policy can adjust translation and orientation jointly instead of tying each destination to a single forward-facing approach. The figure illustrates the approach to the selected endpoint; sustained holding is assessed separately in Figure 9.