CONTRASTIVE REINFORCEMENT LEARNING OVER ADVERSARIAL SKILL PRIORS FOR LONG-HORIZON HUMANOID NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon humanoid navigation couples dynamic balance with goal-directed exploration across large environments. Contrastive reinforcement learning (CRL) enables goal-conditioned exploration without reward engineering, but end-to-end training on a 28-DoF humanoid forces joint-space exploration, spending most of the sample budget discovering basic locomotion rather than navigating. Pretrained motor priors provide stable gaits from decoded latents, yet they are typically driven by policy-gradient methods that struggle with long horizons. We propose ASE-CRL, which pairs a frozen Adversarial Skill Embedding (ASE) prior learned from motion capture with a contrastive goal-conditioned learner. Neither tier requires reward shaping: ASE trains via adversarial imitation, and CRL learns purely through goal identification. The advantage of the contrastive objective over reward-driven learning grows with maze scale once exploration operates over skills rather than joint targets. ASE-CRL solves every maze layout in our suite, exceeding 97% success on the largest, whereas reward-driven and end-to-end baselines collapse as navigation horizons grow. On whole-body pushing, ASE-CRL is the only goal-conditioned method that transfers from character to object goals, where hindsight-relabeled SAC fails. Finally, we release HGC-NavBench, an Isaac Lab benchmark suite for online humanoid goal reaching.

1 INTRODUCTION

Training humanoid agents capable of long-horizon navigation remains a fundamental challenge in embodied reinforcement learning. Unlike navigation with low-dimensional embodiments, where learning can focus primarily on *where to move*, humanoid agents must also learn *how to move*. That is, they need to discover stable, human-like locomotion through high-dimensional continuous control while learning to explore the environment and reach distant goals.

Two lines of work have addressed different aspects of this problem. Contrastive reinforcement learning (CRL) targets the navigation half, learning goal-reaching from hindsight-relabeled experience without hand-designed reward shaping (Eysenbach et al., 2022). Recent work further shows that scaling critic and actor depth can substantially improve CRL on long-horizon navigation tasks (Wang et al., 2025). Yet, these gains have transferred only partially to humanoid control. Methods that are competitive on AntMaze (8-DoF) degrade sharply on HumanoidMaze (Park et al., 2025); depth scaling does improve CRL on the 17-DoF Brax humanoid mazes of Wang et al. (2025), but only the smaller layouts are solved, and CRL still does not outperform SAC (Haarnoja et al., 2018a) there despite the same scaling strategy providing no benefit to SAC. We observe the same limitation with our 28-DoF humanoid: end-to-end CRL can eventually solve smaller mazes, but becomes ineffective as the navigation horizon grows. This raises our central question: *what can contrastive RL achieve when relieved of learning locomotion?*

The other line addresses humanoid control through pretrained motor priors learned by adversarial imitation, such as Adversarial Skill Embeddings (ASE) (Peng et al., 2022), which expose a latent skill space a higher-level controller can act in instead of issuing joint-level commands. While such priors provide stable, human-like locomotion, their high-level policies are typically trained with

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.

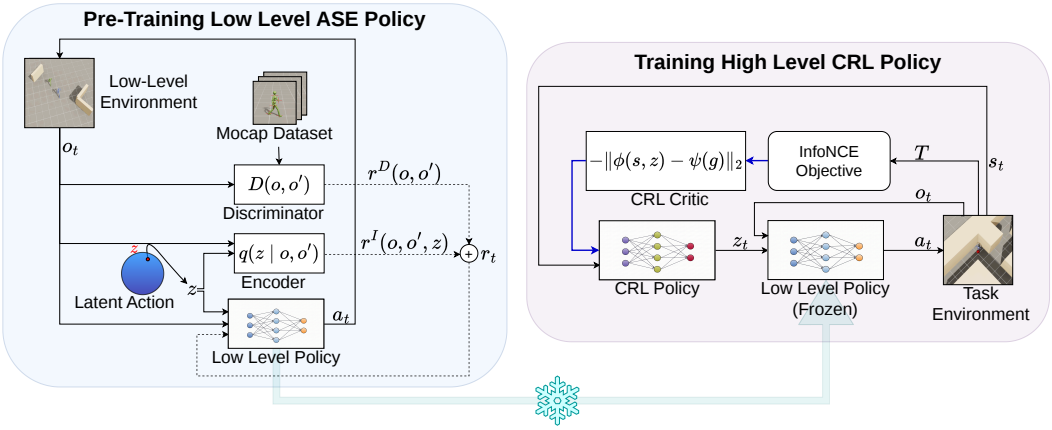


Figure 1: System overview. The ASE prior (left) is pretrained once by adversarial imitation from motion capture and then frozen. The contrastive high-level policy (right) acts by issuing a latent skill $\boldsymbol{z}_t$ to the frozen prior. Its critic, formed by the embeddings $\phi(\boldsymbol{s}_t, \boldsymbol{z}_t)$ and $\psi(\boldsymbol{g})$, is trained with InfoNCE (Section 3) using trajectories T from the task environment, and the policy is trained to maximize the critic; both use hindsight-relabeled goals. Neither tier receives a task reward: the prior's reward is learned by the discriminator and encoder, and the high level uses none. Dark blue arrows denote gradient updates; dotted lines denote the learned reward and the low-level policy update it drives; $\boldsymbol{o}_t$ denotes the low-level proprioceptive state and $\boldsymbol{s}_t$ the high-level state. For simplicity the figure indexes all quantities by the control step t; the high level in fact acts every K steps (Section 4.1).

dense or shaped task rewards (Peng et al., 2022; Juravsky et al., 2023; Tessler et al., 2023), leaving long-horizon exploration dependent on reward engineering and conventional reward-driven RL.

Placed side by side, these two lines locate the bottleneck: not in the contrastive objective itself, but in the space over which it explores. Acting directly in joint space forces CRL to discover *how to move* before it can explore *where to go*, and a pretrained motor prior removes exactly that burden. Learned latent action spaces are already known to provide useful behavioral priors for downstream exploration (Pertsch et al., 2020; Ajay et al., 2021; Singh et al., 2021); what remains unclear is how they interact with a contrastive objective, whose learning signal depends directly on the states its own exploration reaches. We therefore hypothesize that the prior will help a contrastive learner more than a reward-driven one, and most on the longest horizons. This combination has an additional benefit: neither level requires task-specific reward engineering. The motor prior is learned through adversarial imitation and skill diversity (Peng et al., 2022), while the high-level contrastive policy learns goal-reaching purely from hindsight-relabeled experience.

We introduce ASE-CRL, which trains a contrastive goal-conditioned policy entirely in the latent skill space of a frozen ASE motor prior (Figure 1). To introduce temporal abstraction, our high-level policy selects a latent action every K environment steps. During this interval of length K, the latent action is held constant (i.e., temporally extended), while the low-level policy dynamically queries the evolving state to output joint controls at each simulation step. We evaluate ASE-CRL against PPO (Schulman et al., 2017) and SAC+HER (Haarnoja et al., 2018a; Andrychowicz et al., 2017) under the same motor prior, isolating the effect of the high-level learning objective, and against end-to-end PPO and CRL to evaluate the overall benefit of motor abstraction (Section 5).

The two factors interact rather than simply adding. At the end-to-end tier the objective buys little on the smaller layouts, where CRL and PPO are comparable and both trail every hierarchical arm. The gap opens on Hard, where end-to-end CRL reaches four times end-to-end PPO's success rate despite PPO consuming a shaped reward it does not, but by Giant both end-to-end arms have collapsed. The motor prior benefits PPO as well, with ASE-PPO outperforming end-to-end PPO at every maze scale, but not enough for long-horizon navigation. What carries the largest layouts is goal conditioning over the prior, and the contrastive objective is the best of the goal-conditioned choices: on Giant, ASE-CRL reaches 97% against 82% for ASE-SAC+HER and 37% for ASE-PPO

(Table 1). Measured as the fraction of the end-to-end failure rate it removes, the prior closes 97 to 99% of the gap for CRL on every layout but 36 to 87% for PPO, with the widest difference on Giant (97% against 36%). Concretely, we contribute the following.

- **Empirical characterization of CRL's challenge in high-DoF humanoid control.** We show that end-to-end CRL struggles to scale to long-horizon navigation with a 28-DoF humanoid: success drops from 75% on U-Maze to below 10% on Giant Maze, with substantial variance across seeds. End-to-end CRL also requires roughly six times the per-step compute of its hierarchical counterpart.
- **ASE-CRL: contrastive goal-conditioned learning over a pretrained motor prior.** We introduce ASE-CRL, a hierarchical RL approach that trains a contrastive high-level policy over the latent skill space of a frozen ASE prior. ASE-CRL outperforms all baselines across every maze scale. A proprioception ablation further shows that this advantage cannot be explained by the compact high-level observation (Table 1, Section A.6). The advantage extends to whole-body pushing, where ASE-CRL reaches 98% in the open arena against 68% for ASE-PPO and 16% for ASE-SAC+HER, and 22% against 8% and 0% on the maze-embedded variant that remains hard for every arm.
- **HGC-NavBench: a benchmark for online humanoid goal-conditioned control.** We release HGC-NavBench (High-level Goal-Conditioned Navigation Benchmark), a GPU-parallel Isaac Lab benchmark comprising six tasks for online humanoid learning: four increasingly large navigation mazes and two whole-body object-pushing tasks.

2 Related Work

Goal-conditioned and contrastive reinforcement learning. Goal-conditioned RL trains a single policy to reach arbitrary goals rather than optimizing for a tailored reward, making it a natural fit for long-horizon navigation where dense rewards are difficult to design (Kaelbling, 1993; Schaul et al., 2015). A prominent line of work replaces scalar reward regression with alternative learning targets, casting reachability as recursive classification (Eysenbach et al., 2021), regressing dynamical distances (Hartikainen et al., 2020), or learning a forward-backward representation of the dynamics of a reward-free MDP (Touati & Ollivier, 2021). Eysenbach et al. (2022) show that contrastive representation learning can be cast as an RL algorithm. Wang et al. (2025) show that increasing the depth of contrastive critic and actor networks well beyond the 2 to 5 layers typical of prior work yields gains of $2\times$ to $50\times$ across locomotion, navigation, and manipulation tasks, with the largest gains on long-horizon navigation tasks. We adopt their residual architecture and InfoNCE critic. Two refinements bear on our choices without being adopted, so that the high-level learner stays identical to the end-to-end baseline: quasimetric RL (Wang et al., 2023), whose learned temporal distance can represent the directional asymmetry that our symmetric critic cannot, and contrastive difference predictive coding (Zheng et al., 2024), whose temporal-difference occupancy estimator our geometric relabeling approximates by Monte Carlo. Both represent natural extensions to our approach.

Evaluating such methods has largely relied on offline suites such as OGBench (Park et al., 2025), whose 21-DoF HumanoidMaze is the closest offline benchmark to ours. For online evaluation, massively parallel simulators such as Isaac Gym (Makoviychuk et al., 2021) and Isaac Lab (Mittal et al., 2023; NVIDIA, 2025) have traditionally emphasized velocity tracking and rough-terrain locomotion over long-horizon, sparse-reward goal reaching. While JaxGCRL (Bortkiewicz et al., 2025) provides GPU-accelerated goal navigation for lower-DoF embodiments, an online counterpart in Isaac Lab for humanoids of our scale remains absent; we introduce HGC-NavBench to fill this gap.

Motor priors and learned latent action spaces. A parallel line of work in physics-based character animation learns reusable motor skills by imitation, sidestepping task-specific locomotion rewards. Early formulations relied on per-clip tracking objectives (Peng et al., 2018) or scaled tracking to large motion libraries through a mixture of experts (Won et al., 2020). Adversarial Motion Priors (Peng et al., 2021) showed that a discriminator trained on unstructured motion-capture clips can instead supply a style reward, producing naturalistic movement without explicit tracking terms. ASE (Peng et al., 2022) adds a mutual-information objective to learn a continuous latent skill space,

which when frozen exposes a temporally extended, behaviorally structured action interface to downstream policies. Later work makes that interface more directable (Juravsky et al., 2023; Tessler et al., 2023) or rebuilds it with different generative machinery, distilling a large-scale motion tracker (Luo et al., 2023) into a variational latent space (Luo et al., 2024). A different related strategy appears outside character animation, where continuous action priors or latent plan spaces are learned from unstructured play or demonstrations, as in SPiRL (Pertsch et al., 2020), Learning Latent Plans from Play (Lynch et al., 2019), OPAL (Ajay et al., 2021), and PARROT (Singh et al., 2021). Where these latent spaces are used for downstream control, the downstream policy is typically optimized with a per-task reward. The notable exception, Lynch et al. (2019), is goal-conditioned but trained by imitation on play data rather than by reinforcement learning. We treat ASE's skill space as a fixed, off-the-shelf abstraction and replace the downstream reward with a contrastive objective.

Hierarchical control for humanoid locomotion and navigation. Hierarchical decomposition has a long history in legged locomotion and continuous control, separating low-level motor execution from high-level planning (Heess et al., 2016; Merel et al., 2019a). HIRO (Nachum et al., 2018) establishes the subgoal template, featuring a high-level policy proposing goals in state space and an off-policy correction for the drifting low level; DeepLoco (Peng et al., 2017) applies a similar split to humanoid locomotion by placing a footstep planner over a tracking controller. Outside the adversarial-imitation works, neural probabilistic motor primitives (Merel et al., 2019b) established a comparable interface, distilling motion-capture data into a latent space decoded by a frozen low-level policy, with a high-level policy commanding latents for downstream tasks. Our low-level interface is ASE (Peng et al., 2022); our low-level prior is learned adversarially and the high-level objective is contrastive rather than using a task-specific reward. Within goal-conditioned RL, subgoal-based methods such as RIS (Chane-Sane et al., 2021) and HIQL (Park et al., 2023) demonstrate that deriving subgoals from a learned value function substantially improves long-horizon credit assignment. However, these subgoals are learned end-to-end from the value function rather than grounded in a pretrained motor prior.

3 PRELIMINARIES

Before presenting our ASE-CRL method, this section reviews two foundational concepts: goal-conditioned contrastive RL and adversarial skill embeddings (ASE). We detail the hierarchical policy structure used, focusing on how the low-level policy (ASE) remains frozen while the high-level policy (goal-conditioned contrastive RL) is trained on top of it. We use $\boldsymbol{o}_t$ to refer to the proprioceptive state given to the low-level policy and $\boldsymbol{s}_t$ to refer to the state given to the high-level policy (detailed in Section 4).

Goal-conditioned RL. Goal-conditioned RL is reinforcement learning over a goal-conditioned MDP, defined as a tuple $\mathcal{M}_g = (\mathcal{S}, \mathcal{A}, p_0, p, p_g, r_g, \gamma)$, in which the agent interacts with the environment to reach arbitrary goals $\boldsymbol{g} \in \mathcal{G}$ (Kaelbling, 1993; Schaul et al., 2015). A mapping $\varphi : \mathcal{S} \to \mathcal{G}$ relates states to the goals they realize, and the reward is the probability density of reaching the goal at the next step, $r_g(\boldsymbol{s}_t, \boldsymbol{a}_t) \triangleq (1-\gamma)\, p(\varphi(\boldsymbol{s}_{t+1}) = \boldsymbol{g} \mid \boldsymbol{s}_t, \boldsymbol{a}_t)$. Under this definition, the Q-function of a goal-conditioned policy $\pi(\boldsymbol{a} \mid \boldsymbol{s}, \boldsymbol{g})$ is exactly the discounted state visitation distribution, $Q^\pi(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g}) \triangleq p^\pi_\gamma(\boldsymbol{g} \mid \boldsymbol{s}, \boldsymbol{a})$ (Eysenbach et al., 2022). Value estimation can therefore be viewed as a density-estimation problem.

Contrastive RL. Eysenbach et al. (2022) estimate p^π_γ as a density ratio by contrastive learning (Gutmann & Hyvärinen, 2012). We use the InfoNCE loss (van den Oord et al., 2018), specifically following the formulation of Eysenbach et al. (2022, Appendix C). This scores a positive future goal $\boldsymbol{g}_+ = \varphi(\boldsymbol{s}_+)$, with $\boldsymbol{s}_+ \sim p^\pi_\gamma(\cdot \mid \boldsymbol{s}, \boldsymbol{a})$, against negative goals $\boldsymbol{g}^{(i)}_- = \varphi(\boldsymbol{s}^{(i)}_-)$ sampled from the marginal distribution $\boldsymbol{s}_- \sim p^\pi_\gamma(\cdot)$ under a shared softmax,

$$\mathcal{L}_{\text{NCE}} = -\,\mathbb{E}_{(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}_+),\,\{\boldsymbol{s}^{(i)}_-\}}\left[\log \frac{\exp C(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g}_+)}{\exp C(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g}_+) + \sum_i \exp C(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g}^{(i)}_-)}\right]. \tag{1}$$

Much like standard contrastive representation learning, this objective learns representations such that a state–action pair and the future goals it actually reaches (positive pairs) have similar represen-

tations, whereas the state–action pair and unrelated goals (negative pairs) have dissimilar representations. At its optimum C recovers the log goal-conditioned action-value function up to two additive terms, one depending only on the goal and one only on the state and action, so representation learning and value learning coincide: no Bellman target is regressed and no reward function is required at any point. Section A.1 gives more details on the optimal critic.

Positive pairs come from *hindsight relabeling* (Andrychowicz et al., 2017). Instead of evaluating the policy against the originally commanded goal, we sample future states that the agent actually reached along its trajectory and relabel them as positive goals. Specifically, $\boldsymbol{s}_+$ is sampled geometrically in γ over these future states, truncated at the end of the episode, which gives an estimator of p^{π}_{γ} that is unbiased up to that truncation. This relabeling is what makes the objective reward-free in practice, since any random rollout naturally supplies valid training goals for its own prior steps.

The actor maximizes the learned critic through a reparameterized policy gradient with an automatically tuned entropy temperature, as in SAC (Haarnoja et al., 2018b). Following Wang et al. (2025) we use the negative Euclidean critic $C(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g}) = -\|\phi(\boldsymbol{s}, \boldsymbol{a}) - \psi(\boldsymbol{g})\|_2$ with deep residual encoders at both tiers.

Adversarial Skill Embeddings. Adversarial Skill Embeddings (ASE) (Peng et al., 2022) distill a large, unstructured motion-capture corpus into a continuous space of reusable motor skills without a task reward. A latent-conditioned low-level policy $\pi_{\mathrm{LL}}(\boldsymbol{a} \mid \boldsymbol{o}, \boldsymbol{z})$ outputs actions, while a discriminator evaluates realism compared to a reference dataset. Section A.1 gives the full objective. Three properties matter for what follows: the latent $\boldsymbol{z}$ is continuous and lies on the unit hypersphere in $\mathbb{R}^{64}$; distinct latents decode to distinct behaviors, so that action space is behaviorally structured; π_{LL} is frozen after pretraining and reused unchanged.

4 ASE-CRL

ASE-CRL trains a contrastive goal-conditioned policy in the latent skill space of a frozen ASE prior, so that exploration ranges over temporally extended motor behaviors rather than over joint targets. The construction below fixes the induced skill-level decision process and the observation shared by all three hierarchical arms (Section 4.1); Section 4.2 adds only what the contrastive objective itself requires.

4.1 Hierarchical formulation

Frozen low-level motor prior. We use Adversarial Skill Embeddings (ASE; Peng et al., 2022) as the low-level motor prior. ASE pretraining produces a latent-conditioned policy

$$\pi_{\mathrm{LL}}(\boldsymbol{a}_t \mid \boldsymbol{o}_t, \boldsymbol{z}_k),$$

where $\boldsymbol{a}_t \in \mathbb{R}^{28}$ specifies joint PD targets, $\boldsymbol{o}_t$ is the humanoid's proprioceptive state, and $\boldsymbol{z}_k \in \mathbb{R}^{64}$ is a latent motor skill projected onto the unit hypersphere. The prior is trained once from motion-capture data using adversarial imitation and skill diversity. The prior is additionally trained to be robust to wall collisions by spawning walls in random locations during its pretraining. After pretraining, we discard the ASE discriminator and encoder and keep π_{LL} frozen throughout all downstream tasks.

High-level control over skills. Instead of controlling the humanoid directly, the high-level policy

$$\boldsymbol{z}_k \sim \pi_{\mathrm{HL}}(\cdot \mid \boldsymbol{s}_k, \boldsymbol{g})$$

selects a latent skill conditioned on the current high-level state $\boldsymbol{s}_k$ and commanded goal $\boldsymbol{g}$. The selected latent $\boldsymbol{z}_k$ is held fixed for K low-level control steps. During these steps, the frozen motor prior repeatedly maps the current proprioceptive state and the selected skill to joint-level actions. We use $K = 5$: the low-level policy operates at 30 Hz, while the high-level policy makes decisions at 6 Hz. This fixed execution induces a stationary decision process where the actions are motor skills rather than joint commands.

In an episode containing T low-level decisions, this reduces the high-level decision horizon from T to T/K; for example, the 60-second Giant environment contains 1800 low-level control decisions but only 360 high-level skill selections.

High-level state and goal representation. For maze navigation, the high-level policy observes only the humanoid root position, root orientation, and commanded goal,

$$\boldsymbol{s} = [\, \boldsymbol{x}_{\text{root}} \in \mathbb{R}^3,\ \boldsymbol{q}_{\text{root}} \in \mathbb{R}^4,\ \boldsymbol{g} \in \mathbb{R}^3 \,]$$

giving a 10-dimensional observation. The frozen low-level policy internally consumes a 140-dimensional proprioceptive state; unlike standard ASE, where the high-level policy also receives proprioception, this state is withheld from our high-level learner. Ablating this choice shows that splicing the proprioceptive state back in leaves ASE-CRL essentially unchanged (Section A.6), while removing it actually improves ASE-PPO performance.

We define an achieved-goal projection $\varphi : \mathcal{S} \to \mathcal{G}, \quad \varphi(\boldsymbol{s}_k) = \boldsymbol{x}_{\text{root}}$, which maps each visited high-level state to the navigation goal realized by that state. This projection provides the interface required for hindsight goal relabeling.

4.2 Contrastive Goal-Conditioned Learning over Skills

We train the high-level policy with the contrastive objective of Section 3, applied to the induced stationary decision process of Section 4.1: the critic scores $(\boldsymbol{s}_k, \boldsymbol{z}_k, \boldsymbol{g})$ in place of $(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g})$, so ϕ embeds a state and a skill rather than a state and a joint command, and positives are achieved goals $\varphi(\boldsymbol{s}_{k+\Delta})$ with Δ drawn geometrically in γ over the episode-bounded future. The learned quantity is thus which latent motor skill, held for K steps by the frozen prior, carries the character toward a goal, rather than which joint target does. The critic parameterization, the actor update and the entropy temperature α are unchanged; only the arguments differ.

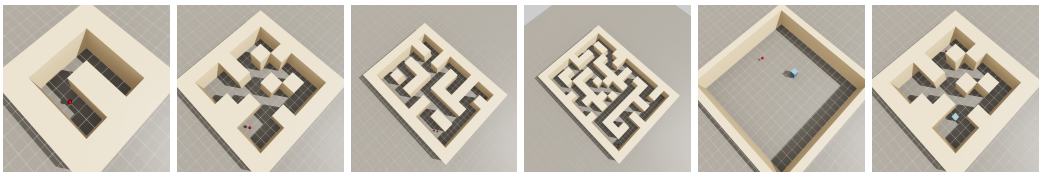
Relabeling at the macro step rather than the control step has the consequence that the relabeled goal distribution inherits the prior's coverage: positives are places a walking controller reached, not places an untrained 28-DoF body happened to fall. The contrastive critic learns from the states its own rollouts visit, so this is the mechanism by which we expect the prior to benefit a contrastive learner more than a reward-driven one, which receives long-range signal from shaping regardless of where it has been.

Algorithm 1 in the appendix summarizes the procedure.

5 Experiments

5.1 Experimental Setup

Methods compared. We evaluate five agents across two control tiers. At the *end-to-end* tier, PPO (Schulman et al., 2017) and CRL act directly on the humanoid's 28 actuated joints at 30 Hz, emitting position targets that the simulator's per-joint proportional-derivative controllers convert to torques. At the *hierarchical* tier, PPO, SAC with hindsight experience replay (SAC+HER; Haarnoja et al., 2018a; Andrychowicz et al., 2017) and CRL act on the latent of a single frozen ASE prior at 6 Hz; we refer to the last as ASE-CRL, which is our proposed approach. The three hierarchical agents share the same prior, observation, action space, control rate and episode budget, so comparisons among them isolate the high-level learning algorithm. Both goal-conditioned arms relabel at the macro step, and differ in the critic, in the relabeling distribution and in depth. ASE-CRL draws the substituted goal geometrically in γ over the episode-bounded future and relabels every sampled transition, since the pairing *is* its objective, while ASE-SAC+HER draws it uniformly over the future and relabels 80% of the batch. Each setting follows the standard configuration of its own objective, so this is a comparison between two complete goal-conditioned algorithms rather than a single-factor ablation of the critic loss: it establishes what off-policy replay with hindsight relabeling achieves over the frozen prior without a contrastive critic (Section A.4). The end-to-end and hierarchical CRL agents are one implementation attached to different environments, so comparisons between tiers vary the interface while holding the contrastive machinery fixed. The tiers differ in the several respects listed in Table 4; these are constitutive properties of the abstraction rather than nuisance variables, so we make no attempt to present end-to-end-versus-hierarchical comparisons as a controlled manipulation of a single factor. Neither end-to-end agent is run on the pushing tasks, so the pushing comparison is between the three hierarchical arms only (Section A.4).



U-Maze | Large | Hard | Giant | Push-cube (open arena) | Push-cube (Large)

Figure 2: The six HGC-NavBench environments: the four maze layouts, then whole-body pushing in an open arena and embedded in the Large layout. The red sphere marks the commanded goal and the blue block the object to be pushed. Panels are framed to fit each layout and are therefore not to a common scale, but corridor width is 2 m in every panel and can be read as the implicit ruler; extents, goal counts and episode budgets are in Table 2.

Environments. All environments were built in Isaac Lab (Mittal et al., 2023; NVIDIA, 2025) on top of MimicKit (Peng, 2025); the GPU-parallel PhysX backend both build on was introduced by Isaac Gym (Makoviychuk et al., 2021). The maze layouts emulate the AntMaze layouts of D4RL (Fu et al., 2020) and OGBench (Park et al., 2025) and are shown in Figure 2. They scale from a 10×10 m U-Maze with 6 goal cells and a 10 s limit to a 24×32 m giant maze with 85 goal cells and 60 s; Section A.3 gives the full specifications. Corridor width is held fixed at 2 m across every layout, so scale changes the navigation horizon without changing local mobility. Episode length scales with maze extent so that agents are given time commensurate with the distances involved.

The agent spawns at a fixed start cell in every episode, at both tiers and for every method. At the hierarchical tier the character is initialized from the first frame of a walking clip with a uniformly random heading, so the frozen prior begins from a pose it recognizes; the end-to-end tier begins from a fixed standing pose with fixed heading (Section A.3). Training goals are drawn uniformly from the layout's goal cells. Evaluation runs on a slice of environments appended to the 512 training environments and excluded from training: those environments step in the same batched simulation but act under a frozen greedy snapshot of the policy, and their transitions never enter the replay buffer or any gradient. The slice cycles a fixed, deterministic battery drawn from the same goal set the agent trains on, three concurrent environments per goal cell in the mazes and a frozen set of 100 spawn-target instances in the pushing tasks, so the metric does not measure generalization to unseen goals (Section A.3).

The pushing task requires the humanoid to move a 20 kg block, bottom-weighted so that it slides rather than topples under impulsive contact, until its centroid comes within 0.5 m of a goal. This is indirect goal reaching: progress is governed by where the block ends up rather than by the character's own position, and the block can be pushed but not pulled. We evaluate two variants, Push-cube (open arena) and Push-cube (Large), the latter embedded in the Large maze layout, which is harder than navigating that layout for reasons beyond horizon (Section A.3).

Architecture, training and evaluation. The contrastive agents use the residual, depth-scaled architecture of Wang et al. (2025) at width 512. The low-level ASE policy is taken unchanged from MimicKit (Peng, 2025), trained once from 62 locomotion clips in an environment with randomly spawned walls so that it is robust to collisions, and reused frozen across all four mazes and both pushing variants, so the skill space it exposes is locomotion and nothing else: on the pushing tasks the character can only move the block by walking into it.

Every method and environment combination is run with five seeds (0 to 4). Success means the achieved quantity $\varphi(\boldsymbol{s})$ comes within $\delta = 0.5$ m of the goal in the horizontal plane at any point during the episode. We report the mean over seeds with a 95% percentile-bootstrap confidence interval of that mean (10^4 resamples), each seed first summarized by the mean of its evaluation snapshots over the final tenth of its environment-step budget, and read at a shared per-environment budget. Because the interval is one for the mean rather than for the seed spread, and is coarse at five seeds, we also plot every individual seed. Reported environment steps count training environments only, and match methods within each task; the prior's one-off pretraining cost, 14.85 B environment steps, is amortized across all six environments, three high-level algorithms and five seeds, and is

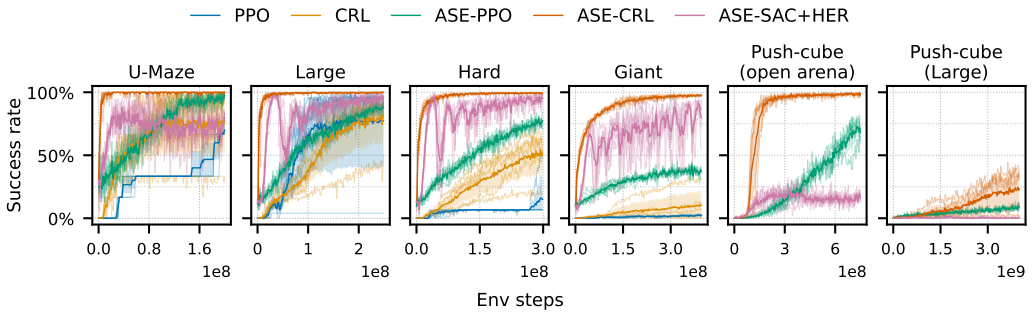


Figure 3: Success rate against environment steps, one panel per HGC-NavBench environment. The solid line is the mean over five seeds, the shaded band its 95% percentile-bootstrap confidence interval, and the thin lines the individual seeds. Each environment is drawn to its own shared budget, so the x ranges differ, and the maze-embedded pushing task runs an order of magnitude longer than the mazes.

Table 1: Success rate (%) at the end of training, mean $\pm$ standard deviation over five seeds, read at the shared per-environment budget of each task (Section 5). Dashes mark configurations not run: neither end-to-end agent is run on the pushing tasks. Table 10 repeats these numbers with 95% percentile-bootstrap confidence intervals of the mean, and Section A.12 draws every seed individually.

	Maze				Push-cube	
Method	**U-Maze**	**Large**	**Hard**	**Giant**	**Open arena**	**Large**
End-to-end PPO	61.4 ± 8.0	77.1 ± 40.9	12.0 ± 11.8	2.2 ± 0.6	—	—
End-to-end CRL	75.4 ± 27.6	78.8 ± 21.1	50.4 ± 18.8	9.7 ± 12.3	—	—
ASE-PPO	94.8 ± 1.6	87.0 ± 1.5	75.4 ± 1.2	37.3 ± 1.0	68.0 ± 9.6	8.1 ± 0.8
ASE-SAC+HER	72.7 ± 3.6	94.7 ± 2.6	93.8 ± 2.6	82.1 ± 5.6	15.6 ± 2.2	0.2 ± 0.3
ASE-CRL	$\mathbf{99.8} \pm 0.2$	$\mathbf{99.8} \pm 0.1$	$\mathbf{99.4} \pm 0.2$	$\mathbf{97.4} \pm 0.5$	$\mathbf{98.5} \pm 0.5$	$\mathbf{22.3} \pm 15.8$

accounted in full in Section A.13. Hyperparameters, per-seed curves and per-run compute are in Section A.4, Section A.12 and Section A.13; Section A.5 specifies the evaluation protocol in full.

5.2 Results

Table 1 reports final success rates, and Figure 3 shows learning curves across environments. While smaller layouts saturate near ceiling, the harder mazes and pushing tasks cleanly separate both the sample efficiency and final stability of each paradigm.

Contrastive goal conditioning is key across scales and tasks. Because the maze layouts hold corridor width fixed and vary only scale and complexity, movement along Table 1 directly isolates the navigation horizon. Over the shared frozen prior, both goal-conditioned methods overtake ASE-PPO as this horizon expands. This divergence widens to a sixty-point margin on Giant, where ASE-PPO collapses to 37.3% while ASE-CRL degrades by only two points from U-Maze (99.8% to 97.4%). While ASE-SAC+HER shares the macro-step relabeling interface and prior with ASE-CRL, it falters at the extremes: on U-Maze (72.7%), where re-evaluating the same checkpoints under a longer budget shows it reaches the far goals but too slowly for the 10 s limit, closing to within a point of ASE-PPO by 20 s (Section A.8); on Giant (82.1%) (Section A.7); and most acutely on pushing, where achieved goals remain static until contact occurs. In the open arena it reaches 15.6%, and once the push is embedded in the maze it does not learn the task at all (0.2%, no seed above 0.6% over a 4 B-step budget), below even ASE-PPO's 8.1%: the block moves so rarely that there is essentially nothing to relabel on. The same agent reaches 82.1% on Giant, so the failure is one of the objective under indirect goals, not of the implementation. ASE-CRL suffers no such reversals,

maintaining tight seed consistency on every maze, reaching 98.5% on open-arena pushing, and remaining the only algorithm to achieve any notable success on the maze-embedded push (22.3%). Hindsight relabeling over frozen skills provides the foundation, but the contrastive critic enables superior long-horizon exploration and is the only goal-conditioned objective that survives indirect manipulation.

The objective matters little without the abstraction. At the end-to-end tier, both objectives perform similarly on the two smaller layouts and collapse on Giant. They separate only on Hard, where end-to-end CRL achieves four times end-to-end PPO's success rate, notably doing so without any reward signal while PPO relies on dense shaping (Equation (3)). Regardless, neither is competitive with the hierarchical tier: both trail every hierarchical arm at all evaluated scales and remain unstable across seeds.

The compact interface is not the explanation. Splicing the withheld 140-dimensional proprioceptive state into the high-level observation leaves ASE-CRL essentially unchanged on Giant, 97.4 against 97.9, and more than halves ASE-PPO, 37.3 against 14.3. The hierarchical gap is therefore not an artifact of the 10-dimensional interface: the contrastive agent comfortably tolerates (and marginally benefits from) the extra dimensions, whereas the reward-driven one is actively hurt by them (Section A.6).

6 Discussion and Conclusion

Trained over a frozen ASE prior, a contrastive goal-conditioned policy solves every maze layout in HGC-NavBench, and its margin over ASE-PPO widens with the navigation horizon: on Giant it reaches 97% against 82% for hindsight-relabeled SAC and 37% for reward-driven PPO over the same prior. The two ingredients do not simply add. The abstraction alone improves a reward-driven learner at every scale but not enough to carry the largest layouts, and the contrastive objective alone, at the end-to-end tier, spends its budget discovering locomotion before it can explore. The prior does not enable CRL, since end-to-end CRL runs without it; it determines how far CRL scales.

Several limitations bound these claims. The two tiers differ in observation, action space, control rate, initialization and access to motion data at once (Table 4), so we read that comparison as the joint effect of the abstraction and the objective rather than as a controlled manipulation of either; we isolate one of those axes, the decision rate, and find it accounts for none of the gap on its own (Section A.9). The prior was trained in an environment with randomly spawned walls, a downstream-informed choice, offset by being reused frozen across four mazes and both pushing variants. The start cell is fixed in every episode and the evaluation battery is drawn from the same goal set the agent trains on, so we measure goal-reaching reliability rather than generalization to unseen goals. We report no motion-quality metric and therefore make no claim about naturalness, though we do measure the specific jitter failure mode Peng et al. (2022) identify, the step-to-step change in the commanded latent, and find that both goal-conditioned arms command markedly smoother latent sequences than the reward-driven one (Section A.10). Two further limitations, the symmetry of the negative-Euclidean critic and the state the high-level observation omits, are set out in Section A.14.

Pushing tests whether the interaction survives indirect goal reaching, where the achieved goal is the block's position rather than the character's own. It is the regime in which ASE-SAC+HER does worst, falling to 15.6% in the open arena and 0.2% in the maze (Section 5.2). By contrast, ASE-CRL leads in the open arena and remains the only algorithm to achieve any notable success once the push is embedded in a maze. Even so, no method comes close to solving the maze-embedded task. We hypothesize that the combination of indirect navigation and an extended horizon creates a severe bottleneck: because the locomotion-only prior moves the block only by walking into it, the environment yields extremely sparse feedback. For relabeling-based methods, this lack of novel object states means there are almost no meaningful achieved goals to relabel, effectively starving the learner. We release HGC-NavBench, its six environments and the full training and evaluation pipeline so that it, and the scaling behavior above, can be tested against other goal-conditioned objectives on the same footing.

AI USE STATEMENT

In this work we used a generative AI assistant in three ways. For the manuscript, it reviewed drafts of the abstract, introduction, related work, preliminaries and experimental setup for clarity, internal consistency and unsupported claims, and proposed revised wording that the authors reviewed and edited; it additionally drafted portions of the appendix by extracting environment, reward, hyperparameter and compute specifications from our released code, and the authors checked every such passage against the source. For the codebase, it drafted portions of the implementation, which the authors reviewed and tested before use. It was also used to suggest candidate related work; every suggested reference was verified by the authors against the primary source before citation, and suggestions that could not be verified were removed.

We did not use generative AI to generate research ideas or to design or run experiments. All numerical results reported here were produced by the released code and inspected by the authors. We take responsibility for the final content of this work, including all text, claims and artifacts, including those portions drafted with AI assistance.

ETHICS STATEMENT

The low-level prior is trained by adversarial imitation from locomotion clips obtained through the upstream MimicKit release (Peng, 2025), which derive from the CMU Motion Capture Database (CMU Graphics Lab, 2003) and from SMPL/AMASS-derived sources (Loper et al., 2015; Mahmood et al., 2019). These clips are not covered by our release: they are redistributed by their original providers under their own terms of use, which we have followed and which downstream users must observe independently. Our released code and environments are Apache-2.0 licensed as a derivative of MimicKit; the motion data is deliberately excluded from that release and is obtained by users from the upstream project. The clips record human movement and, although we use them only as a distribution over kinematics rather than as data about identifiable individuals, we neither redistribute them nor attempt to recover any attribute of the performers.

REPRODUCIBILITY STATEMENT

The full training and evaluation pipeline, all environment definitions, and ready-made run configurations for every method and environment combination reported here are released as anonymous supplementary material, together with the HGC-NavBench environments. Section A.4 specifies every hyperparameter, network architecture and random seed; Section A.13 reports environment steps, gradient steps and wall-clock time per run; Section A.3 gives the complete physical parameters of the maze and pushing environments. Character assets and motion clips are obtained from the upstream MimicKit release and retain their original terms of use.

REFERENCES

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. OPAL: offline primitive discovery for accelerating offline reinforcement learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=V69LGwJ0lIN.

Marcin Andrychowicz, Dwight Crow, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5048–5058, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html.

Michal Bortkiewicz, Wladyslaw Palucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Lukasz Kucinski, and Benjamin Eysenbach. Accelerating goal-conditioned reinforcement learning algorithms and research. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=4gaySj8kvX.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1430–1440. PMLR, 2021. URL http://proceedings.mlr.press/v139/chane-sane21a.html.

CMU Graphics Lab. CMU graphics lab motion capture database. http://mocap.cs.cmu.edu/, 2003.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-Learning: learning to achieve goals via recursive classification. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=tc5qisoB-C.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/e7663e974c4ee7a2b475a4775201ce1f-Abstract-Conference.html.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. URL https://arxiv.org/abs/2004.07219.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Skc-Fo4Yg.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *J. Mach. Learn. Res.*, 13:307–361, 2012. doi: 10.5555/2503308.2188396. URL https://dl.acm.org/doi/10.5555/2503308.2188396.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1856–1865. PMLR, 2018a. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *CoRR*, abs/1812.05905, 2018b. URL http://arxiv.org/abs/1812.05905.

Kristian Hartikainen, Xinyang Geng, Tuomas Haarnoja, and Sergey Levine. Dynamical distance learning for semi-supervised and unsupervised skill discovery. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=H1lmhaVtvr.

Nicolas Heess, Gregory Wayne, Yuval Tassa, Timothy P. Lillicrap, Martin A. Riedmiller, and David Silver. Learning and transfer of modulated locomotor controllers. *CoRR*, abs/1610.05182, 2016. URL http://arxiv.org/abs/1610.05182.

Jordan Juravsky, Yunrong Guo, Sanja Fidler, and Xue Bin Peng. PADL: language-directed physics-based character control. *CoRR*, abs/2301.13868, 2023. doi: 10.48550/ARXIV.2301.13868. URL https://doi.org/10.48550/arXiv.2301.13868.

Leslie Pack Kaelbling. Learning to achieve goals. In Ruzena Bajcsy (ed.), *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*, pp. 1094–1099. Morgan Kaufmann, 1993.

Hojoon Lee, Dongyoon Hwang, Donghu Kim, Hyunseung Kim, Jun Jet Tai, Kaushik Subramanian, Peter R. Wurman, Jaegul Choo, Peter Stone, and Takuma Seno. SimBa: simplicity bias for scaling up parameters in deep reinforcement learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=jXLiDKsuDo.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015. doi: 10.1145/2816795.2818013. URL https://doi.org/10.1145/2816795.2818013.

Zhengyi Luo, Jinkun Cao, Alexander Winkler, Kris Kitani, and Weipeng Xu. Perpetual humanoid control for real-time simulated avatars. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 10861–10870. IEEE, 2023. doi: 10.1109/ICCV51070.2023.01000. URL https://doi.org/10.1109/ICCV51070.2023.01000.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris M. Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=OrOd8PxOO2.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1113–1132. PMLR, 2019. URL http://proceedings.mlr.press/v100/lynch20a.html.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: archive of motion capture as surface shapes. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 5441–5450. IEEE, 2019. doi: 10.1109/ICCV.2019.00554. URL https://doi.org/10.1109/ICCV.2019.00554.

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 Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/28dd2c7955ce926456240b2ff0100bde-Abstract-round2.html.

Josh Merel, Arun Ahuja, Vu Pham, Saran Tunyasuvunakool, Siqi Liu, Dhruva Tirumala, Nicolas Heess, and Greg Wayne. Hierarchical visuomotor control of humanoids. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019a. URL https://openreview.net/forum?id=BJfYvo09Y7.

Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. Neural probabilistic motor primitives for humanoid control. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL https://openreview.net/forum?id=BJl6TjRcY7.

Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics Autom. Lett.*, 8(6):3740–3747, 2023. doi: 10.1109/LRA.2023.3270034. URL https://doi.org/10.1109/LRA.2023.3270034.

Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 3307–3317, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/e6384711491713d29bc63fc5eeb5ba4f-Abstract.html.

Michal Nauman, Mateusz Ostaszewski, Krzysztof Jankowski, Piotr Milos, and Marek Cygan. Bigger, regularized, optimistic: scaling for compute and sample efficient continuous control. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/cd3b5d2ed967e906af24b33d6a356cac-Abstract-Conference.html.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In Ivan Bratko and Saso Dzeroski (eds.), *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999), Bled, Slovenia, June 27 - 30, 1999*, pp. 278–287. Morgan Kaufmann, 1999.

NVIDIA. Isaac Lab: A GPU-accelerated simulation framework for multi-modal robot learning. *CoRR*, abs/2511.04831, 2025. doi: 10.48550/ARXIV.2511.04831. URL https://doi.org/10.48550/arXiv.2511.04831.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. HIQL: offline goal-conditioned RL with latent states as actions. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6d7c4a0727e089ed6cdd3151cbe8d8ba-Abstract-Conference.html.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: benchmarking offline goal-conditioned RL. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=M992mjgKzI.

Xue Bin Peng. MimicKit: A reinforcement learning framework for motion imitation and control. *CoRR*, abs/2510.13794, 2025. doi: 10.48550/ARXIV.2510.13794. URL https://doi.org/10.48550/arXiv.2510.13794.

Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel van de Panne. DeepLoco: dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.*, 36(4): 41:1–41:13, 2017. doi: 10.1145/3072959.3073602. URL https://doi.org/10.1145/3072959.3073602.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. DeepMimic: example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.*, 37(4):143, 2018. doi: 10.1145/3197517.3201311. URL https://doi.org/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 Trans. Graph.*, 40(4):144:1–144:20, 2021. doi: 10.1145/3450626.3459670. URL https://doi.org/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 Trans. Graph.*, 41 (4):94:1–94:17, 2022. doi: 10.1145/3528223.3530110. URL https://doi.org/10.1145/3528223.3530110.

Karl Pertsch, Youngwoon Lee, and Joseph J. Lim. Accelerating reinforcement learning with learned skill priors. In Jens Kober, Fabio Ramos, and Claire J. Tomlin (eds.), *4th Conference on Robot Learning, CoRL 2020, 16-18 November 2020, Virtual Event / Cambridge, MA, USA*, volume 155 of *Proceedings of Machine Learning Research*, pp. 188–204. PMLR, 2020. URL https://proceedings.mlr.press/v155/pertsch21a.html.

Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In Francis R. Bach and David M. Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1312–1320. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/schaul15.html.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1506.02438.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Avi Singh, Huihan Liu, Gaoyue Zhou, Albert Yu, Nicholas Rhinehart, and Sergey Levine. PARROT: data-driven behavioral priors for reinforcement learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=Ysuv-WOFeKR.

Chen Tessler, Yoni Kasten, Yunrong Guo, Shie Mannor, Gal Chechik, and Xue Bin Peng. CALM: conditional adversarial latent models for directable virtual characters. In Erik Brunvand, Alla Sheffer, and Michael Wimmer (eds.), *ACM SIGGRAPH 2023 Conference Proceedings, SIGGRAPH 2023, Los Angeles, CA, USA, August 6-10, 2023*, pp. 37:1–37:9. ACM, 2023. doi: 10.1145/3588432.3591541. URL https://doi.org/10.1145/3588432.3591541.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 13–23, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/003dd617c12d444ff9c80f717c3fa982-Abstract.html.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018. URL http://arxiv.org/abs/1807.03748.

Kevin Wang, Ishaan Javali, Michal Bortkiewicz, Tomasz Trzcinski, and Benjamin Eysenbach. 1000 layer networks for self-supervised RL: scaling depth can enable new goal-reaching capabilities. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/e74ee34cc0f2d0780f34ee77d8fba25b-Abstract-Conference.html.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume

202 of *Proceedings of Machine Learning Research*, pp. 36411–36430. PMLR, 2023. URL https://proceedings.mlr.press/v202/wang23al.html.

Jungdam Won, Deepak Gopinath, and Jessica K. Hodgins. A scalable approach to control diverse behaviors for physically simulated characters. *ACM Trans. Graph.*, 39(4):33, 2020. doi: 10.1145/3386569.3392381. URL https://doi.org/10.1145/3386569.3392381.

Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive difference predictive coding. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=0akLDTFR9x.

A Appendix

A.1 Background

This subsection expands Section 3; the goal-conditioned MDP, the sparse arrival reward and Equation (1) are stated there and not repeated. The one thing worth making explicit first is why a value function is estimable by density-ratio methods at all: under the arrival reward, $Q^\pi(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g}) = p^\pi_\gamma(\boldsymbol{g} \mid \boldsymbol{s}, \boldsymbol{a})$ is literally the discounted state-occupancy density of the goal, so estimating the value function *is* estimating a density.

The optimal InfoNCE critic. Using the achieved-goal projection $\varphi(\boldsymbol{s}) = \boldsymbol{g}$, Equation (1) scores one positive future goal $\boldsymbol{g}_+ \sim p^\pi_\gamma(\cdot \mid \boldsymbol{s}, \boldsymbol{a})$ against negatives drawn from the marginal goal distribution $p^\pi_\gamma(\boldsymbol{g})$ under a shared softmax. The softmax normalizes over the goal argument only, so at the optimum the critic recovers the log density ratio up to any function of the state–action pair,

$$C^*(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g}) = \log p^\pi_\gamma(\boldsymbol{g} \mid \boldsymbol{s}, \boldsymbol{a}) - \log p^\pi_\gamma(\boldsymbol{g}) + c(\boldsymbol{s}, \boldsymbol{a}), \tag{2}$$

with c unidentifiable because adding it to every logit leaves the softmax unchanged. At fixed $(\boldsymbol{s}, \boldsymbol{a})$, Equation (2) therefore ranks goals by the ratio $Q^\pi(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g})/p^\pi_\gamma(\boldsymbol{g})$ rather than by the action-value function itself. The actor, however, compares actions at a fixed state and goal, since the commanded goal is constant within an episode and each relabeled goal is fixed per sample. There the $-\log p^\pi_\gamma(\boldsymbol{g})$ term is constant and drops out, but $c(\boldsymbol{s}, \boldsymbol{a})$ does not, so maximizing C^* over actions maximizes Q^π only if c does not vary with the action. InfoNCE alone does not enforce this; the LogSumExp penalty of Table 5 does. For a batch of N goals drawn from the marginal, the density ratio averages to one, so $\log \sum_i \exp C^*(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g}_i) \approx c(\boldsymbol{s}, \boldsymbol{a}) + \log N$, and regularizing this quantity toward zero holds c approximately constant across state–action pairs. Separately, the negative-Euclidean parameterization $C(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g}) = -\|\phi(\boldsymbol{s}, \boldsymbol{a}) - \psi(\boldsymbol{g})\|_2$ cannot represent an arbitrary $c(\boldsymbol{s}, \boldsymbol{a})$, so it restricts the achievable critic class; it also measures a symmetric distance in embedding space and so is limited in expressing goal pairs that are cheap to traverse in one direction only (see Section 6).

Hindsight relabeling. Positives are drawn by sampling a future index from a geometric distribution over the remainder of the same trajectory, $\Delta \sim \mathrm{Geom}(1-\gamma)$, and taking $\varphi(\boldsymbol{s}_{k+\Delta})$ as the goal. This is an unbiased estimator of p^π_γ up to episode truncation: the geometric weights $(1-\gamma)\gamma^\Delta$ are exactly the discounted occupancy weights, so sampling a future state this way and treating it as the commanded goal turns any trajectory into a positive pair for the goal it happened to reach. Negatives are the relabeled goals of the other transitions in the same minibatch, which are marginal draws from the same visitation distribution.

Adversarial motion priors and ASE. AMP (Peng et al., 2021) trains a discriminator D to separate state transitions produced by the policy from transitions sampled from a motion-capture dataset, and rewards the policy with $r_{\text{style}}(\boldsymbol{o}, \boldsymbol{o}') = -\log(1 - D(\boldsymbol{o}, \boldsymbol{o}'))$, so that naturalistic motion is obtained without per-clip tracking objectives. ASE (Peng et al., 2022) adds a latent $\boldsymbol{z}$ drawn from a uniform prior on the unit hypersphere and an encoder $q(\boldsymbol{z} \mid \boldsymbol{o}, \boldsymbol{o}')$ trained jointly with the policy, contributing a diversity reward $r_{\text{diversity}}(\boldsymbol{o}, \boldsymbol{o}', \boldsymbol{z}) = \log q(\boldsymbol{z} \mid \boldsymbol{o}, \boldsymbol{o}')$ that maximizes the mutual information between the latent and the resulting transition, in the manner of earlier mutual-information skill discovery (Gregor et al., 2017; Eysenbach et al., 2019). The low-level objective is a weighted sum of the two; no task reward enters. The result is a continuous, reusable skill space in which nearby latents decode to related movements, which is the interface our high-level policy acts through.

A.2 Full reward definitions and the reward-design ablation

Only the PPO arms consume any of the rewards in this subsection. ASE-CRL consumes no reward, and ASE-SAC+HER consumes only the sparse goal indicator of Section A.5.

Maze task reward. The contrastive and hindsight-relabeled agents are given a goal and a success radius and nothing else. End-to-end PPO and ASE-PPO additionally require a task reward, which we specify below. With $\boldsymbol{x}_t$ the character root position, $\boldsymbol{g}$ the goal and $d_t = \|\boldsymbol{x}_t - \boldsymbol{g}\|$, taking the undiscounted decrement in distance to the goal in the style of potential-based shaping (Ng et al., 1999),

$$r_t^{\text{maze task}} = r_{\text{alive}} + \lambda_{\text{prog}} \left(d_{t-1} - d_t\right) + r_{\text{goal}} \, \mathbf{1}_{d_t < \delta}, \tag{3}$$

with $\lambda_{\text{prog}} = 16$, $r_{\text{alive}} = r_{\text{goal}} = 1$ and $\delta = 0.5\,\text{m}$. The progress term telescopes, so no closed loop of motion earns net reward; the constants were fixed once on U-Maze and transferred unchanged to every larger maze and to both control rates.

Pushing task reward. Let $\boldsymbol{c}_t$ be the block centroid and $\boldsymbol{x}_t$ the character root. Writing $d_t^{\text{blk}} = \|\boldsymbol{c}_t - \boldsymbol{g}\|$ for the block-to-goal distance and $d_t^{\text{reach}} = \|\boldsymbol{x}_t^{xy} - \boldsymbol{c}_t^{xy}\|$ for the planar agent-to-block distance, the reward is

$$r_t^{\text{push}} = r_{\text{alive}} + \lambda_{\text{prog}} \left(d_{t-1}^{\text{blk}} - d_t^{\text{blk}}\right) + \lambda_{\text{reach}} \left(d_{t-1}^{\text{reach}} - d_t^{\text{reach}}\right) + r_{\text{goal}} \, \mathbf{1}_{d_t^{\text{blk}} < \delta}. \tag{4}$$

Here $\lambda_{\text{prog}} = 16$, $\lambda_{\text{reach}} = 4$, and r_{alive}, r_{goal}, δ are as above. The reach term is a second potential difference rather than a distance penalty, so it too telescopes. Since $\lambda_{\text{reach}} < \lambda_{\text{prog}}$, moving the block toward the goal always dominates approaching the block. Without the reach term the character never finds the block; it is the one hand-weighted term the pushing tasks require, and it is consumed only by the PPO arms. The agent-to-block distance is planar because the root sits at $z \approx 0.88\,\text{m}$ while the block centroid rests at $z = 0.75\,\text{m}$, so a 3D distance would never approach zero.

Macro-step aggregation. The macro-step average $r_k^{\text{HL}} = \frac{1}{K} \sum_{i=1}^{K} r_{(k-1)K+i}^{\text{task}}$ is, by telescoping, exactly $\frac{1}{K}$ of the net progress made over the macro-step plus the alive and goal terms. One implementation detail: the divisor is the constant K (in our experiments $K = 5$) rather than the number of micro-steps actually executed, so a macro-step in which the episode terminates early is still divided by 5, with the micro-steps after termination contributing zero.

Control-effort penalty (end-to-end tier only). The end-to-end tier additionally pays $-\lambda_{\text{eff}} \sum_j (\tau_{t,j}/\bar{\tau}_j)^2$ with $\lambda_{\text{eff}} = 0.1$, where $\tau_{t,j}$ is the commanded PD drive torque of DoF j read after the step and $\bar{\tau}_j$ its effort limit. The hierarchical tier charges no effort penalty: its actions are latents rather than torques, and the effort machinery is not constructed at all in the high-level environment. This asymmetry is disclosed in Table 4. It does not handicap the end-to-end tier in practice: under exactly this reward end-to-end PPO survives essentially the full episode on the smaller layouts and reaches goals reliably there.

Policy invariance. Equation (3) is potential-based in form, taking the undiscounted difference of $\Phi(\boldsymbol{x}) = -\|\boldsymbol{x} - \boldsymbol{g}\|$, but it is not policy-invariant in the sense of Ng et al. (1999). Strict invariance requires the discounted difference $\gamma\Phi(\boldsymbol{x}_t) - \Phi(\boldsymbol{x}_{t-1})$; we use the undiscounted one, which coincides only at $\gamma = 1$, and we train at $\gamma = 0.99$. The alive and goal terms break invariance regardless, by design: they are what make survival and arrival worth anything. The telescoping property we do rely on is weaker and holds exactly, namely that the summed progress term over any trajectory depends only on its endpoints, so no closed loop of motion earns net progress reward.

Reward-design ablation. The alternative is the non-scale-invariant reward used by Wang et al. (2025), $r_t^{\text{pos}} = -d_t + r_{\text{healthy}}$, which we run as an ablation on HL PPO for U-Maze and Giant. It is not scale-invariant: because $-d_t$ charges more per step for a distant goal than a near one, the return-maximizing behavior for far goals is to end the episode early unless r_{healthy} exceeds the largest goal distance in the layout. r_{healthy} must therefore be retuned per layout, and we grant the ablation that tuning ($r_{\text{healthy}} = 5.0$ on U-Maze and 17.5 on Giant at the hierarchical tier). Equation (3) needs no such constant and transfers unchanged across all four layouts and both control rates. Measured over five seeds at the same per-environment budget as Table 1, ASE-PPO falls from 94.8% $[93.5, 96.0]$

to 73.3% $[70.7, 77.5]$ on U-Maze under the position reward, and from 37.3% $[36.6, 38.2]$ to 31.6% $[28.5, 34.2]$ on Giant. The U-Maze gap is the telling one: the position reward is granted its tuned r_{healthy} there and still loses by over twenty points, because the $-d_t$ term makes early termination competitive with walking to a distant goal. Figure 4 and Figure 5 give the learning curves.

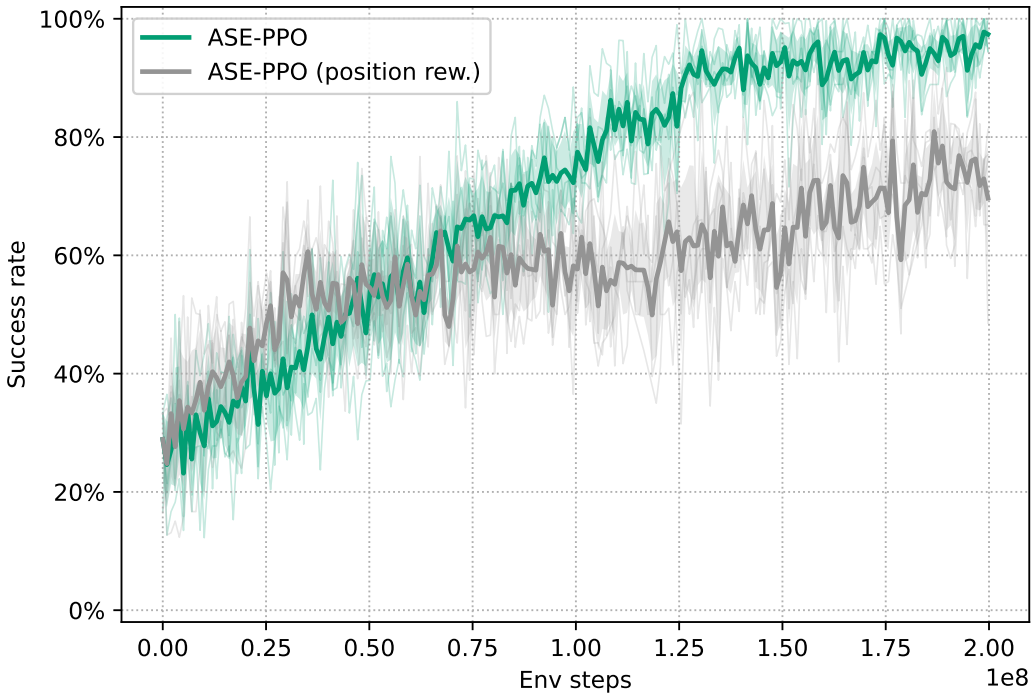


Figure 4: Reward-design ablation on U-Maze: Equation (3) against the non-scale-invariant position reward of Wang et al. (2025), five seeds each.

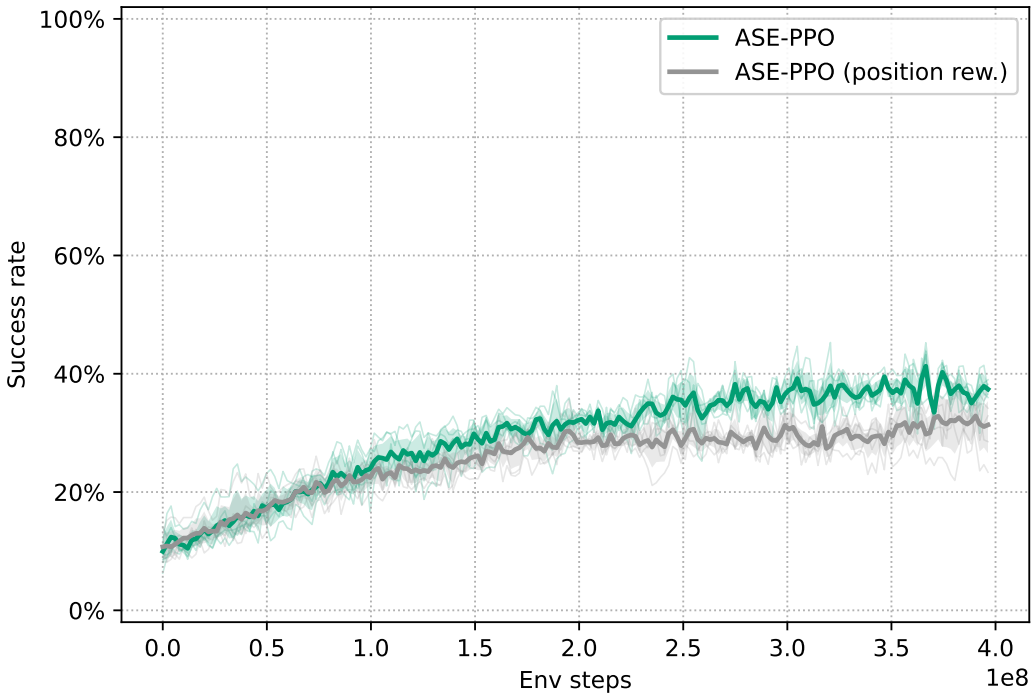


Figure 5: The same ablation on Giant, where the position reward is granted the per-layout r_{healthy} tuning that Equation (3) does not need.

A.3 Environment specifications

Maze layouts. Each layout is a grid of cells, every cell either a solid wall block or free floor, scaled by $s = 2.0\,\text{m}$. Walls are axis-aligned cuboids one full cell across, so corridor width and wall thickness are both $2.0\,\text{m}$ in every layout and only the topology and extent change with scale. Wall height is $3.0\,\text{m}$, tall enough that the character cannot see or climb over. Table 2 gives the four layouts, together with wall counts of 18, 38, 62 and 106 cells and playable interiors of 6×6, 12×12, 14×20 and $20\times28\,\text{m}$. Root and goal positions are normalized to roughly $[-1,1]$ by the interior extent $\left(\frac{(n_{\text{row}}-2)s}{2}, \frac{(n_{\text{col}}-2)s}{2}\right)$.

Table 2: HGC-NavBench environments. Extent is the outer footprint; the playable interior is smaller by one wall cell on each side. Corridor width (2 m), wall thickness (2 m), wall height (3 m) and cell scaling are identical across layouts by construction. The last two columns give the decisions each tier makes in one episode, which follow from the 30 Hz and 6 Hz control rates. The maze-embedded pushing task reuses the Large layout, drawing an ordered pair of distinct goal cells for the block's spawn and its target, so its task distribution is over cell pairs rather than single cells.

				Per episode		
Layout	**Grid**	**Extent (m)**	**Goals**	**Limit (s)**	**End-to-end**	**Hierarchical**
U-Maze	5×5	10×10	6	10	300	60
Large	8×8	16×16	25	20	600	120
Hard	9×12	18×24	45	30	900	180
Giant	12×16	24×32	85	60	1800	360
Push-cube (open arena)	—	16×16	continuous	30	900	180
Push-cube (Large)	8×8	16×16	25, paired	30	900	180

Initialization. Every layout has a single fixed start cell, used by every episode, every method and both tiers; start randomization is available in the code but disabled throughout. At the hierarchical tier the initial character state is the first frame of a walking clip with the root translated to the start cell and the heading drawn uniformly from $[0, 2\pi)$, so the prior begins from a pose it recognizes. At the end-to-end tier the character begins from a fixed neutral standing pose with fixed heading. Episodes end on timeout, on the root leaving the healthy height band $[0.60, 1.40]$ m, or on ground contact by any body other than the two feet.

Goal sampling. During training, goals are drawn uniformly with replacement from the layout's designated goal cells; the start cell is never a goal. Start randomization is disabled throughout. Early termination and the healthy root-height range (0.60 to 1.40 m) are identical across tiers. Evaluation uses a slice of environments disjoint from the training environments, stepped in the same batched simulation but never written to the replay buffer and always acting with the frozen snapshot policy's mode action. The eval slice is sized at three environments per goal cell, each pinned to a fixed goal for the whole run, and a metric row is emitted only once every goal has contributed its full quota of three episodes, so every goal is weighted equally in every reported number. What is held out is the experience, not the goals: the eval battery is the layout's full goal set, which is also the set training goals are drawn from, so the reported success rate measures goal-reaching reliability under a frozen policy and not generalization to unseen goals.

Pushing environments. The block is a $0.75 \times 0.75 \times 1.5$ m box of mass 20 kg with static and dynamic friction 0.12 and restitution 0. Its center of mass is displaced 0.47 m below its geometric center, putting it at ≈ 0.28 m above the floor at rest, which is what keeps the block sliding instead of tipping. Its resting centroid height, and the goal height, are 0.75 m. In the open-arena variant the arena is a 16×16 m walled box; the block spawns on a ring of radius $[2.0, 4.0]$ m about the character's start and the target on a ring of radius $[2.5, 6.0]$ m about the block, both directions uniform, with the block clamped to ± 6 m and the goal to ± 7 m. In the maze-embedded variant the block spawn and target are two distinct goal cells of the Large layout, both distinct from the fixed agent start. Because the block can be pushed but not pulled, the block's own path must stay within free cells, and a block driven into a corner or flush against a wall is very difficult to recover from. Both variants evaluate against a frozen battery of 100 instances drawn once offline and pinned one per evaluation environment, so the evaluation distribution is identical across methods, seeds and checkpoints. The open-arena battery holds absolute $(\text{block}, \text{goal})$ positions drawn with seed 12345 from the same rings, with a minimum block-to-goal distance of 2.51 m so that no instance is solved at spawn. The maze-embedded battery holds 100 distinct (block cell, target cell) index pairs drawn with seed 6789 over the Large layout's 25 goal cells, with block $\neq$ target, every one of the 25 cells appearing as a block spawn and no cell reused more than six times. Because these tasks have no per-cell battery, evaluation balances episodes across evaluation environments rather than across goal cells, unlike the mazes.

Why the maze-embedded variant is hard. It is considerably harder than navigation in the same layout for two reasons beyond horizon. First, goal reaching is indirect, so exploration must chain reaching the block and then moving it before any relabeled goal is informative. Second, the block can be pushed but not pulled, so it wedges into corners and against walls, and recovering it requires striking it at close to the right angle with enough force to open a gap; unlike Sokoban these states are not formally irreversible, but they are hard to escape.

Observations and actions. Table 3 gives the observation layouts. The 140-dimensional character block is root height, root rotation, root linear and angular velocity, joint rotations, joint velocities and five key-body positions, expressed in the heading-invariant local frame. Only the position components are coordinate-normalized; the character block and the block velocity pass through raw. The end-to-end action is 28 joint PD position targets, clipped to per-DoF bounds derived from the joint limits. The high-level action is a 64-dimensional latent, projected onto the unit hypersphere before it is handed to the frozen prior.

Simulation. All environments run in Isaac Lab (Mittal et al., 2023; NVIDIA, 2025) at a physics rate of 120 Hz with a control rate of 30 Hz, i.e. four physics sub-steps per control step. The hierarchical tier issues one latent every $K = 5$ control steps, giving 6 Hz high-level control. Every run

Table 3: Observation layouts. Components in order; subscripts give dimensions.

Tier / task	Components	Total
End-to-end, maze	$\boldsymbol{x}^3_{\text{root}}$, $\boldsymbol{q}^4_{\text{root}}$, char^{140}, $\boldsymbol{g}^3$	150
Hierarchical, maze	$\boldsymbol{x}^3_{\text{root}}$, $\boldsymbol{q}^4_{\text{root}}$, $\boldsymbol{g}^3$	10
Hierarchical, maze, proprioception ablation	$\boldsymbol{x}^3_{\text{root}}$, $\boldsymbol{q}^4_{\text{root}}$, char^{140}, $\boldsymbol{g}^3$	150
End-to-end, pushing	$\boldsymbol{c}^3$, $\dot{\boldsymbol{c}}^3$, $\boldsymbol{x}^3_{\text{root}}$, $\boldsymbol{q}^4_{\text{root}}$, char^{140}, $\boldsymbol{g}^3$	156
Hierarchical, pushing	$\boldsymbol{c}^3$, $\dot{\boldsymbol{c}}^3$, $\boldsymbol{x}^3_{\text{root}}$, $\boldsymbol{q}^4_{\text{root}}$, $\boldsymbol{g}^3$	16

uses 512 training environments, with the evaluation slice appended on top rather than taken out of them: 18, 75, 135 and 255 environments on U-Maze, Large, Hard and Giant (three per goal cell) and 100 in each pushing variant, for totals of 530 to 767 parallel environments.

A.4 Hyperparameters and network specifications

Contrastive and off-policy architectures. All contrastive networks use the residual architecture of Wang et al. (2025). A residual block is four $\text{Linear} \to \text{LayerNorm} \to \text{SiLU}$ units wrapped in an identity skip; depth counts linear layers, so a depth of 32 is eight blocks. Every network is width 512. The state–action encoder ϕ and goal encoder ψ each consist of an input stem, the residual trunk, and a final linear projection to a fixed 64-dimensional representation. The actor shares the trunk and emits a mean and a log standard deviation, the latter squashed by a $\tanh$ into $[-5, 2]$. ASE-SAC+HER reuses the identical trunk and differs only in its head, a scalar Q rather than a 64-dimensional embedding, instantiated twice with Polyak-tracked targets, with the temperature autotuned to the target entropy of Table 5 (Haarnoja et al., 2018b). All linear layers use LeCun-uniform initialization with zero bias, except the actor's mean and log-std heads, initialized uniformly in $[-0.005, 0.005]$.

With 512 training environments, 62 rollout steps per iteration against 800 gradient steps is an update-to-data ratio of 1:39.7, identical for all three off-policy agents by construction. Table 5 lists the off-policy settings and Table 6 the PPO settings. The sampling window is $300/600/900/1800$ steps for end-to-end CRL and $60/120/180/360$ macro-steps for the hierarchical agents on U-Maze, Large, Hard and Giant respectively, and 900 / 180 on both pushing tasks.

Table 7 gives the training budgets.

Agent selection and coverage. ASE-SAC+HER is matched to ASE-CRL on rollout structure, replay buffer, actor parameterization, entropy temperature, network width, batch size, learning rates, discount and update-to-data ratio. It differs in three respects, each the standard configuration of its own objective: it learns twin TD-bootstrapped Q functions from the sparse goal reward rather than the contrastive critic of Equation (1); it relabels uniformly over future states for 80% of each batch where CRL relabels geometrically throughout; and it runs at depth 4 rather than 32. That depth follows the JaxGCRL SAC+HER configuration (Bortkiewicz et al., 2025) together with the finding of Wang et al. (2025) that the depth scaling which benefits the contrastive critic does not carry over to SAC in the goal-conditioned setting. We report it as a per-algorithm configuration choice and not as a claim that SAC cannot use depth: Nauman et al. (2024) and Lee et al. (2025) both obtain substantial gains from scaling normalized residual critics of the kind our shared trunk instantiates, in the reward-driven setting. What their results and Wang et al. (2025) together suggest is that the depth an off-policy actor-critic can exploit depends on the objective and on the learning signal it is trained against, which is a property of the comparison we are making rather than a nuisance variable. We therefore give each arm the depth its own literature supports rather than matching the two at a value that suits neither. Each arm accordingly uses the relabeling distribution its objective calls for and the depth that suits it. We also ran ASE-SAC+HER at depth 32 on Giant to check this directly, and it does not change the result (Section A.7).

Contrastive depth was chosen from preliminary runs at 32 and 64 at both tiers. The hierarchical agent did not benefit from the extra depth and ran slower, so we report it at 32; the end-to-end agent did benefit and is reported at 64, the depth Wang et al. (2025) identify as critical for humanoid

Table 4: Differences between the end-to-end and hierarchical tiers. All are consequences of the motor abstraction rather than independently manipulable factors, with the exception of contrastive depth, which is set in each case to the value that favors that tier.

	End-to-end	**Hierarchical**
Observation	full state, incl. 140D proprioception	10D root pose and goal
Action	28 joint PD targets	64D ASE latent
Control rate	30 Hz	6 Hz ($K = 5$)
Decisions, Giant	1800	360
Effort penalty	yes, $\lambda_{\text{eff}} = 0.1$	none
Initialization	fixed standing pose	walk-clip first frame, random heading
Motion data	none	CMU and SMPL/AMASS MoCap via the frozen prior
Contrastive depth	64 layers	32 layers

Table 5: Hyperparameters for the off-policy agents. The sampling window is the trajectory length used for geometric relabeling (CRL) and future sampling (HER), and is the episode length in the units of that tier. Values differing between agents are in the body; everything else is shared.

	End-to-end CRL	**ASE-CRL**	**ASE-SAC+HER**
Actor / critic depth (linear layers)	64	32	4
Network width	512	512	512
Representation dimension	64	64	— (scalar Q)
Critic	$-\|\phi - \psi\|_2$	$-\|\phi - \psi\|_2$	twin Q, target Polyak $\tau = 0.005$
Relabeling	geometric, all	geometric, all	uniform future, 80%
Batch size	512	512	512
Actor / critic / α learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}
Discount γ	0.99	0.99	0.99
Target entropy	$-0.5\,\|\mathcal{A}\|$	$-0.5\,\|\mathcal{A}\|$	$-0.5\,\|\mathcal{A}\|$
Replay capacity (per env)	10^4	10^4	10^4
Warmup before learning	—	—	10^3
Rollout steps per iteration	62	62	62
Gradient steps per iteration	800	800	800
LogSumExp penalty	0.1	0.1	—
Observation normalization	off	off	off
Gradient clipping	off	off	off

mazes. Neither end-to-end agent is run on the pushing tasks: at 64 layers and the long episodes those environments require, five seeds of end-to-end CRL are among the most GPU-expensive runs in the suite, roughly six times its hierarchical counterpart per environment step (Table 11), and the capacity was better spent on the hierarchical arms where the comparison of interest lies.

Low-level ASE prior. The prior is trained once with a 64-dimensional latent, actor, critic and discriminator of shape $[1024, 1024, 512]$ and an encoder of shape $[1024, 512]$, at learning rates 2×10^{-5} for the actor and 5×10^{-5} for the critic, discriminator and encoder. The discriminator uses a replay buffer of 2×10^5 transitions, logit regularization 0.01, gradient penalty 5 and reward scale 2; the diversity term is weighted 0.01. The style and diversity rewards are weighted 0.5 each and the task reward weight is zero, so no task signal enters low-level training. Latents are resampled at intervals drawn uniformly from $[0, 5]$ s. Training ran with 2048 environments and 10 s episodes.

Worth noting is that this leaves the prior driven well outside its typical pretraining regime downstream. During pretraining a latent is held for 2.5 s on average; at $K = 5$ and 30 Hz the high-level policy switches every 0.167 s, at the very bottom of the training distribution. The prior nonetheless

Table 6: PPO hyperparameters. Advantages use generalized advantage estimation (Schulman et al., 2016). Values differ between the two tiers in several rows, most notably network shape, learning rate and schedule. The learning rate, entropy bonus and action std of ASE-PPO are identical to those used by Peng et al. (2022).

	End-to-end PPO	**ASE-PPO**
Actor / critic network	$[1024, 1024, 1024, 512]$	$[1024, 1024, 512]$
Learning rate	3×10^{-4}, adaptive (KL 0.008)	2×10^{-5}, constant
Clip ratio	0.2	0.2
GAE λ / discount	0.95 / 0.99	0.95 / 0.99
Rollout steps per iteration	256	256
Epochs $\times$ minibatch	6×32	6×32
Entropy bonus	0.005	0
Value clipping	on, 0.2	off
Action std	learned, init 0.1	fixed, 0.1
Advantage norm. clip	4.0	4.0
Gradient clipping	1.0	off

Algorithm 1 ASE-CRL

Require: frozen low-level policy π_{LL} with mean action $\boldsymbol{\mu}_{\text{LL}}$, macro-step K, episode length T (control steps), discount γ, achieved-goal projection φ

initialize encoders ϕ, ψ, high-level actor π_{HL}, replay buffer $\mathcal{D}$
for each iteration **do**
  for each parallel environment **do** ▷ rollout
    sample task goal $\boldsymbol{g}$; reset
    for $k = 1, \ldots, T/K$ **do**
      observe $\boldsymbol{s}_k$; sample $\boldsymbol{z}_k \sim \pi_{\text{HL}}(\cdot \mid \boldsymbol{s}_k, \boldsymbol{g})$
      for $i = 1, \ldots, K$ **do**
        observe proprioceptive state $\boldsymbol{o}$
        $\boldsymbol{a} \leftarrow \boldsymbol{\mu}_{\text{LL}}(\boldsymbol{o}, \boldsymbol{z}_k)$
        step the simulator with $\boldsymbol{a}$
      end for
      store $(\boldsymbol{s}_k, \boldsymbol{z}_k)$ in the current trajectory
    end for
    append trajectory to $\mathcal{D}$
  end for
  for each gradient step **do** ▷ learning
    sample a batch of transitions $(\boldsymbol{s}_k, \boldsymbol{z}_k)$ from $\mathcal{D}$
    for each, sample $\Delta \sim \text{Geom}(1-\gamma)$, truncated to the remaining episode, and set $\boldsymbol{g}_+ \leftarrow \varphi(\boldsymbol{s}_{k+\Delta})$
    collect negatives for each transition: the $\boldsymbol{g}_+$ of the other transitions in the batch
    update ϕ, ψ by minimizing Equation (1) with the LogSumExp penalty of Table 5
    update π_{HL} by maximizing $C(\boldsymbol{s}_k, \boldsymbol{z}, \boldsymbol{g}_+) - \alpha \log \pi_{\text{HL}}(\boldsymbol{z} \mid \boldsymbol{s}_k, \boldsymbol{g}_+)$, $\boldsymbol{z} \sim \pi_{\text{HL}}(\cdot \mid \boldsymbol{s}_k, \boldsymbol{g}_+)$, reparameterized
    update the entropy temperature α by descending $-\alpha\,(\log \pi_{\text{HL}}(\boldsymbol{z} \mid \boldsymbol{s}_k, \boldsymbol{g}_+) + \bar{\mathcal{H}})$, with target entropy $\bar{\mathcal{H}} = -0.5|\mathcal{Z}|$
  end for
end for

produces stable locomotion under that command rate, which we did not tune for: K was chosen to shorten the high-level horizon, not to match the prior's training statistics. Whether a larger K trades horizon length against fidelity to the prior is an obvious question we do not answer. The motion dataset is 62 locomotion clips (walking, running, and turning left and right). The single resulting checkpoint is loaded frozen by every hierarchical run in the paper, on all four mazes and both pushing tasks, without retraining or fine-tuning.

Table 7: Training budgets. All five methods are matched on environment steps within each task. Because the high-level agent acts once per $K = 5$ low-level steps, its own step budget is one-fifth of the environment-step budget.

Task	**Environment steps**	**High-level agent steps**
U-Maze	200 M	40 M
Large	250 M	50 M
Hard	300 M	60 M
Giant	400 M	80 M
Push-cube (open arena)	750 M	150 M
Push-cube (Large)	4000 M	800 M

Every reported configuration is run with five seeds, 0 to 4, passed explicitly so that runs are reproducible; the code refuses to silently use a time-based seed.

A.5 Evaluation protocol and metric definitions

All evaluation quantities are computed on the evaluation slice, which acts with the frozen snapshot policy's mode action and is never written to the replay buffer. Every quantity is accumulated per micro-step, latched at episode end, and pooled over episodes with each goal weighted equally.

- **Success rate.** The any-time planar criterion of Section 5, latched as a binary flag per episode. It measures whether the goal was ever attained, not where the character finished.
- **Success rate (2 m).** Identical, at a $2.0\,\mathrm{m}$ radius. Reported as a coarser diagnostic that separates near-misses from failures to approach at all.
- **Mean distance to goal.** The three-dimensional distance from the achieved quantity to the goal, averaged over the control steps the episode actually survived. Because the divisor is steps survived, this quantity is not comparable across methods with very different termination behavior without also reading the episode length.
- **Return and episode length.** The undiscounted sum of per-step rewards, and the number of agent steps. Both are in the units of their own tier: a hierarchical return sums macro-step rewards and a hierarchical episode length counts macro-steps, so neither is directly comparable across tiers. Success rate and distance to goal are comparable across tiers, and are what we report.

Environment steps are reported on a common low-level axis for both tiers, so learning curves are budget-matched by construction, and count the 512 training environments only: the evaluation slice steps in the same batched simulation but is not charged to the budget.

A.6 Proprioception ablation

The high-level observation supplies root position, root orientation and the goal, and withholds the 140-dimensional proprioceptive character state that the frozen prior consumes internally (Table 3). A natural objection to the hierarchical results is that this compact interface, rather than the objective, drives the difference between the arms. We test that directly by splicing the proprioceptive state back in, giving a 150-dimensional high-level observation, and retraining ASE-PPO and ASE-CRL on Giant for five seeds each under an identical budget. The agent configurations are otherwise unchanged, so the networks simply infer the larger input dimension; per-run compute for these four configurations is included in Table 11 and the learning curves in Figure 6.

The result is asymmetric, and both baseline arms are read at the same per-environment budget as Table 1. ASE-CRL is essentially unaffected: 97.4% $[97.0, 97.7]$ with the 10-dimensional observation against 97.9% $[97.8, 98.0]$ with proprioception spliced back in, a difference of half a point. ASE-PPO more than halves, from 37.3% $[36.6, 38.2]$ to 14.3% $[11.4, 17.3]$. The compact interface is therefore not what separates the two arms; if anything it is load-bearing for the reward-driven one, which appears unable to exploit the extra 140 dimensions and is hurt by having to ignore them.

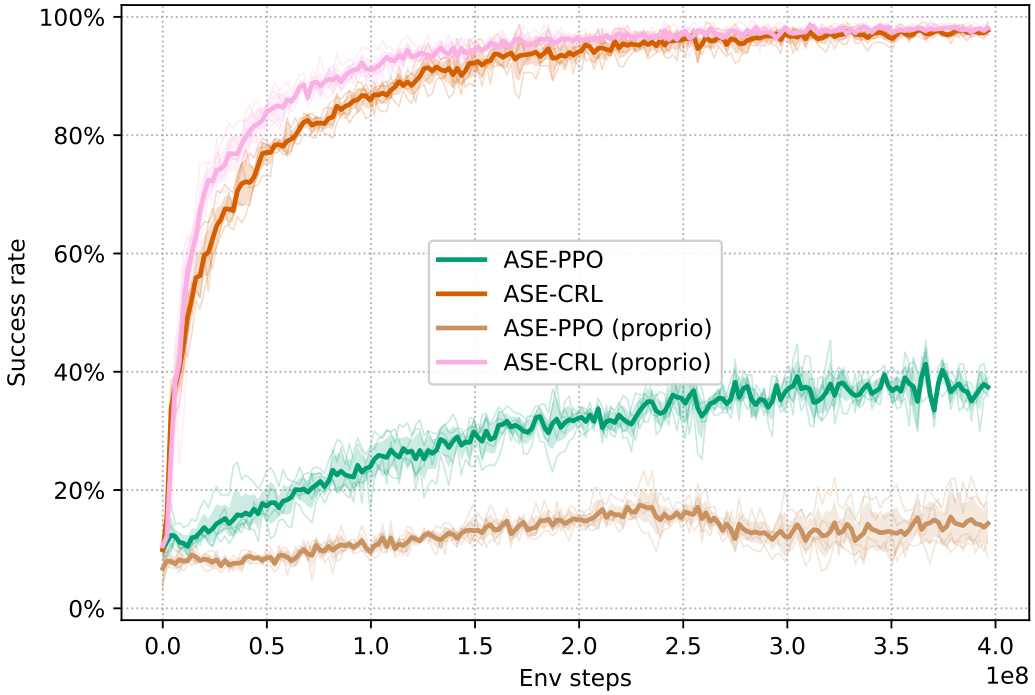


Figure 6: Proprioception ablation on Giant. Mean over five seeds with a 95% percentile-bootstrap confidence interval and the individual seeds drawn underneath. ASE-CRL and ASE-CRL (proprio) lie on top of one another, while ASE-PPO separates from ASE-PPO (proprio) by more than twenty points: the contrastive agent is essentially unaffected by the extra 140 dimensions and the reward-driven one is hurt by them.

A.7 Depth control for ASE-SAC+HER

ASE-CRL and ASE-SAC+HER differ in the critic, in the relabeling distribution and in depth, 32 against 4 (Section A.4). Depth is the one of the three that is not intrinsic to either objective, so the natural objection to the headline hierarchical comparison is that the gap is a capacity gap. We test it directly on Giant, the layout where the gap is largest, by raising ASE-SAC+HER to ASE-CRL's depth of 32 and changing nothing else, and running five seeds at the same per-environment budget as Table 1.

The result is flat. Depth 32 reaches 83.1% $[77.9, 88.5]$ against 82.1% $[77.9, 86.4]$ at depth 4, a difference of one point with almost entirely overlapping intervals, and the learning curves are indistinguishable (Figure 7). The fifteen-point gap to ASE-CRL on this layout is therefore not explained by the depth difference, and matching the two arms at 32 would change neither the ordering nor its magnitude.

Table 8: Depth control on Giant. Success rate (%) at the end of training over five seeds, summarized exactly as in Table 10. ASE-CRL runs at depth 32 and reaches 97.4 $[97.0, 97.7]$ on this layout.

Environment	Method	Mean	SD	95% CI
Giant	ASE-SAC+HER	82.1	5.6	[77.9, 86.4]
	ASE-SAC+HER (depth 32)	83.1	6.7	[77.9, 88.5]

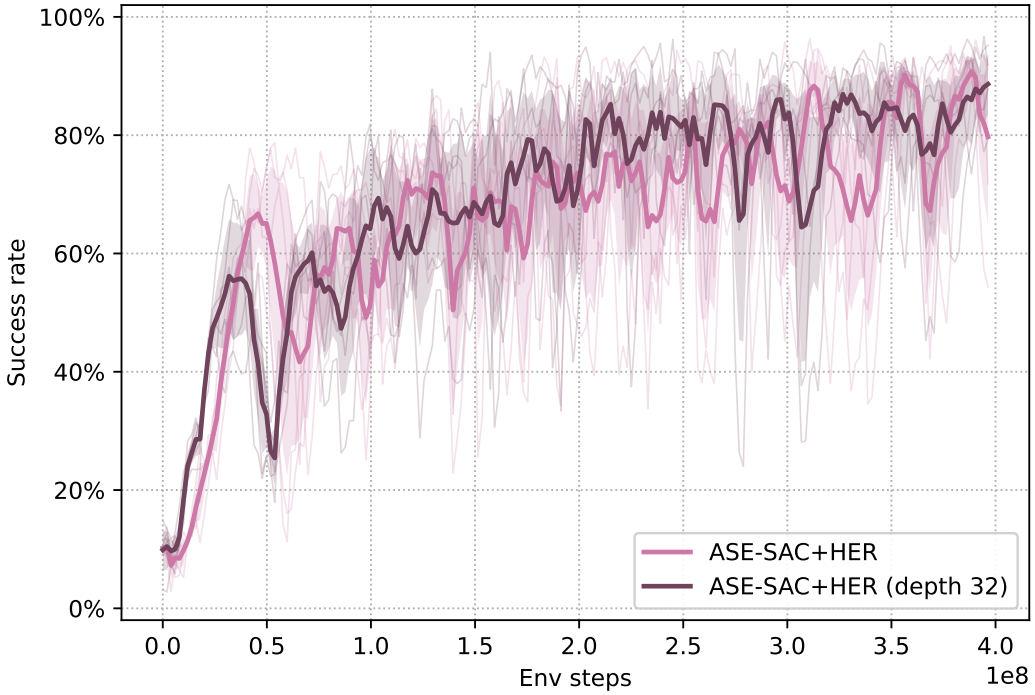


Figure 7: Depth control on Giant: ASE-SAC+HER at its own depth of 4 against the same agent at ASE-CRL's depth of 32, five seeds each. Mean over seeds with a 95% percentile-bootstrap confidence interval and the individual seeds drawn underneath.

A.8 The U-Maze time budget binds for ASE-SAC+HER

ASE-SAC+HER is the one hierarchical arm that does worse on U-Maze (72.7%) than on the three larger layouts (94.7, 93.8 and 82.1%; Table 1), which is the opposite of the ordering every other hierarchical method follows. Watching its rollouts suggested a reason: the agent reaches the far goals but takes longer to do so, and U-Maze has the shortest episode budget of the suite, 10 s. We test that explanation directly by re-evaluating the same checkpoints with the budget extended.

Protocol. For each of the three hierarchical arms and each of the five seeds we take the final checkpoint of the run reported in Table 1 and run it on U-Maze with the episode length raised from 10 to 30 s, changing nothing else. Each run is 120 parallel episodes, 20 per goal cell, from the fixed start cell with the heading drawn uniformly as in training, under the frozen policy's mode action, so each method contributes 600 episodes. The high-level observation carries no time input (Table 3) and the success criterion is latched on first arrival without ending the episode (Section A.5), so the extension is purely an evaluation-time change: no policy can behave differently in its first 10 s, and success under any budget $t \leq 30$ s is read off the same episodes as the fraction whose first arrival time is at most t. The evaluation costs about a minute per checkpoint on one laptop GPU and is not counted in Table 11. One run per checkpoint therefore yields the full curve of success against time budget, and its value at $t = 10$ s is the headline metric re-measured on the final weights of each run. It differs from Table 1 in what is averaged: the table summarizes each seed by the mean of its evaluation snapshots over the last 10% of training, 18 episodes per snapshot, whereas here each seed contributes 120 episodes from its final checkpoint alone. The two agree within seed noise: 95.7 ± 1.5, 99.8 ± 0.4 and 80.8 ± 6.7% for ASE-PPO, ASE-CRL and ASE-SAC+HER here, against 94.8, 99.8 and 72.7% in Table 1.

Result. The gap is a matter of time, not of reachability (Figure 8). Under the 10 s budget ASE-SAC+HER reaches 80.8 ± 6.7% of its goals; under 20 s it reaches 97.3 ± 1.2%, against 97.5 ± 1.0% for ASE-PPO and 99.8 ± 0.4% for ASE-CRL; by 30 s the three plateau at 98.3, 98.2 and 99.8%

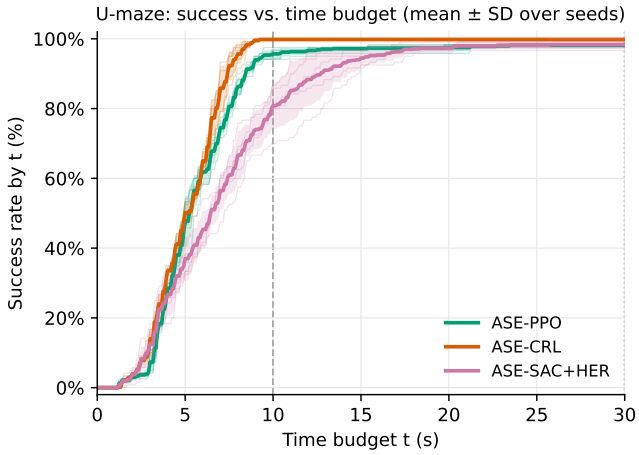


Figure 8: Success on U-Maze as a function of the time budget, from the same final checkpoints as Table 1 evaluated under a 30 s limit. A point (t, p) means a fraction p of episodes first reached the 0.5 m goal radius within t seconds. Mean over five seeds with a ± 1 standard deviation band and the individual seeds drawn underneath; the dashed line is the 10 s budget of Table 1. Reading the curves at that line reproduces the headline ordering; by 20 s ASE-SAC+HER has closed to within a point of ASE-PPO.

respectively. Its curve tracks the others over the first few seconds and then flattens rather than stopping: the episodes that the 10 s budget counts as failures are, almost without exception, episodes that arrive between 10 and 20 s. Falls are not the mechanism, since every arm survives the 30 s episodes at a similar rate (32, 15 and 29 falls in 600 episodes for ASE-PPO, ASE-CRL and ASE-SAC+HER).

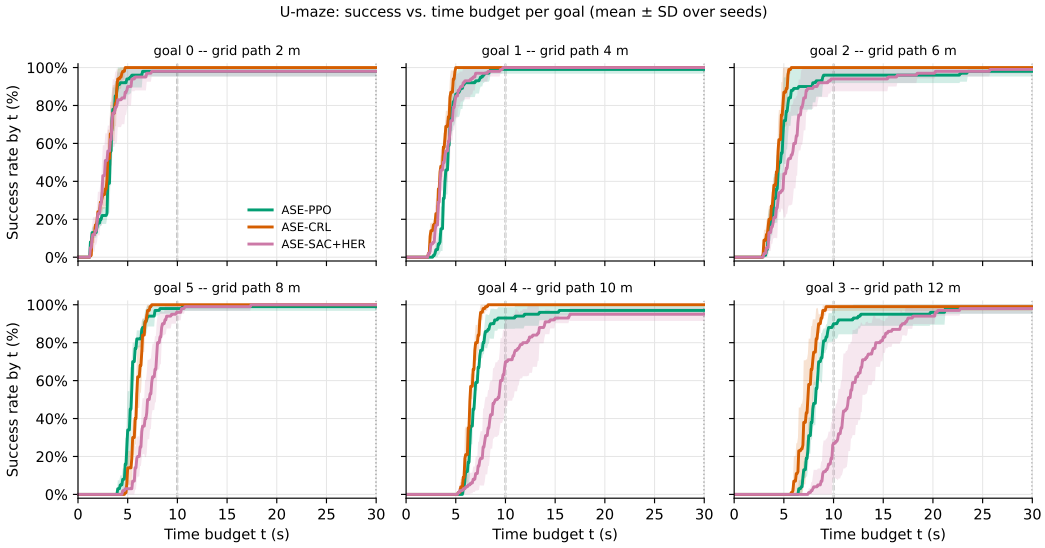


Figure 9: The curves of Figure 8 split by goal cell, ordered by grid path length from the start cell, the Manhattan distance between cell centres through free cells at 2 m per cell (2 to 12 m; the far goals require two turns). This is an upper bound on the metric shortest route, by up to 30% on the corner goals, since the character can cut corners; it is the same yardstick for every arm. Mean over five seeds with a ± 1 standard deviation band. The near goals are a control that every arm passes; the whole ASE-SAC+HER deficit at 10 s sits in the two farthest cells.

Splitting by goal makes the localization explicit (Figure 9). On the four goals with grid path lengths of 2 to 8 m all three arms are at or above 94% within 10 s. On the 10 m goal ASE-SAC+HER reaches $70 \pm 24\%$ at 10 s and $95 \pm 4\%$ at 20 s; on the 12 m goal, $27 \pm 18\%$ at 10 s and $94 \pm 4\%$ at 20 s. ASE-PPO and ASE-CRL are at 90 and 99% on that same 12 m goal within 10 s, so the budget is not slack for them either: ASE-PPO gains six points on it from the extension.

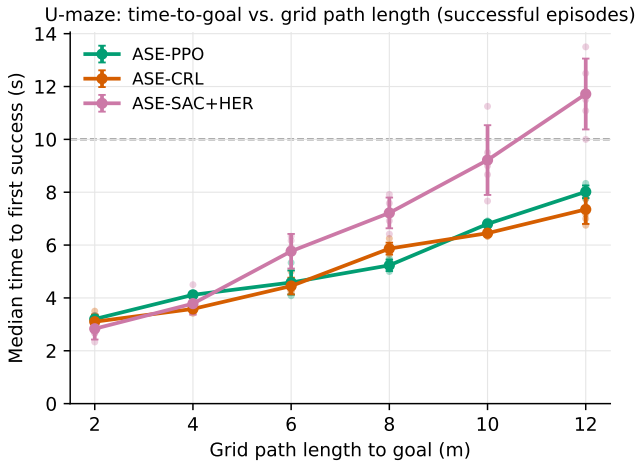


Figure 10: Median time to first arrival on U-Maze against the grid path length of the goal (Figure 9), successful episodes only. Small markers are per-seed medians, large markers and error bars the mean and standard deviation over seeds; the dashed line is the 10 s budget. The three arms arrive at the nearest goal at the same time and separate with distance: ASE-SAC+HER's median arrival at the 12 m goal falls past the budget.

Figure 10 shows why those two cells and no others. Median arrival time is close to linear in grid path length for every arm, and the three coincide at the 2 m goal (about 3 s, the start-up cost of turning from a random heading and getting under way). They then separate in slope: 0.47 and 0.45 s per grid metre for ASE-PPO and ASE-CRL against 0.89 s per grid metre for ASE-SAC+HER, so ASE-SAC+HER moves along the route at about half the effective speed of the other two arms. Because the grid length overstates the route that a corner-cutting character actually walks, we report the slopes as a ratio rather than as absolute speeds. At the 12 m goal the median arrival is 7.3 s for ASE-CRL, 8.2 s for ASE-PPO and 11.5 s for ASE-SAC+HER (interquartile range 9.8 to 13.5 s), so under a 10 s budget more than half of ASE-SAC+HER's arrivals at that goal are counted as failures. The lower effective speed comes with longer paths: the distance the root travels before first arrival is $1.15\times$ the grid path length for ASE-SAC+HER against $1.04\times$ and $1.00\times$ for ASE-PPO and ASE-CRL (ratios to the grid length rather than to the metric shortest route, so 1.00 does not mean the geodesic was walked), consistent with the detours and wall contact we observe qualitatively in its rollouts, though we do not measure contact directly.

Why U-Maze and not the larger layouts. The budgets in Table 2 were set in proportion to extent, and by the crude measure of the farthest goal's grid path length over the budget they are similar across the suite: 1.2 grid metres per second on U-Maze against 1.1, 1.0 and 1.0 on Large, Hard and Giant. What differs is the room for a per-episode overhead. Roughly 3 s of every episode is spent getting under way, and any further detour is paid in seconds rather than in a fraction of the layout, so a fixed cost of a few seconds is a third of a 10 s budget and a twentieth of a 60 s one. A slower but reliable agent is therefore penalized most on the smallest layout, which is the pattern Table 1 shows.

What this does and does not change. The headline table keeps the 10 s budget, which every method was trained and evaluated under; this appendix explains the U-Maze entry rather than replacing it. Read with the budget relaxed, ASE-SAC+HER solves U-Maze about as reliably as ASE-PPO and remains behind ASE-CRL on every layout, so the ordering of the two goal-conditioned arms is unchanged. What the comparison adds is that the U-Maze anomaly is a speed difference rather

than an exploration failure, which is a different kind of shortcoming from the ones ASE-SAC+HER shows on Giant (Section A.7) and on pushing (Section 5.2).

A.9 Control frequency alone does not explain the hierarchical gap

The hierarchical tier differs from the end-to-end tier in two ways that could each shorten the learning problem: it acts in a 64-dimensional skill space, and it decides five times less often, at 6 Hz rather than 30 Hz. The second is a horizon reduction, and this paper's own premise is that contrastive RL is horizon-sensitive, so the gap cannot be attributed to the skill space while the decision rate also differs. We therefore give the end-to-end contrastive agent the hierarchical decision rate on its own.

The ablation is the Giant end-to-end CRL configuration with one change, an action repeat of five: the policy emits 28 joint position targets once every five control steps and those targets are held for the intervening steps. The PD loop still runs at 30 Hz and the physics at 120 Hz, so the character is controlled exactly as before and only the agent's decision rate changes, from 1800 decisions per episode to 360, which is what the hierarchical arm gets. Everything else, including architecture, optimizer and step budget, is unchanged. Five seeds, read at the same shared budget as Table 1.

Matching the decision rate does not help. End-to-end CRL reaches 4.6% $[1.6, 8.4]$ at 6 Hz against 9.7% $[1.5, 20.5]$ at 30 Hz, and ASE-CRL reaches 97.4% $[97.0, 97.7]$ at that same 6 Hz rate. The two end-to-end intervals overlap, so we do not read the low-frequency arm as significantly worse; what the comparison rules out is the hypothesis that the horizon reduction is what the motor prior supplies, since granting that reduction on its own moves the end-to-end agent no closer to the hierarchical one.

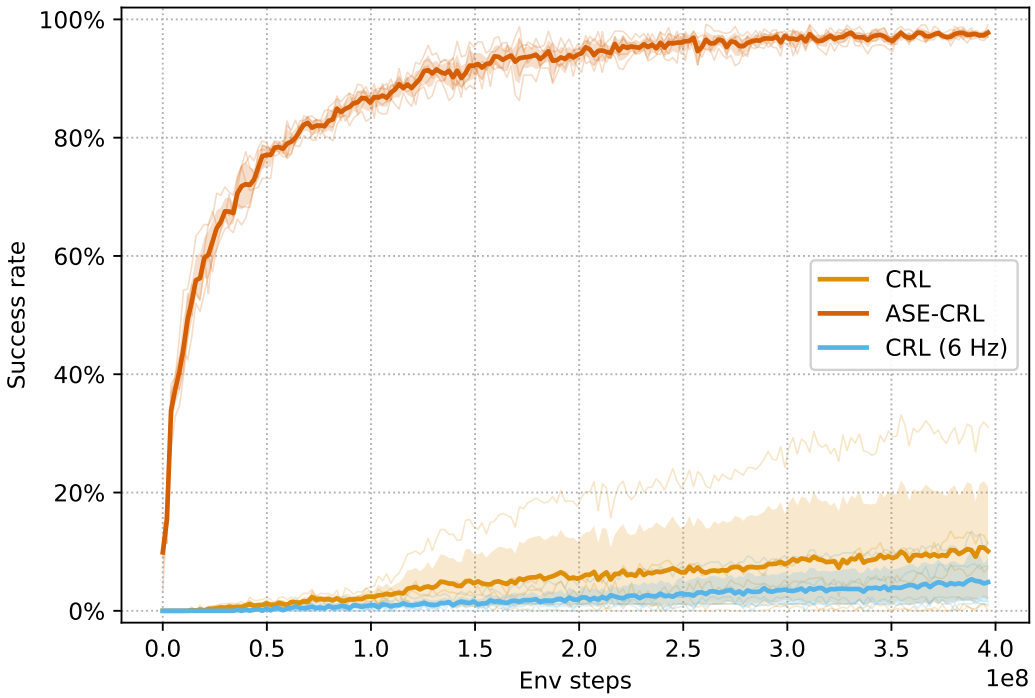


Figure 11: Control-frequency ablation on Giant. End-to-end CRL at the hierarchical tier's 6 Hz decision rate, against end-to-end CRL at 30 Hz and ASE-CRL, which decides at 6 Hz through the frozen prior. Mean over five seeds with a 95% percentile-bootstrap confidence interval and the individual seeds drawn underneath.

Episode survival says why. Episodes end when the root leaves the healthy height band or a non-foot body touches the ground (Section A.3), so how long an episode lasts measures how long the character stays on its feet. Lowering the decision rate buys no stability: over the same evaluation window the 6 Hz arm's episodes last 49.5% $[20, 101]$ as long as the 30 Hz arm's on the same layout and budget. The interval reaches parity, so we do not claim the reduction is significant at five seeds;

what it excludes is any gain, and the point estimate runs the other way. The reason it cannot help is mechanical: a joint target that suits the start of a 167 ms interval need not suit the end of it, and the policy has no opportunity to correct within the interval. Lowering the decision rate of a controller that cannot yet stand takes away the corrections that were keeping it upright.

This is the sense in which the prior is a precondition for temporal abstraction rather than an independent accelerator. Deciding at 6 Hz is only useful if each decision is safe to commit to for the whole interval, and that is exactly what the frozen prior provides: a latent held for five control steps decodes to a stable gait, so the same fivefold horizon reduction that does not help the end-to-end agent, and may hurt it, costs the hierarchical one nothing. Temporal abstraction and a stable low-level controller are not two independent benefits that happen to arrive together; the first is only available because of the second.

Two caveats. This is one environment, chosen because it is where the end-to-end and hierarchical arms are furthest apart, and a rate sweep rather than a single alternative rate would say more about where the trade-off turns. The 6 Hz arm also keeps the depth and hyperparameters tuned for the 30 Hz agent, so it is a control on the decision rate and not a tuned low-frequency baseline.

A.10 Latent smoothness

Peng et al. (2022) note that a frozen low-level policy does not by itself guarantee natural motion downstream: the high-level policy can still command latents that change sharply from one decision to the next, so the character keeps switching skill and moves jerkily. Since our three hierarchical arms drive an identical prior and differ only in the high-level learner, the step-to-step behavior of their latent sequences is a property of the learner alone, and one worth reporting alongside success rate.

We measure it geometrically. Latents lie on the unit hypersphere, so we take the angle $\theta_k = \arccos\langle \boldsymbol{z}_k, \boldsymbol{z}_{k+1} \rangle$ between consecutive commanded latents and average it over an episode. Two reference regimes bound the scale. Latents drawn independently each macro-step are near-orthogonal in 64 dimensions, giving $\theta \approx 90°$; the prior's own pre-training distribution, which holds a latent for a dwell time drawn from $U(0, 5)$ s at 6 Hz (a 6.7% per-step switch rate), gives $\theta \approx 6°$. The first is the jitter regime ASE warns about and the second is the regime the prior was trained under. Statistics are pooled over five seeds per configuration and the full evaluation battery.

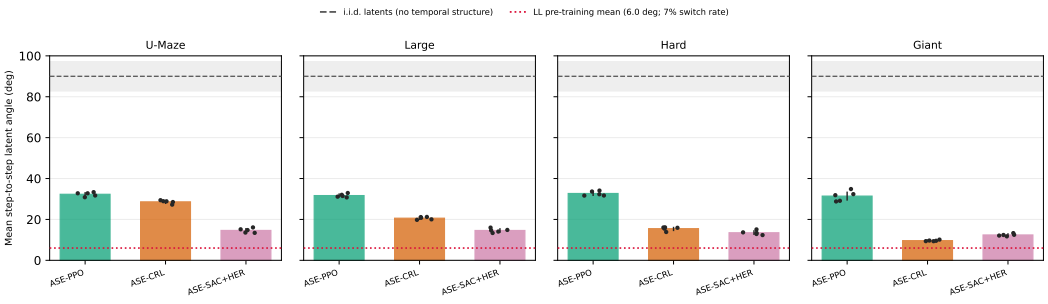


Figure 12: Mean step-to-step latent angle per layout, five seeds per bar with the individual seeds overlaid. The dashed line is the i.i.d. regime (90°) and the dotted line the prior's pre-training regime (6°). Every arm sits far below the i.i.d. null, so all three learn temporally structured latents rather than resampling skills at random.

ASE-PPO is the least smooth arm on every layout, at 31 to 33°, and it is flat across scale: 5.2 to 5.4× the pre-training angle whether the maze is U-Maze or Giant. Both goal-conditioned arms are smoother than it everywhere, by better than a factor of two on the two largest layouts. That is the same grouping the success rates produce, and it is not something the sparse goal indicator obviously buys. No arm receives any penalty on latent change: the end-to-end tier's control-effort term is dropped at the hierarchical tier for all three arms, since an action-smoothness penalty is not physically meaningful on a 64-dimensional latent the way it is on joint targets (Section A.2). The smoother latent sequences are therefore a by-product of what these objectives make the high-level

policy represent, not of an explicit smoothness term given to some arms and withheld from others. It also means the classic ASE recipe, a reward-driven high-level policy over a frozen prior, is the arm most exposed to the failure mode Peng et al. (2022) raise.

Between the two goal-conditioned arms the levels are comparable, with the exception of the smallest maze layout, and the trend with scale separates them more than the level does. ASE-SAC+HER is smoother on the Large and Hard layouts and ASE-CRL is clearly smoother on Giant, at $1.6\times$ pre-training against $2.1\times$. ASE-CRL's latent trajectory becomes monotonically smoother as the layout grows, from $4.8\times$ on U-Maze through $3.4\times$ and $2.6\times$ to $1.6\times$ on Giant, while ASE-SAC+HER barely moves ($2.4\times$ to $2.1\times$) and ASE-PPO not at all.

Table 9: Latent smoothness by layout and method. $\bar{\theta}$ is the mean step-to-step latent angle with a 95% percentile-bootstrap confidence interval over five seeds, and $\times$ pre-train expresses it as a multiple of the prior's own 6° pre-training regime.

Layout	**Method**	$\bar{\theta}$ (deg)	$\times$ pre-train	**Ep. len**
U-Maze	ASE-PPO	32.3 [31.5, 33.0]	5.4	60
	ASE-CRL	28.6 [27.9, 29.1]	4.8	60
	ASE-SAC+HER	14.6 [13.8, 15.5]	2.4	60
Large	ASE-PPO	31.7 [31.1, 32.3]	5.3	119
	ASE-CRL	20.6 [20.1, 21.1]	3.4	120
	ASE-SAC+HER	14.6 [13.8, 15.4]	2.4	119
Hard	ASE-PPO	32.7 [31.8, 33.6]	5.4	178
	ASE-CRL	15.5 [14.6, 16.0]	2.6	180
	ASE-SAC+HER	13.5 [12.8, 14.4]	2.2	177
Giant	ASE-PPO	31.4 [29.5, 33.3]	5.2	353
	ASE-CRL	9.6 [9.4, 9.9]	1.6	357
	ASE-SAC+HER	12.4 [12.0, 12.9]	2.1	353

Figure 13 decomposes the same quantity. The upper row shows the per-dimension increment $\boldsymbol{z}_{k+1} - \boldsymbol{z}_k$ across all 64 dimensions: every arm is sharply peaked at zero with heavy tails, and the Gaussian fits overlaid on the histograms understate that peak, which is why the excess kurtosis is reported alongside. ASE-CRL on Giant is the most extreme case ($\sigma = 0.028$, $\kappa = +18.4$), meaning it usually holds the latent almost fixed and occasionally moves it a long way, rather than drifting steadily. The lower row shows the norm $\|\Delta \boldsymbol{z}\|$, where 0 is a held skill and $\sqrt{2}$ an i.i.d. jump; the goal-conditioned arms concentrate closer to zero while ASE-PPO's mass sits around 0.5 on every layout.

Two caveats. This is a measurement of the commanded latent sequence, not of the resulting motion: it quantifies the specific failure mode Peng et al. (2022) identify, and a smoother latent trajectory is a necessary rather than a sufficient condition for natural movement, so we continue to make no claim about motion quality (Section 6). Second, the arms differ in success rate on these same layouts, and an episode spent failing is not composed of the same states as one spent reaching the goal, so part of the difference between arms may reflect where the policy spends its time rather than how it commands skills.

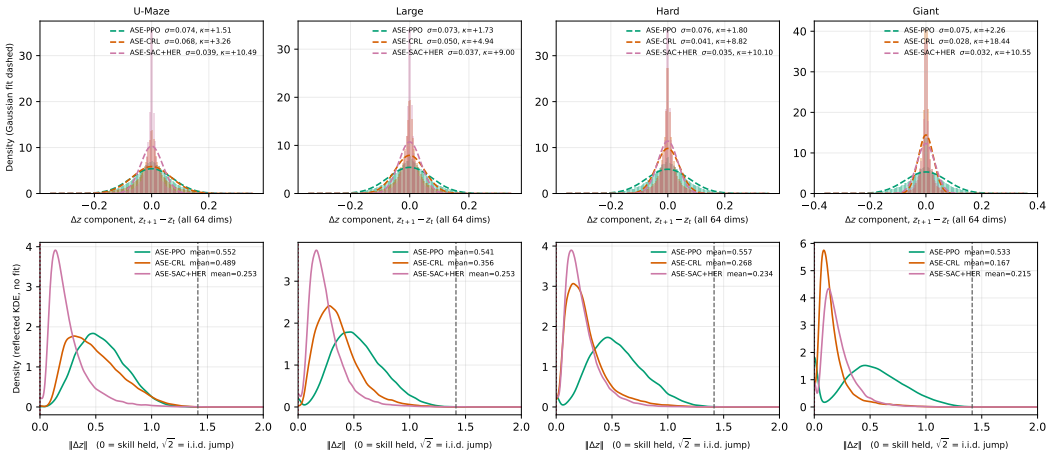


Figure 13: Distribution of the latent increment. Top: per-dimension components of $\boldsymbol{z}_{k+1} - \boldsymbol{z}_k$ over all 64 dimensions, with a Gaussian fit (dashed) and its standard deviation and excess kurtosis. Bottom: the norm $\|\Delta \boldsymbol{z}\|$, kernel density with no parametric fit, where 0 means the skill was held and the dashed line at $\sqrt{2}$ marks an i.i.d. resample.

A.11 Final success rates

Table 10 repeats Table 1 with the bootstrap confidence intervals and the full per-environment grouping.

Table 10: Success rate (%) at the end of training over five seeds. Each seed is summarized by the mean of its evaluation snapshots over the final tenth of the shared per-environment budget (Section 5); the mean, standard deviation and 95% percentile-bootstrap confidence interval of the mean (10^4 resamples) are then taken over the five per-seed summaries. The interval is one for the mean, not for the seed spread, and is coarse at five seeds, so the standard deviation is given alongside it and every seed is drawn individually in Section A.12.

Environment	Method	Mean	SD	95% CI
U-Maze	End-to-end PPO	61.4	8.0	[54.4, 66.1]
	End-to-end CRL	75.4	27.6	[50.7, 93.6]
	ASE-PPO	94.8	1.6	[93.5, 96.0]
	ASE-SAC+HER	72.7	3.6	[69.7, 75.2]
	ASE-CRL	**99.8**	0.2	[99.7, 99.9]
Large	End-to-end PPO	77.1	40.9	[40.4, 95.8]
	End-to-end CRL	78.8	21.1	[59.8, 89.6]
	ASE-PPO	87.0	1.5	[85.7, 88.0]
	ASE-SAC+HER	94.7	2.6	[92.7, 96.7]
	ASE-CRL	**99.8**	0.1	[99.8, 99.8]
Hard	End-to-end PPO	12.0	11.8	[6.7, 22.6]
	End-to-end CRL	50.4	18.8	[34.3, 62.6]
	ASE-PPO	75.4	1.2	[74.5, 76.4]
	ASE-SAC+HER	93.8	2.6	[91.8, 95.8]
	ASE-CRL	**99.4**	0.2	[99.2, 99.5]
Giant	End-to-end PPO	2.2	0.6	[1.8, 2.7]
	End-to-end CRL	9.7	12.3	[1.5, 20.5]
	ASE-PPO	37.3	1.0	[36.6, 38.2]
	ASE-SAC+HER	82.1	5.6	[77.9, 86.4]
	ASE-CRL	**97.4**	0.5	[97.0, 97.7]
Push-cube (open arena)	ASE-PPO	68.0	9.6	[59.7, 73.8]
	ASE-SAC+HER	15.6	2.2	[14.0, 17.3]
	ASE-CRL	**98.5**	0.5	[98.2, 98.9]
Push-cube (Large)	ASE-PPO	8.1	0.8	[7.5, 8.8]
	ASE-SAC+HER	0.2	0.3	[0.0, 0.4]
	ASE-CRL	**22.3**	15.8	[10.0, 34.4]

A.12 PER-SEED LEARNING CURVES

Figure 3 overlays the seeds of every method in one panel per environment. Figures 14 to 19 separate them: one panel per method, all five seeds drawn raw and unsmoothed, with the mean overlaid as a dashed reference. They are the evidence behind the seed-stability claims in Section 5.2, and in particular behind the wide intervals on the end-to-end arms, which come from bimodal outcomes rather than from uniform noise: on Large (Figure 15), four end-to-end PPO seeds solve the layout and one collapses.

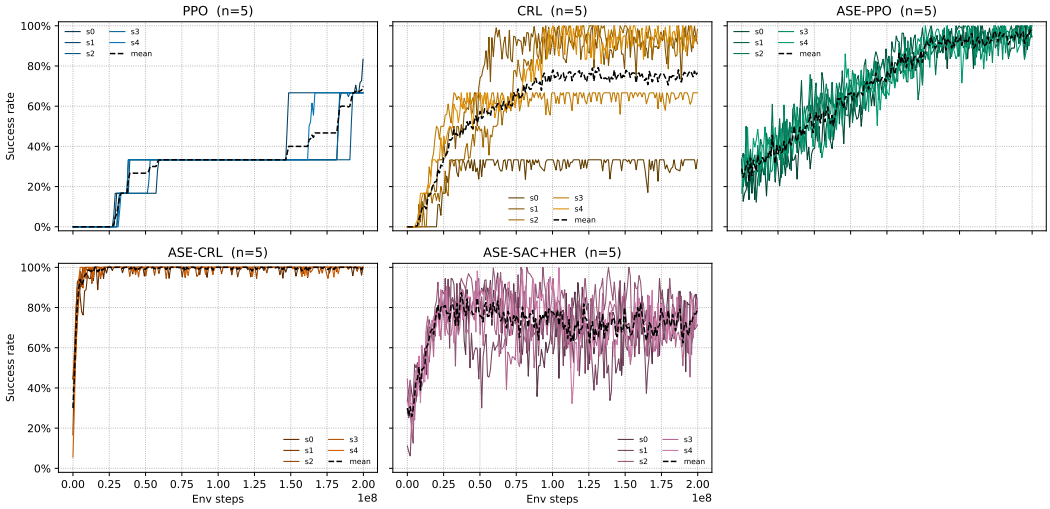


Figure 14: U-Maze, per seed.

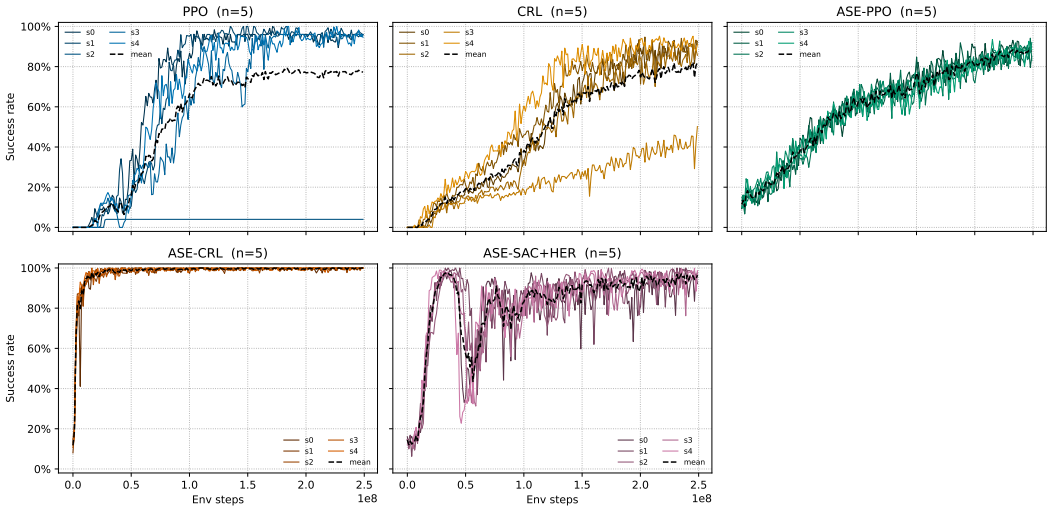


Figure 15: Large, per seed.

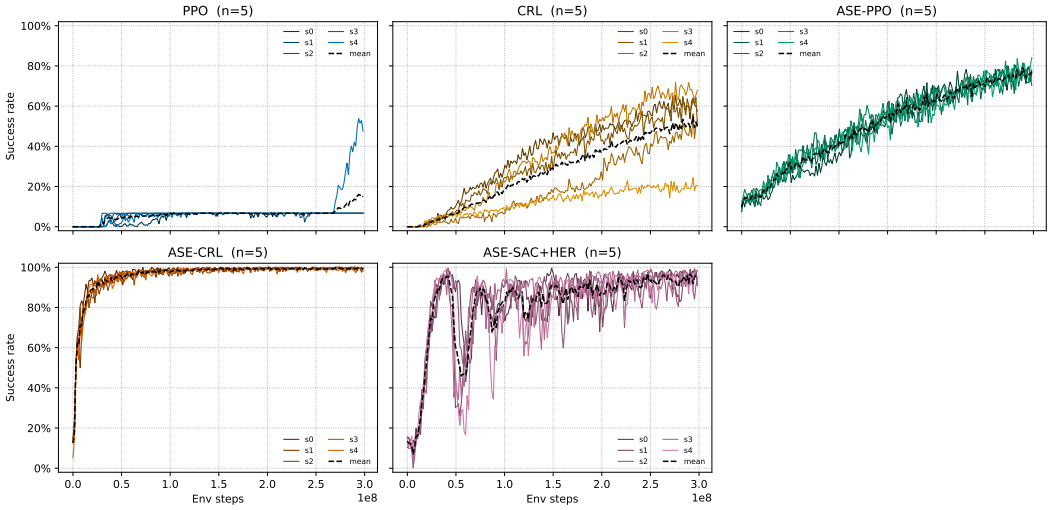


Figure 16: Hard, per seed.

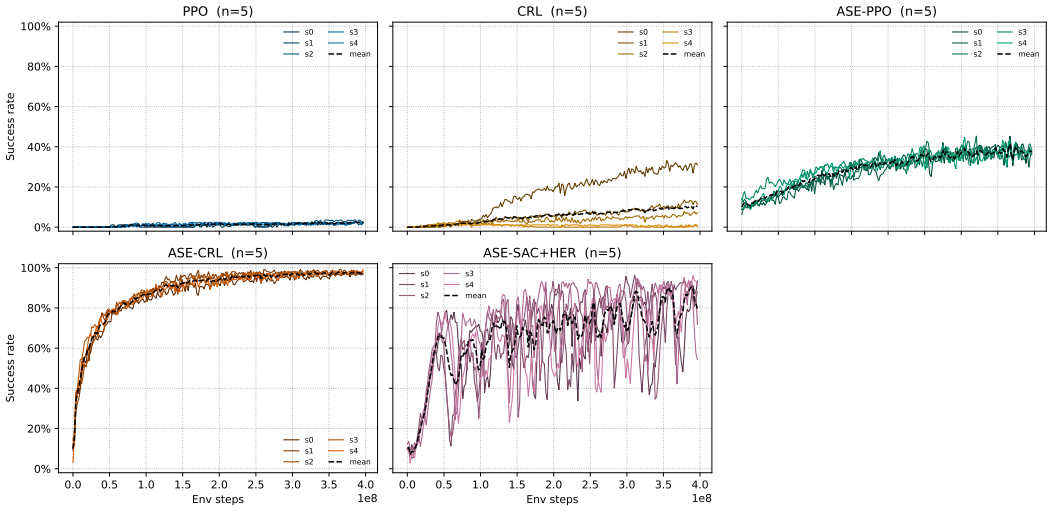


Figure 17: Giant, per seed.

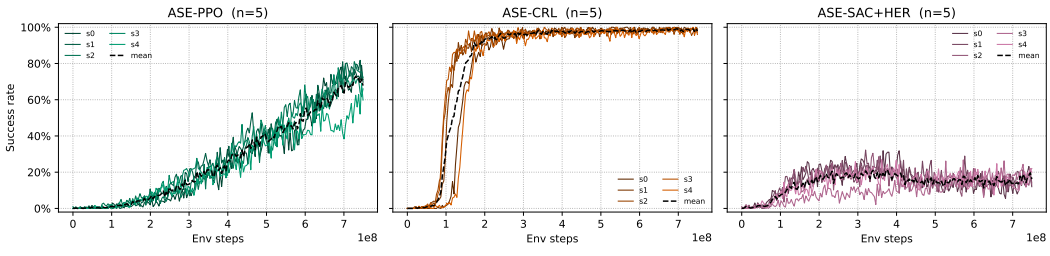


Figure 18: Push-cube (open arena), per seed.

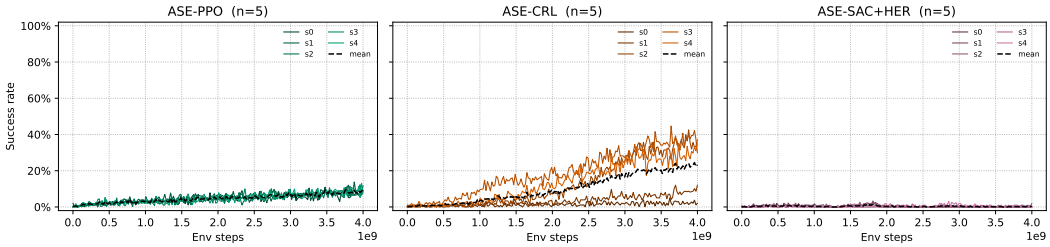


Figure 19: Push-cube (Large), per seed.

A.13 COMPUTE

All training runs were executed on NVIDIA GeForce RTX 4090 GPUs (24 GB each, driver 545.23.08), one training job pinned to one GPU, so the wall-clock times below are a direct proxy for GPU-hours. The software stack is PyTorch 2.7.0 built for CUDA 12.8, Isaac Sim 5.1.0.0 and Isaac Lab 0.54.3; the complete frozen environment is released with the code.

Table 11 gives measured wall-clock hours per configuration. The dominant cost is not the largest maze but the maze-embedded pushing task, whose 4 B environment-step budget is an order of magnitude above the mazes. The second observation is the price of the end-to-end tier: end-to-end CRL at depth 64 costs roughly six times its hierarchical counterpart per environment step, and does so at every maze scale.

Table 11: Measured compute, five seeds per configuration. Hours are per-seed wall-clock ranges and their sum across seeds.

Task	Method	Seeds	Hours per seed	Total (h)
U-Maze	End-to-end PPO	5	4.9–5.3	25
	End-to-end CRL	5	36.9–38.4	188
	ASE-PPO	5	2.8–3.4	15
	ASE-SAC+HER	5	5.0–5.3	25
	ASE-CRL	5	5.9–6.3	31
Large	End-to-end PPO	5	5.6–5.9	29
	End-to-end CRL	5	45.9–47.4	234
	ASE-PPO	5	3.6–4.3	20
	ASE-SAC+HER	5	6.2–7.3	32
	ASE-CRL	5	7.4–8.0	38
Hard	End-to-end PPO	5	5.5–6.0	28
	End-to-end CRL	5	51.6–53.2	261
	ASE-PPO	5	4.2–4.9	23
	ASE-SAC+HER	5	7.4–7.5	37
	ASE-CRL	5	8.5–8.7	43
Giant	End-to-end PPO	5	6.8–7.2	35
	End-to-end CRL	5	69.7–71.7	354
	ASE-PPO	5	5.9–6.8	31
	ASE-SAC+HER	5	12.1–13.1	63
	ASE-CRL	5	11.6–11.9	58
Push-cube (open arena)	ASE-PPO	5	9.7–11.5	54
	ASE-SAC+HER	5	18.0–18.3	90
	ASE-CRL	5	21.6–21.9	109
Push-cube (Large)	ASE-PPO	5	70.3–75.7	359
	ASE-SAC+HER	5	118.0–131.5	628
	ASE-CRL	5	122.9–126.1	623
Giant, proprioception	ASE-PPO	5	6.6–7.6	36
	ASE-CRL	5	12.1–12.3	61
Reward ablation	ASE-PPO, U-Maze	5	3.1–3.2	16
	ASE-PPO, Giant	5	5.7–6.8	32

The runs in Table 11 total approximately 3220 GPU-hours. Hours are measured at the shared per-environment budget. To this must be added the one-off cost of the low-level prior: 14.85 B environment steps over 226,600 iterations, 107 wall-clock hours. That cost is incurred once and amortized over every hierarchical run reported here (four mazes, two pushing tasks, three learning algorithms and five seeds) and would be amortized further by any downstream task reusing the same prior. It is nonetheless real, and a fair accounting of the hierarchical arms should include it: it is roughly comparable to the total cost of the ASE-CRL maze runs themselves.

A.14 FURTHER LIMITATIONS

Two of the limitations noted in Section 6 need more than a clause.

The critic is symmetric. We follow Wang et al. (2025) in scoring goals with the negative Euclidean distance $C(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{g}) = -\|\phi(\boldsymbol{s}, \boldsymbol{a}) - \psi(\boldsymbol{g})\|_2$. Beyond restricting the achievable critic class (Section A.1), this parameterization measures a symmetric distance in embedding space, which limits its ability to express a pair of states that is cheap to traverse in one direction and expensive in the other. Our maze layouts are close to symmetric in that sense, since corridors are traversable both ways and the character can turn in place, which is presumably why the restriction costs little here. The pushing tasks are not: a block pushed into a corner is far harder to recover than it was to put there. Quasimetric RL (Wang et al., 2023), noted in Section 2 as a refinement we did not adopt in order to keep the high-level learner identical to the end-to-end baseline, drops the symmetry requirement while keeping the triangle inequality, and is the natural comparison for a version of this benchmark that stresses irreversibility.

The high-level observation is a projection. The high-level policy sees root position, root orientation and the goal, plus the block centroid and its velocity on the pushing tasks (Table 3). It does not see proprioception, root velocity, or block orientation, so it faces a POMDP that we address with a memoryless policy rather than with recurrence or a history buffer. The proprioception ablation (Section A.6) bounds the cost of the character-state omission directly, and finds it negligible for ASE-CRL. It says nothing about the other two. Block orientation is the omission most likely to bind: the block's yaw governs which faces are presented to the character and how it slides under an off-center push, and unlike the character's own state it cannot be recovered from an observation history that a memoryless policy does not keep. A reader should therefore treat the pushing results as a lower bound on what this interface supports, and read the maze results, where the omitted quantities matter far less, as the cleaner test of the claim.