OUTPERFORMING 1000 EUCLIDEAN LAYERS WITH HYPERBOLIC GOAL-CONDITIONED RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Scaling self-supervised contrastive reinforcement learning (CRL) to 1000-layer Euclidean neural networks has enabled state-of-the-art goal-reaching performance, yet wall-clock time scales linearly with depth. We find that changing the geometry achieves stronger performance with far shallower networks. This matters in complex navigation environments, where walls can make route distance deviate substantially from Euclidean distance. In a U-shaped maze, we show that the longer the route, the more it deviates from Euclidean distance. Hyperbolic CRL addresses this mismatch by replacing Euclidean distances in CRL's contrastive objective with geodesic distances in hyperbolic space. On hard navigation tasks, inductive bias beats scale: depth 64 hyperbolic agents outperform 1024-layer-deep Euclidean networks while training more than $12\times$ faster and using about $13\times$ fewer parameters. We observe that hyperbolic agents behave qualitatively better in hard humanoid mazes, where they follow corridors to reach goals, whereas Euclidean agents try straight-line shortcuts over walls. Our results show that matching the representation geometry to the task can substitute for substantial network depth.

1 INTRODUCTION

Consider a humanoid having to find the goal in a maze. Navigating purely based on straight-line distance would lead this agent down many dead ends, because straight-line distance is a misleading measure of how far a goal is in the maze. In this setting, the *straight-line distance* between two positions ignores walls, whereas the *route distance* is the length of the actual path between a state and a goal. Goal-conditioned RL (GCRL) trains agents to reach arbitrary goals without a hand-designed reward. Contrastive reinforcement learning (CRL) (Eysenbach et al., 2022) casts GCRL as a classification problem and uses an InfoNCE objective (van den Oord et al., 2018) to predict whether a goal is reachable from a state-action pair, based on the Euclidean distance between their embeddings (Bortkiewicz et al., 2025a). Euclidean embeddings make straight-line proximity easy to represent, and this proxy works at short range but misleads once walls are present. Wang et al. (2025, Figure 9) show that on Ant U4-Maze, Euclidean agents learn to follow route structure only when using deep networks.

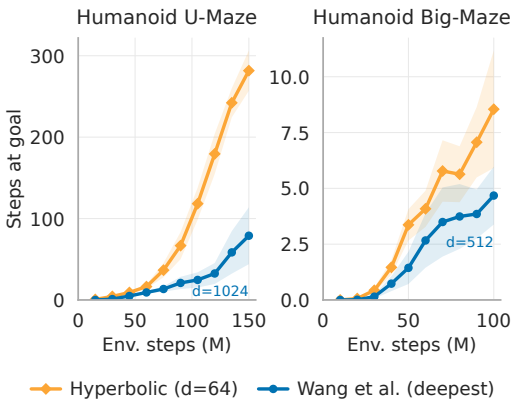


Figure 1: **Hyperbolic agents outperform much deeper Euclidean ones.** A hyperbolic agent of depth 64 reaches a higher time at goal than the depth 1024 Euclidean agent while using $12\times$ less wall-clock time. Euclidean results are taken from Wang et al. (2025, Figure 12), we use their step budget. Wall-clock times are reported in Appendix F.

This reliance on depth is expensive. Recent work scales CRL with GPU-accelerated simulation (Bortkiewicz et al., 2025a) and networks up to 1024 layers deep (Wang et al., 2025). However, these Euclidean agents perform best at extreme depth, making training very expensive (Figure 1).

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.

Wang et al. (2025, Section 4.4) evaluate 1024-layer networks on a single task, Humanoid U-Maze, and report that computational constraints kept them from scaling further. Our Ant U5-Maze study shows why: Euclidean representations must learn to "unroll" the maze to reflect route distance, which can require substantial depth. We therefore hypothesize that deep Euclidean agents use part of their capacity to compensate for the mismatch between the geometry of their embeddings and the geometry of the environment. This raises our paper's central question: *Is Euclidean distance the right geometry for goal-reaching?*

Hyperbolic space offers a natural alternative. It is approximately Euclidean at small scales but tree-like at large scales, and trees embed in it with arbitrarily low distortion (Sarkar, 2011). Existing hyperbolic RL methods (Cetin et al., 2023; Klein et al., 2026), however, assign no meaning to embedding distance and learn hierarchy only indirectly through their standard RL objectives. Quasimetric approaches (Wang & Isola, 2022; Liu et al., 2023; Wang et al., 2023; Myers et al., 2024) use asymmetric distances but retain their Euclidean embedding geometry. Consequently, quasimetric CRL agents perform poorly on mazes (Wang et al., 2025, Appendix A.4). To our knowledge, no prior work has adopted hyperbolic spaces for improving CRL.

Our work investigates this change. Hyperbolic CRL keeps the contrastive objective but replaces Euclidean distances of embedded points with geodesic distances, at a small, fixed computational cost. We motivate this choice with a case study in Ant U5-Maze. Locally, route distances in this environment coincide with straight-line distances, while globally they become increasingly tree-like. Hyperbolic space matches this metric structure. At equal depth, hyperbolic agents outperform Euclidean agents on most tasks. On the Humanoid mazes, they beat Euclidean agents with many times deeper networks. These results suggest that a better choice of representation geometry can substitute for substantial network depth.

Key Contributions

1. **Hyperbolic CRL.** We introduce a drop-in modification to CRL that projects embeddings onto hyperbolic space and estimates goal reachability by geodesic distance. This gives the agent an inductive bias toward tree-like route structure at a small computational cost.
2. **Geometry of goal-reaching.** Section 4 shows that maze route distances are approximately Euclidean at short range and increasingly tree-like at long range. Hyperbolic space captures this structure more faithfully than Euclidean geometry, losing its straight-line bias.
3. **A better performance-efficiency frontier.** At fixed depth, hyperbolic agents outperform Euclidean agents on most tasks and stitch long routes more reliably. Figure 1 shows depth 64 hyperbolic agents outperforming Euclidean baselines up to $16\times$ deeper while training in at least $7\times$ less wall-clock time.

2 Background

We model goal reaching (Kaelbling, 1993; Andrychowicz et al., 2017) as a goal-conditioned MDP $\mathcal{M}_g = (\mathcal{S}, \mathcal{A}, p_0, p, p_g, r_g, \gamma)$. The agent starts in a state $s_0 \sim p_0$. At each step t, it observes a state $s_t \in \mathcal{S}$, takes an action $a_t \in \mathcal{A}$, and moves to $s_{t+1} \sim p(\cdot \mid s_t, a_t)$. Goals $g \in \mathcal{G}$ are sampled from p_g, and a function $f : \mathcal{S} \to \mathcal{G}$ extracts the goal-relevant part of a state, such as the agent's (x, y) position (Wang et al., 2025). We write $g_t \doteq f(s_t)$ for the goal achieved at step t. Following Eysenbach et al. (2022), the reward is the density of achieving g at the next step, $r_g(s_t, a_t) \doteq (1-\gamma)\, p(g_{t+1} = g \mid s_t, a_t)$, with discount factor $\gamma \in [0, 1)$. A goal-conditioned policy $\pi_g(a \mid s) \doteq \pi(a \mid s, g)$ with commanded goal g is trained to maximize the expected return $\mathbb{E}_{p_0, p_g, \pi_g}\big[\sum_{t=0}^{\infty} \gamma^t r_g(s_t, a_t)\big]$. Letting p^{π_g} denote densities under trajectories generated by the policy π_g. Then, the Q-function of π_g is a density over goals:

$$Q_g^{\pi}(s, a) = p_{\gamma}^{\pi_g}(g \mid s, a) \doteq (1-\gamma) \sum_{k=0}^{\infty} \gamma^k\, p^{\pi_g}(g_{t+k+1} = g \mid s_t = s, a_t = a). \tag{1}$$

Thus, $Q_g^{\pi}(s, a)$ is the discounted visitation density of the policy at the goal.

Contrastive Reinforcement Learning. Contrastive Reinforcement Learning (CRL) estimates this density, averaged over goals, via classification rather than temporal-difference (TD) learning (Eysenbach et al., 2022). CRL uses two encoders $\phi\colon \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$ and $\psi\colon \mathcal{G} \to \mathcal{Z}$ that embed into a shared

representation space $\mathcal{Z}$. An energy is computed using the similarity between the two embeddings:

$$e(s, a, g) \triangleq \mathrm{sim}\big(\phi(s, a),\ \psi(g)\big) \tag{2}$$

Here $\mathrm{sim}\colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is a similarity function on $\mathcal{Z}$, e.g., a negative distance (Bortkiewicz et al., 2025a), the dot product, or the cosine similarity. CRL is an actor-critic method: its critic is the energy e and its actor is the policy $\pi_\theta(a \mid s, g)$ with parameters θ. The policy is then trained to maximize e. The critic is trained with the InfoNCE objective (van den Oord et al., 2018; Bortkiewicz et al., 2025a) over minibatches of (s_t, a_t, g_{t+k}) tuples, where the positive sample g_{t+k} is the goal achieved at a future step $t+k$ of the same trajectory, with $k \sim \mathrm{Geom}(1-\gamma)$ on $\{1, 2, \dots\}$. Achieved goals from other trajectories in the minibatch are treated as negative samples. The implementation we build on uses the negative Euclidean distance (Bortkiewicz et al., 2025a) as sim. This paper replaces it with the negative geodesic distance in hyperbolic space.

Hyperbolic geometry. Hyperbolic space is a simply connected Riemannian manifold of constant negative sectional curvature $-c$, with curvature magnitude $c > 0$. In this work we use the *hyperboloid* (Lorentz) model, which represents hyperbolic space as the forward sheet of a two-sheeted hyperboloid $\mathbb{H}^d_c = \{\boldsymbol{x} \in \mathbb{R}^{n+1} : \langle \boldsymbol{x}, \boldsymbol{x} \rangle_L = -1/c,\ x_0 > 0\}$ in Minkowski space (Lee, 2018). Here, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_L := -x_0 y_0 + \sum_{i=1}^{n} x_i y_i$ denotes the Minkowski inner product.

Hyperbolic encoding. The output of a Euclidean encoder $\boldsymbol{u} \in \mathbb{R}^n$ is mapped to the hyperboloid in two steps (Nickel & Kiela, 2018). First, $\boldsymbol{u}$ is identified with the vector $\boldsymbol{v} = (0, \boldsymbol{u})$ of the tangent space $\mathcal{T}_{\bar{\mathbf{0}}}\mathbb{H}^d_c$ at the origin $\bar{\mathbf{0}} = (1/\sqrt{c}, \mathbf{0})$. Second, the exponential map $\exp_{\bar{\mathbf{0}}}$ projects $\boldsymbol{v}$ onto the manifold. The closed-form expressions are given in Appendix B.1 alongside other hyperbolic space operations.

3 Related Work

Goal-conditioned reinforcement learning. Goal-conditioned RL (GCRL) trains a policy to reach any target state by conditioning it on that goal (Kaelbling, 1993; Schaul et al., 2015). Hindsight experience replay (Andrychowicz et al., 2017) implemented GCRL by relabeling failed trajectories with the goals they actually reached. We build on Contrastive RL (CRL) (Eysenbach et al., 2022), which trains the critic using an InfoNCE objective (van den Oord et al., 2018; Bortkiewicz et al., 2025a) rather than explicitly learning rewards or distances. Our implementations follow Bortkiewicz et al. (2025a) for online CRL and OGBench (Park et al., 2025) for offline CRL, which include recent improvements such as stronger regularization (Zheng et al., 2024). Beyond GCRL, scaling network size has long been difficult in deep RL. Recent work has made them trainable using regularized residual critics (Nauman et al., 2024; Lee et al., 2025a;b) or mixture-of-experts layers (Obando-Ceron et al., 2024). Wang et al. (2025) adapted these architectures to CRL and trained agents with up to 1024 layers. Our work is complementary: we keep the CRL objective and change the geometry of its representations.

Quasimetric RL. Bortkiewicz et al. (2025a) observe that CRL works best when energies satisfy the triangle inequality, e.g., the Euclidean distance outperforms the squared Euclidean distance. This fits the view that temporal distances satisfy the triangle inequality and that goal-reaching distances should enforce it by construction (Wang et al., 2023; Myers et al., 2024). Temporal distances are asymmetric, so prior work builds quasimetrics into both architectures (Wang & Isola, 2022; Liu et al., 2023) and goal-reaching algorithms (Wang et al., 2023; Myers et al., 2024; 2025; Zheng et al., 2026). While hyperbolic geodesic distances also satisfy the triangle inequality, they depart from Euclidean distance in a different way. Quasimetrics are asymmetric but preserve the flat coordinates of Euclidean geometry, whereas hyperbolic distances preserve symmetry while changing the curvature. Wang et al. (2025, Figure 15) report that the quasimetric CRL agent of Myers et al. (2024) performs poorly maze-like tasks, which motivate our work.

Hyperbolic representation learning. Hyperbolic embeddings outperform Euclidean ones on hierarchical data, especially in low dimensions (Nickel & Kiela, 2017; 2018), because hyperbolic space embeds trees with arbitrarily low distortion (Sarkar, 2011; Sala et al., 2018). Since Ganea et al. (2018) introduced hyperbolic neural networks, later work has added linear layers (Shimizu et al., 2021; Yang et al., 2024; van der Klis et al., 2026) that avoid Riemannian optimization (Bonnabel, 2013; Bécigneul & Ganea, 2019), convolutional layers (Bdeir et al., 2024), and self-attention (Yang et al., 2024). Contrastive learning optimizes relative distances, and is thus well-suited for hyperbolic geometry (Suris et al., 2021; Ge et al., 2023; Mettes et al., 2024), particularly in vision-language alignment (Desai et al., 2023; Pal et al., 2025). In deep RL, Cetin et al. (2023) introduced

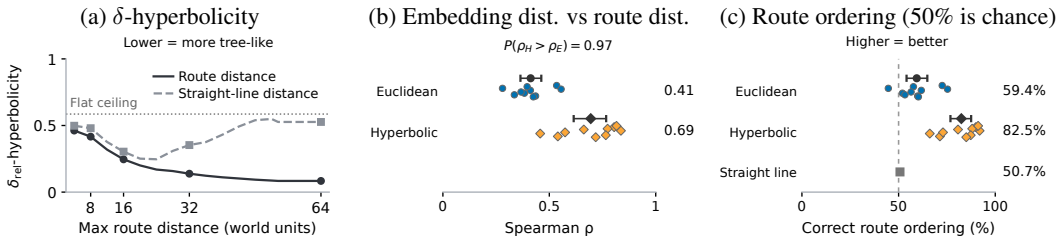


Figure 2: **Hyperbolic critics better reflect the geometric structure of Ant U5-Maze. 2a)** *The maze becomes increasingly more tree-like at longer task horizons under route distance, when walls are taken into account.* We report relative δ-hyperbolicity of Ant U5-Maze (lower is more tree-like) under increasing task horizons. **2b)** *Hyperbolic embedding distances track route distances more closely than Euclidean embedding distances.* $P(\rho_H > \rho_E)$ estimates the probability that a hyperbolic run has a higher Spearman correlation than a Euclidean run. **2c)** We sample maze locations with similar straight-line distance but different route distance. The percentage shows how often the critic assigns the longer route a larger embedding distance. *The hyperbolic critic guides the agent along the correct route more frequently.* In 2b and 2c, we show the 10 seeds per method with means and 95% bootstrap confidence intervals.

hyperbolic representations and argued that MDP states evolve like a tree by measuring the relative δ-hyperbolicity (Gromov, 1987; Khrulkov et al., 2020) of learned representations. Klein et al. (2026) identified why hybrid hyperbolic-Euclidean agents fail to train and stabilized them with regularization, improving over prior hyperbolic agents on ProcGen and Atari. Neither work tests whether the environment itself is tree-like. To our knowledge, we are the first to use hyperbolic representations in goal-conditioned RL.

4 Case Study: Analyzing the Geometry of a Deceptive Maze

We study the geometry of goal-reaching in Ant U5-Maze, a complex navigation task in which the agent starts on the left corner of a U-shaped corridor (Figure 3). In this task, distant goals lie in the opposing right corner, and reaching them requires traversing the entire environment. We measure distances in **world units**, where one world unit is one unit of the simulator's (x, y) coordinates, and use the following two distances:

Route distance: length of the shortest path between two positions within the maze.

Straight-line distance: Euclidean distance between two positions, ignoring walls.

Following Section 2, Euclidean and hyperbolic critics use the negative embedding distance as the energy: $Q(s, a, g) = e(s, a, g) = -d\big(\phi(s, a), \psi(g)\big)$, where d is the Euclidean distance or the geodesic distance on the hyperboloid, respectively. We analyze these critics from three perspectives: the geometry of the environment, the learned embedding distances, and the spatial variation in the critic's value. Appendix C.2 contains details on how all of the following experiments are implemented.

Is the environment tree-like? Intuitively, navigation in a tree-like environment depends on shared routes that branch towards distinct goals. This structure is often difficult to discern locally but becomes clear at global scales. Thus, we examine whether the maze's geometry becomes increasingly tree-like at longer task horizons. Since route distances reflect how different parts of the environment connect, they let us quantify tree-like geometry using Gromov's δ-hyperbolicity (Gromov, 1987). This measure quantifies how closely the distances among four given locations in the environment resemble a tree, where a smaller δ indicates a more tree-like geometry (see Appendix B.2). To make comparison meaningful across the different scales, we normalize the value by the task horizon and report relative hyperbolicity $\delta_{\text{rel}} \in [0, 1]$ instead (Khrulkov et al., 2020; Ermolov et al., 2022; Bdeir et al., 2024).

Figure 2a evaluates δ_{rel} over progressively larger horizons, comparing route distances with straight-line distances between the same points (see Appendix C.2). As the horizon grows, the maze becomes

increasingly tree-like with respect to route distance, while straight-line distances show the opposite trend. In particular, Ant U5-Maze is locally close to flat and tree-like at a global scale. This transition from local Euclidean to globally tree-like geometry is characteristic of hyperbolic geometry: the hyperboloid is approximately Euclidean at scales below its curvature radius $1/\sqrt{c}$ and exhibits tree-like geometry at larger scales. Section 5 shows that matching the critic's geometry to this structure improves performance in mazes. Importantly, this does not mean that hyperbolic representations are necessary to solve Ant U5-Maze. Euclidean networks can also learn the maze's tree-like geometry, but they may require much larger depth to do so. In contrast, hyperbolic geometry provides this tree-like structure directly as an inductive bias. Figure 1 shows the effects of depth and geometry for Humanoid U-Maze.

Do embedding distances track route distance? Next, we study whether distant locations in the environment are associated with larger embedding distances between their representations. We evaluate every critic on the same set of 20,000 fixed state-goal pairs, sampled uniformly from rollouts of Euclidean and hyperbolic agents across the observed range. For each critic, we compute the Spearman rank correlation ρ between embedding distance and route distance. This measures how well the critic orders goals by the distance the agent must travel to reach them. Figure 2b shows that hyperbolic critics order goals by route distance more consistently than Euclidean critics, as seen by the significantly higher ρ. A high correlation alone does not indicate that the critic has learned the maze's geometry since straight-line distance also correlates with route distance. Therefore, a critic that only tracks straight-line distance has a high correlation, too. To isolate this effect, we select pairs of goals with similar straight-line distance but route distances that differ by at least 10 world units. Then, we check whether the critic assigns the larger embedding distance to the goal with the longer route. Ranking by random chance yields exactly 50%, which a ranking by straight-line distance roughly achieves. Figure 2c shows that hyperbolic critics order such goals more precisely than Euclidean critics. Thus, hyperbolic critics place distant locations farther apart in embedding space, even when their straight-line distance suggests otherwise.

Does the critic point the agent in the right direction? Finally, we ask whether these geometric differences change the direction in which the critic guides the agent. To that end, we compute the gradient $\nabla_{x,y} Q(s, a, g)$ of the critic with respect to the agent's position (x, y) in state s, holding the action a fixed. We evaluate this gradient on state-goal pairs that are far apart in terms of route distance but close in straight line distance, so that moving straight towards the goal leads into a wall. Hence, the agent must traverse the maze. Figure 3b reports the cosine similarity between $\nabla_{x,y} Q$ and two directions: straight towards the goal through the wall, and the direction that minimizes route distance. Euclidean critics tend to direct the agent toward the wall rather than along the route, whereas hyperbolic critics do not exhibit this bias. The value function visualization in Figure 3a shows the same pattern. While the Euclidean critic assigns high values across walls, the hyperbolic critic does not.

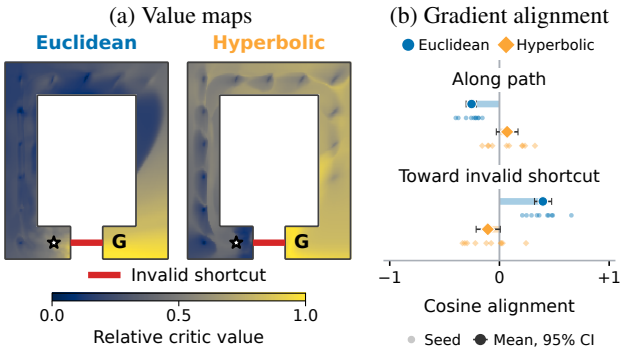


Figure 3: **Critic values and gradients in Ant U5-Maze.** Both critics have depth 16. **3a)** *The Euclidean critic assigns high Q-values across the wall near the start, whereas the hyperbolic critic values increase along the route towards the goal (brighter colors).* ⋆ = start, **G** = goal, and — = invalid shortcut. **3b)** For state-goal pairs with long routes but short straight-line distances, we measure the cosine similarity between $\nabla_{x,y} Q$, and two directions: along the optimal maze route and straight towards the goal. *Euclidean gradients tend to point towards the wall, while hyperbolic critics do not exhibit this bias*. Results are averaged over 10 seeds.

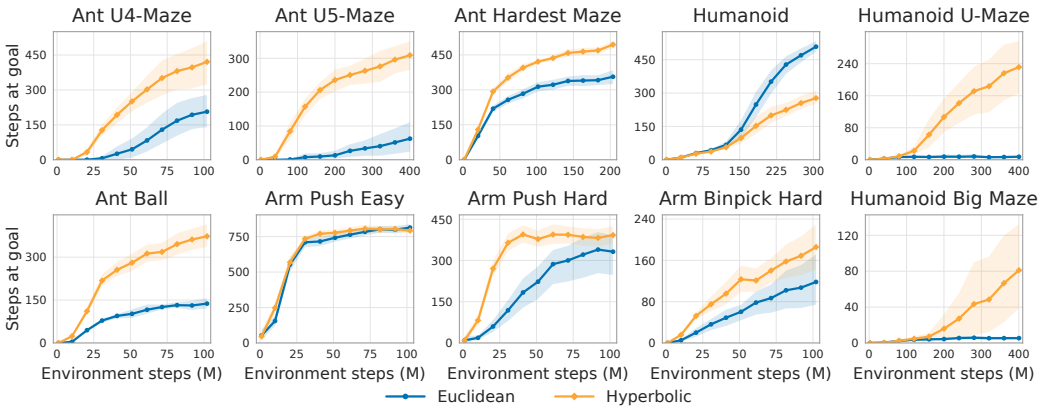


Figure 4: **Hyperbolic agents outperform Euclidean ones at fixed depth.** Hyperbolic improvements are particularly pronounced in hard navigation environments, e.g., Humanoid mazes. Humanoid is not suitable for hyperbolic representations because route distances are straight lines. All agents use depth 16, beyond which gains flatten (Figure 8). We report the mean across 10 seeds with 95% bootstrapped confidence intervals.

5 Experiments

We build on the JaxGCRL implementation of Bortkiewicz et al. (2025a) and follow the benchmark setup of Wang et al. (2025), with two changes: we replace one Ant Maze with Ant Ball to cover a wider range of task types, and we evaluate Humanoid under a different protocol. The difference pertains to the frequency of the physics simulation, detailed in Appendix C.3. Appendix C lists all used environments. We use the steps-at-goal as our performance measure, following Wang et al. (2025), and report per-environment means with 95% bootstrap confidence intervals (CIs) (Agarwal et al., 2021). Both Euclidean and hyperbolic agents use a scaled ResNet encoder for the actor and the critic (Wang et al., 2025). We choose depth 16 for our main performance experiments based on our scaling results in Section 5.4, which presents depth-scaling ablations of our models on selected environments. The hyperbolic agent adds an exponential map at the origin and a single HTC transformation (Yang et al., 2024) after the Euclidean encoders' representation. At depth 16, this adds approximately 0.26% to the total parameter count compared to fully Euclidean networks, but yields substantial improvements across many tasks in the full benchmark suite. We use HTC over other hyperboloid layers due to its simpler implementation, and rely on the hyperbolix library (Klein & Lang, 2026) for JAX implementations of hyperbolic operations. We fix the curvature magnitude to $c = 0.5$ (i.e., sectional curvature $-c$) for all hyperbolic agents. Our baseline is Euclidean CRL, since Wang et al. (2025) show that other methods, such as HER (Andrychowicz et al., 2017) or CMD-1 (Myers et al., 2024), are not competitive.

5.1 Main Results

As Figure 1 highlights, we find that hyperbolic agents improve the most over the baselines on the two hardest tasks in the suite, Humanoid U-Maze and Humanoid Big Maze. In these two environments, our agents substantially outperform depth 1024 and depth 512 Euclidean agents at depth 64, a fraction of the Euclidean depth. Figure 4 shows training curves for the full suite comprising locomotion, navigation, and manipulation tasks. The strong performance of hyperbolic agents on Ant U-Mazes connects directly to Section 4: evaluation in these environments uses only far-away goals, exactly the scenario where our case study predicts the strongest advantage for hyperbolic representations. In both Humanoid mazes, the hyperbolic agents make steady learning progress, while the Euclidean agent barely gets off the ground. Section 5.2 analyzes the behavioral difference between the two agents. We discuss Humanoid and Ant Ball separately because both environments are open arenas without walls. First, Humanoid's difficulty lies in locomotion, and every route to the goal is a straight line in the plane. Without walls, route distance generally equals straight-line distance, which Section 4 finds far less tree-like than route distance in mazes. We therefore expect no hyperbolic advantage on Humanoid. The observed gap comes from our default curvature $c = 0.5$, which

imposes a strongly curved geometry on a "flat" environment. A lower curvature closes the gap, as our ablations in Section 5.4 show. Ant Ball uses the same arena layout as Humanoid but requires pushing a ball to the goal. Surprisingly, the hyperbolic agent wins here. Visual inspection of the learned policies in Figure 14 reveals that hyperbolic agents reposition themselves around the ball more often and push harder. This repositioning requires curved paths even in an open arena, and we hypothesize that this is why hyperbolic representations help on this task.

5.2 Behavioral Evidence of Invalid Shortcuts in Euclidean Agents

The straight-line bias discussed in Section 4 is also evident in the behavior of Euclidean agents. Figure 5 shows a representative successful trajectory of each agent on Humanoid U-Maze for a distant goal that requires navigating around the wall. The Euclidean agent takes a straight-line shortcut: it runs at the wall, vaults over it, lands in the goal region, and falls, which ends the episode. The hyperbolic agent instead walks around the corner and stands at the goal. Wang et al. (2025) report a similar vault for depth 256 Euclidean agents and present it as a new capability. Our depth 16 Euclidean agent also vaults, but falls on landing. We consider this vault undesirable because it achieves the goal without learning the locomotion behavior required to stay on it. Consistent with this, the depth 1024 Euclidean agents of Wang et al. (2025) spend around 100 steps at the goal on Humanoid U-Maze, whereas our depth 64 hyperbolic agent stays for 240. One consistent difference between the agents across all seeds is locomotion behavior. Euclidean agents typically fall within 90 steps and barely manage to stay at the goal, even when the goal is in the same corridor as the starting state. Hyperbolic agents walk to these goals and stay at them in most episodes. Goals behind the inner wall remain difficult for both agents. Scaling Euclidean networks up to depth 64 still yields agents that rarely stay at goals, consistent with the collapse reported by Wang et al. (2025) at depth 64. We hypothesize that the straight-line bias of Section 4 contributes to the Euclidean agents' failure to learn a stable gait at lower depths because they try to take the shortcut over the wall.

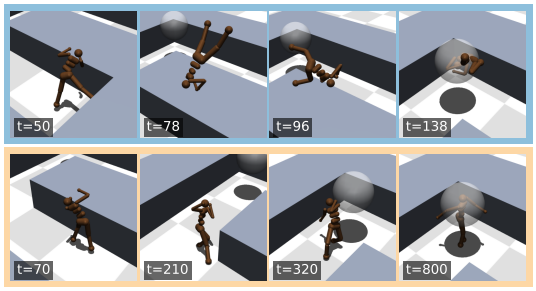


Figure 5: **Behavioral evidence of straight-line bias**. Similar to Wang et al. (2025), we observe Euclidean agents trying to vault themselves over walls instead of learning to navigate the maze. In contrast, hyperbolic agents follow the corridors, learn locomotion behavior earlier, and achieve higher rewards.

5.3 Hyperbolic Representations Improve Stitching

Hyperbolic representations also improve experience stitching, a form of generalization intrinsic to deep RL (Bortkiewicz et al., 2025b). We follow the protocol of Wang et al. (2025) on Ant U-Maze, where agents train only on start-goal pairs at most three units apart and are evaluated on the three goals on the other side of the connecting corridor. This means that every evaluation pair requires combining short training trajectories. Figure 6 shows that hyperbolic agents stay on average longer at the goal than Euclidean agents at every depth, with the largest gap at depth 4. At depth 16, hyperbolic agents already exceed the mean of depth 64 Euclidean agents. Beyond depth 16, gains in mean performance flatten for both geometries, but hyperbolic agents become more consistent across seeds as depth increases to 64, narrowing the confidence intervals. Section 4 suggests why hyperbolic agents stitch better: the evaluation goals lie close to the start in straight-line distance but far apart in the maze distance. This aligns with our previous analysis of Figure 2c, showing that hyperbolic critics order such goals by route distance more reli-

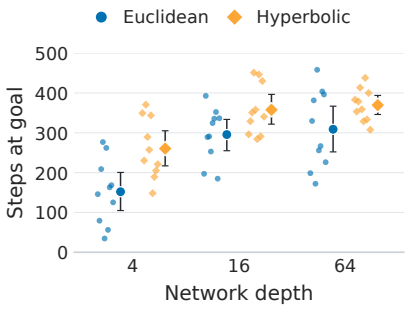


Figure 6: **Stitching on Ant U-Maze**. Markers show the mean time at goal with 95% bootstrap CIs.

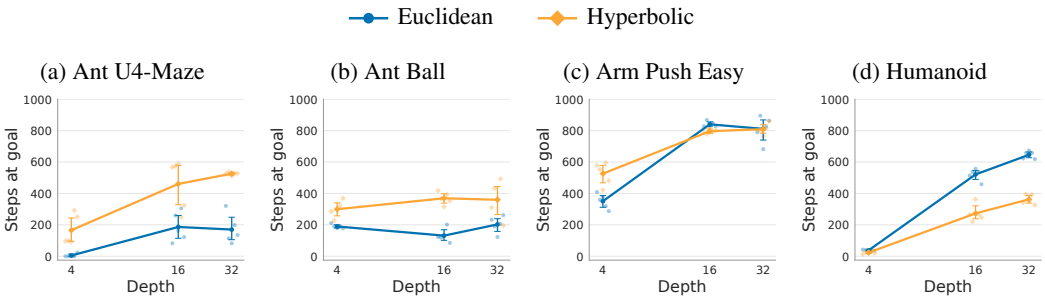


Figure 8: **Effect of actor and critic depth scaling.** Hyperbolic agents generally perform better as the Euclidean encoder grows deeper, although the size of the gain varies across environments. We report the mean over five seeds with 95% bootstrap confidence intervals.

ably than Euclidean critics. Because these long-range start–goal pairs never appear in training, the critic must infer their distances from shorter trajectories. Hyperbolic geometry provides the inductive bias for this long-range structure, whereas Euclidean critics must learn it through depth. This explains the large gap at depth 4.

5.4 Performance Sensitivity across Curvature and Network Depth

The exact geometry of the hyperboloid depends on its sectional curvature $-c$, where $c = 0$ implies an Euclidean geometry. To analyze how curvature affects performance, we run a sensitivity analysis over the magnitudes $c \in \{0.01, 0.1, 0.2, 0.3, 0.5, 1\}$ across four tasks, each with five seeds. Figure 7 compares every setting against our default $c = 0.5$ by the probability that a run at that curvature outperforms a run at $c = 0.5$ (Agarwal et al., 2021). Appendix E.1 shows additional results and compares against Euclidean agents. We find that intermediate curvatures work best on mazes. On Ant U4-Maze and Ant Hardest Maze, reducing c to 0.01 drops performance to the Euclidean level, and on Ant U4-Maze, $c = 0.1$ also falls below the default curvature. At the other end, $c = 1$ performs worse than $c = 0.5$ on all four tasks. Arm Push Easy prefers a lower curvature, with $c = 0.1$ yielding the best results. Humanoid admits a different geometry, which explains our $c = 0.5$ results in Section 5.1. Here, the performance of hyperbolic agents increases monotonically as the curvature magnitude c decreases. Our flattest setting, $c = 0.01$, matches the Euclidean performance. Unsurprisingly, open arenas favor flat geometries, where route distance equals straight-line distance. In mazes, where routes bend around walls, the same flat geometry removes the hyperbolic advantage.

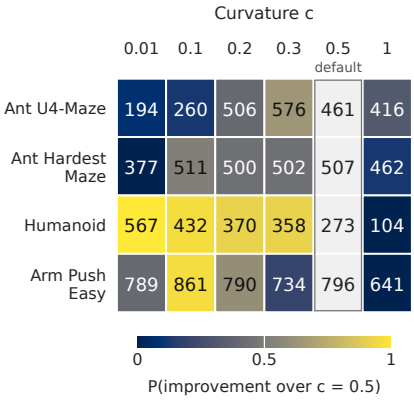


Figure 7: **Effect of curvature.** The best curvature setting depends on the environment, with Humanoid preferring low curvature, and Ant environments preferring higher curvature. Mean over the final 10 evaluation epochs over five seeds.

Hyperbolic agents benefit from a deeper Euclidean encoder on three of the four tasks and maintain their performance on Ant Ball (Figure 8). On Humanoid, hyperbolic agents improve with depth like Euclidean agents, but remain well below them from depth 16 on. The curvature ablation in Figure 7 attributes this deficit to our default curvature magnitude of $c = 0.5$. On Ant U4-Maze, the hyperbolic advantage between the two methods widens with depth, and a 4-layer hyperbolic agent already matches Euclidean agents with 16 or 32 layers. On Arm Push Easy, depth instead closes the gap, and both methods perform alike from 16 layers on. For both methods, the gains flatten as the network grows. Beyond 16 layers, Euclidean agents improve only on Humanoid. We therefore use 16 Euclidean layers for the main results in Figure 4.

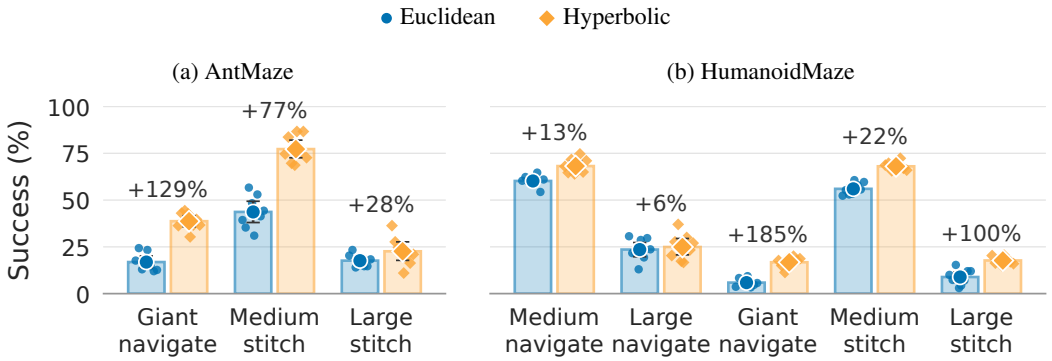


Figure 9: **Offline results on OGBench**. The hyperbolic agent reaches a higher mean success rate on all eight tasks, with the largest relative gains on the giant mazes. Percentages give the relative improvement of the hyperbolic mean over the Euclidean mean. Bars show the mean over 8 seeds with 95% bootstrap CIs.

5.5 Offline Experiments

Increasing network depth does not reliably improve offline GCRL. As Wang et al. (2025, Figure 18) scale the depth of Euclidean CRL from 4 to 64 layers, performance drops sharply on antmaze-giant-navigate and humanoidmaze-medium-navigate. On antmaze-medium-stitch, they report moderate gains. To test whether hyperbolic agents fail in the same way, we build on the CRL implementation of OGBench (Park et al., 2025) and adapt our online agent accordingly, see Appendix D.2 for all changes. Following Park et al. (2025), we use the binary success rate as an evaluation metric rather than steps at goal. Figure 9 shows that the hyperbolic agent reaches a higher mean success rate than the Euclidean agent on all eight tasks. The relative gains are largest on the two giant mazes, although neither agent reliably solves them. On navigation, the hyperbolic agent improves clearly on three of the four tasks, including antmaze-giant-navigate and humanoidmaze-medium-navigate, the two tasks on which deeper Euclidean agents fail (Wang et al., 2025, Figure 18). The hyperbolic agent also shows clear gains on three of the four stitching tasks, with overlapping confidence intervals only for antmaze-large-stitch. This matches our online stitching results in Section 5.3. The largest absolute gain is on antmaze-medium-stitch, where our hyperbolic agent exceeds 75% success, above the roughly 70% that the depth 64 residual Euclidean agent reaches in Wang et al. (2025, Figure 18). Across the three OGBench tasks that Wang et al. (2025) evaluate, depth helps on the stitching task and hurts on both navigation tasks. Hyperbolic geometry improves all three.

6 Conclusion

We introduce hyperbolic CRL, a drop-in modification to CRL that replaces the Euclidean distance in the critic with a hyperbolic geodesic distance, providing the agent with an inductive bias towards tree-like geometry at a small computational cost. With this change, depth 64 hyperbolic agents outperform 1024-layer Euclidean agents on hard humanoid maze tasks. The gains follow our geometric analysis in Section 4. They are largest where walls cause the route distance to deviate from the straight-line distance, and disappear on Humanoid, where route distances coincide with straight distances. More broadly, our results suggest that matching the representation geometry to the environment structure can alleviate much of the network depth needed for long-horizon goal reaching.

Limitations and future work. The results on Humanoid and Arm Push show that hyperbolic agents can match Euclidean performance, but with substantially lower curvature than our default of $c = 0.5$. This suggests that learning an environment-dependent curvature value improves performance. However, we found that dynamically setting or learning the curvature is challenging due to the noise in deep RL training and variability in agent performance. Second, geodesic distance is symmetric, so the critic cannot represent one-way transitions. We believe that combining hyperbolic geometry with quasimetric structure is a natural extension for solving this.

AI USE STATEMENT

In this work, we used generative AI tools for literature search, coding assistance, running experiments, writing assistance, and figure generation. In the literature search, we used AI tools to find and summarize relevant work, which was then manually checked for relevance. For code, we used AI tools for repetitive tasks, such as porting code from one deep learning framework to another and generating matplotlib figures. We reviewed our code, tested it, and verified that our baselines reproduce the results reported in the literature. For the paper, generative AI was used as a tool to iterate on our writing. An example use case is paragraph shortening with "Improve the clarity and conciseness of this paragraph." The idea to use hyperbolic representations in self-supervised RL, our analysis case study, and our ablation studies were generated and designed by us. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the representation geometry of self-supervised goal-conditioned RL agents and raises no direct ethical concerns on its own. All of our experiments are conducted on simulated locomotion and manipulation benchmarks and involve no human data. We cannot rule out downstream misuse because agents that learn to reach arbitrary goals without hand-designed rewards are a building block for autonomous robotic systems. However, such misuse would require substantial extensions beyond the simulated setting evaluated here.

REPRODUCIBILITY STATEMENT

Our code is included in the submission and will be made publicly available upon acceptance. Our environment setup is in Appendix C. Notably, our evaluation protocol differs from Wang et al. (2025) on Humanoid, which we discuss in Appendix C.3. We provide a schematic of our architecture in Appendix D. Appendices D.1 and D.2 contain hyperparameters for our online and offline experiments, respectively. Lastly, wall-clock times for our architecture on an H100 GPU are listed in Appendix F.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. 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. 29304–29320, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

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.

Ahmad Bdeir, Kristian Schwethelm, and Niels Landwehr. Fully hyperbolic convolutional neural networks for computer vision. 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=ekz1hN5QNh.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. 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=r1eiqi09K7.

Silvère Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Trans. Autom. Control.*, 58(9):2217–2229, 2013. doi: 10.1109/TAC.2013.2254619. URL https://doi.org/10.1109/TAC.2013.2254619.

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, 2025a. URL https://openreview.net/forum?id=4gaySj8kvX.

Michal Bortkiewicz, Wladyslaw Palucki, Mateusz Ostaszewski, and Benjamin Eysenbach. Is temporal difference learning the gold standard for stitching in rl? *CoRR*, abs/2510.21995, 2025b. doi: 10.48550/ARXIV.2510.21995. URL https://doi.org/10.48550/arXiv.2510.21995.

Edoardo Cetin, Benjamin Paul Chamberlain, Michael M. Bronstein, and Jonathan J. Hunt. Hyperbolic deep reinforcement learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=TfBHFLgv77.

Karan Desai, Maximilian Nickel, Tanmay Rajpurohit, Justin Johnson, and Shanmukha Ramakrishna Vedantam. Hyperbolic image-text representations. 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. 7694–7731. PMLR, 2023. URL https://proceedings.mlr.press/v202/desai23a.html.

Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7399–7409. IEEE, 2022.

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.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. 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. 5350–5360, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/dbab2adc8f9d078009ee3fa810bea142-Abstract.html.

Songwei Ge, Shlok Mishra, Simon Kornblith, Chun-Liang Li, and David Jacobs. Hyperbolic contrastive learning for visual representations beyond objects. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 6840–6849. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00661. URL https://doi.org/10.1109/CVPR52729.2023.00661.

M. Gromov. *Hyperbolic Groups*, pp. 75–263. Springer New York, New York, NY, 1987. ISBN 978-1-4613-9586-7. doi: 10.1007/978-1-4613-9586-7_3. URL https://doi.org/10.1007/978-1-4613-9586-7_3.

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.

Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan V. Oseledets, and Victor S. Lempitsky. Hyperbolic image embeddings. In *2020 IEEE/CVF Conference on Computer*

Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, pp. 6417–6427. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00645. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Khrulkov_Hyperbolic_Image_Embeddings_CVPR_2020_paper.html.

Timo Klein and Thomas Lang. Hyperbolix: Hyperbolic deep learning in jax, 2026. URL https://github.com/timoklein/hyperbolix.

Timo Klein, Thomas Lang, Andrii Shkabrii, Alexander Sturm, Kevin Sidak, Lukas Miklautz, Claudia Plant, Yllka Velaj, and Sebastian Tschiatschek. Understanding and improving hyperbolic deep reinforcement learning. In *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. OpenReview.net, 2026. URL https://openreview.net/forum?id=7rfdenlP1L.

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, 2025a. URL https://openreview.net/forum?id=jXLiDKsuDo.

Hojoon Lee, Youngdo Lee, Takuma Seno, Donghu Kim, Peter Stone, and Jaegul Choo. Hyperspherical normalization for scalable deep reinforcement learning. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025b. URL https://proceedings.mlr.press/v267/lee25u.html.

John M Lee. *Introduction to Riemannian manifolds*, volume 2. Springer, 2018.

Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric residual network for sample efficient goal-conditioned reinforcement learning. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 8799–8806. AAAI Press, 2023. doi: 10.1609/AAAI.V37I7.26058. URL https://doi.org/10.1609/aaai.v37i7.26058.

Pascal Mettes, Mina Ghadimi Atigh, Martin Keller-Ressel, Jeffrey Gu, and Serena Yeung. Hyperbolic deep learning in computer vision: A survey. *Int. J. Comput. Vis.*, 132(9):3484–3508, 2024. doi: 10.1007/S11263-024-02043-5. URL https://doi.org/10.1007/s11263-024-02043-5.

Vivek Myers, Chongyi Zheng, Anca D. Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37076–37096. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/myers24a.html.

Vivek Myers, Bill Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. 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/1c5956164472d6d8123d574aa75cd063-Abstract-Conference.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.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. 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. 6338–6347, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/59dfa2df42d9e3d41f5b02bfc32229dd-Abstract.html.

Maximilian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. 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. 3776–3785. PMLR, 2018. URL http://proceedings.mlr.press/v80/nickel18a.html.

Johan S. Obando-Ceron, Ghada Sokar, Timon Willi, Clare Lyle, Jesse Farebrother, Jakob Nicolaus Foerster, Gintare Karolina Dziugaite, Doina Precup, and Pablo Samuel Castro. Mixtures of experts unlock parameter scaling for deep RL. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38520–38540. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/obando-ceron24b.html.

Avik Pal, Max van Spengler, Guido Maria D'Amely di Melendugno, Alessandro Flaborea, Fabio Galasso, and Pascal Mettes. Compositional entailment learning for hyperbolic vision-language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=3i13Gev2hV.

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.

Frederic Sala, Christopher De Sa, Albert Gu, and Christopher Ré. Representation tradeoffs for hyperbolic embeddings. 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. 4457–4466. PMLR, 2018. URL http://proceedings.mlr.press/v80/sala18a.html.

Omar Salemohamed, Edoardo Cetin, Sai Rajeswar, and Arnab Kumar Mondal. Hyperbolic deep reinforcement learning for continuous control. In Krystal Maughan, Rosanne Liu, and Thomas F. Burns (eds.), *The First Tiny Papers Track at ICLR 2023, Tiny Papers @ ICLR 2023, Kigali, Rwanda, May 5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=Mrz9PgP3sT.

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In Marc J. van Kreveld and Bettina Speckmann (eds.), *Graph Drawing - 19th International Symposium, GD 2011, Eindhoven, The Netherlands, September 21-23, 2011, Revised Selected Papers*, volume 7034 of *Lecture Notes in Computer Science*, pp. 355–366. Springer, 2011. doi: 10.1007/978-3-642-25878-7_34. URL https://doi.org/10.1007/978-3-642-25878-7_34.

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.

Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. 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=Ec85b0tUwbA.

Didac Suris, Ruoshi Liu, and Carl Vondrick. Learning the predictability of the future. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 12607–12617. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.01242. URL https://openaccess.thecvf.com/content/CVPR2021/html/Suris_Learning_the_Predictability_of_the_Future_CVPR_2021_paper.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.

Robert van der Klis, Ricardo Chávez Torres, Max van Spengler, Yuhui Ding, Thomas Hofmann, and Pascal Mettes. Fast and geometrically grounded lorentz neural networks. *CoRR*, abs/2601.21529, 2026. doi: 10.48550/ARXIV.2601.21529. URL https://doi.org/10.48550/arXiv.2601.21529.

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 and Phillip Isola. Improved representation of asymmetrical distances with interval quasimetric embeddings. *CoRR*, abs/2211.15120, 2022. doi: 10.48550/ARXIV.2211.15120. URL https://doi.org/10.48550/arXiv.2211.15120.

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.

Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. Hypformer: Exploring efficient transformer fully in hyperbolic space. In Ricardo Baeza-Yates and Francesco Bonchi (eds.), *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pp. 3770–3781. ACM, 2024. doi: 10.1145/3637528.3672039. URL https://doi.org/10.1145/3637528.3672039.

Bill Chunyuan Zheng, Vivek Myers, Benjamin Eysenbach, and Sergey Levine. Multistep quasimetric learning for scalable goal-conditioned reinforcement learning. In *The Fourteenth International Conference on Learning Representations, ICLR 2026, Rio de Janeiro, Brazil, April 23-27, 2026*. OpenReview.net, 2026. URL https://openreview.net/forum?id=UElh7vzgKX.

Chongyi Zheng, Benjamin Eysenbach, Homer Rich Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing contrastive RL: techniques for robotic goal

reaching from offline data. 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=Xkf2EBj4w3.

A Appendix

B Extended Background on Hyperbolic Geometry

B.1 Hyperbolic Space Operations

Following the terminology of special relativity, we refer to the first coordinate x_0 of a point $\boldsymbol{x} \in \mathbb{H}^d_c$ as its *time component* and to the remaining coordinates $\boldsymbol{x}_{1:n} = (x_1, \ldots, x_n)$ as its *space component*.

B.1.1 Exponential Map

The *exponential map* $\exp_{\boldsymbol{x}} : \mathcal{T}_{\boldsymbol{x}}\mathbb{H}^d_c \to \mathbb{H}^d_c$ at a point $\boldsymbol{x} \in \mathbb{H}^d_c$ maps a tangent vector $\boldsymbol{v} \in \mathcal{T}_{\boldsymbol{x}}\mathbb{H}^d_c = \{\boldsymbol{v} : \langle \boldsymbol{v}, \boldsymbol{x} \rangle_L = 0\}$ to the manifold such that the curve $t \mapsto \exp_{\boldsymbol{x}}(t\boldsymbol{v})$ is the geodesic starting at $\boldsymbol{x}$ with initial velocity $\boldsymbol{v}$. Since the hyperboloid is geodesically complete, $\exp_{\boldsymbol{x}}$ is defined on the entire tangent space and given by

$$\exp_{\boldsymbol{x}}(\boldsymbol{v}) = \cosh\left(\sqrt{c}\,\|\boldsymbol{v}\|_L\right)\boldsymbol{x} + \sinh\left(\sqrt{c}\,\|\boldsymbol{v}\|_L\right)\frac{\boldsymbol{v}}{\sqrt{c}\,\|\boldsymbol{v}\|_L}.$$

We use the exponential map at the origin to project Euclidean input features onto the manifold. Identifying $\boldsymbol{u} \in \mathbb{R}^d$ with the tangent vector $\boldsymbol{v} = (0, \boldsymbol{u}) \in \mathcal{T}_{\bar{\mathbf{0}}}\mathbb{H}^d_c$, where $\bar{\mathbf{0}} = (1/\sqrt{c}, \mathbf{0})$, yields

$$\exp_{\bar{\mathbf{0}}}(\boldsymbol{v}) = \left(\frac{1}{\sqrt{c}}\cosh(\sqrt{c}\,\|\boldsymbol{u}\|),\ \frac{\sinh(\sqrt{c}\,\|\boldsymbol{u}\|)}{\sqrt{c}\,\|\boldsymbol{u}\|}\,\boldsymbol{u}\right).$$

B.1.2 Distance

For any two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{H}^d_c$, the *geodesic distance* $d_{\mathbb{H}^d_c}(\boldsymbol{x}, \boldsymbol{y})$ is the length of the shortest path on the manifold connecting them. In the hyperboloid model, it depends only on their Lorentzian inner product:

$$d_{\mathbb{H}^d_c}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{\sqrt{c}}\operatorname{arcosh}\left(-c\,\langle \boldsymbol{x}, \boldsymbol{y} \rangle_L\right).$$

B.1.3 Hyperbolic Linear Layers

We use the fully connected hyperbolic linear transformation of Yang et al. (2024), which operates directly on the hyperboloid without mapping to and from a tangent space. Given $\boldsymbol{x} \in \mathbb{H}^d_c$, a weight matrix $\boldsymbol{W} \in \mathbb{R}^{(d+1)\times d'}$, and a bias $\boldsymbol{b} \in \mathbb{R}^{d'}$, it first applies the Euclidean affine map $f(\boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$ to the full input, including its time-like component. The result vector is then treated as the space-like components of the output. The time-like component is determined by the hyperboloid constraint $\langle \boldsymbol{y}, \boldsymbol{y} \rangle_L = -1/c$:

$$\boldsymbol{y} = \left(\sqrt{\|f(\boldsymbol{x})\|^2 + \frac{1}{c}},\ f(\boldsymbol{x})\right) \in \mathbb{H}^{d'}_c.$$

B.2 Delta-hyperbolicity

To quantify how tree-like an embedding or environment is, we use Gromov δ-hyperbolicity, formally defined in terms of the Gromov product. For a metric space (X, d) and points $w, x, z \in X$, the Gromov product (Figure 10) of x and z with respect to w is

$$(x, z)_w = \frac{1}{2}\big(d(w, x) + d(w, z) - d(x, z)\big).$$

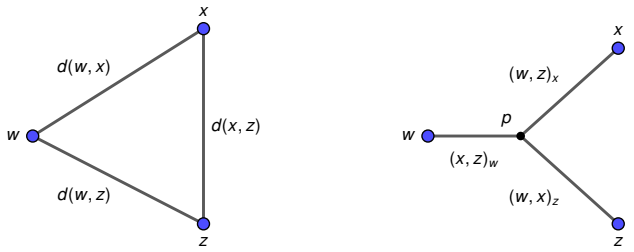


Figure 10: Geometric interpretation of the Gromov product. *Left:* three points w, x, z of a metric space, with their pairwise distances. *Right:* the associated comparison tripod, i.e. the unique metric tree with leaves w, x, z joined at a branching point p in which the path between any two leaves is of the same length as the corresponding distance on the left. The quantity $(x, z)_w = \frac{1}{2}\big(d(w, x) + d(w, z) - d(x, z)\big)$ is the length of the initial segment that shortest paths from w to x and from w to z share in the tripod before they separate at p.

A metric space is called δ-hyperbolic if, for all $x, y, z, w \in X$,

$$(x, z)_w \geq \min\{(x, y)_w, (y, z)_w\} - \delta.$$

The hyperbolicity constant of the space is the smallest $\delta \geq 0$ for which this inequality holds for all quadruples of points. Under this framework, Gromov-hyperbolic spaces can be viewed as metric spaces that are tree-like up to a bounded additive distortion. In particular, metric trees are 0-hyperbolic, while Euclidean spaces $\mathbb{R}^n$ with $n \geq 2$ are not δ-hyperbolic for any finite δ.

While δ provides an absolute measure of tree-likeness, its value cannot be compared directly across finite metric spaces of different diameters. To account for this the hyperbolicity constant is typically normalized by the diameter of the metric space

$$\delta_{\text{rel}} = \frac{2\,\delta(X)}{\operatorname{diam}(X)} \in [0, 1]$$

to provide a scale-invariant quantity. Values close to 0 indicate that deviations from tree-like geometry are small relative to the diameter of the space, and values close to 1 indicate the largest deviation possible at that scale.

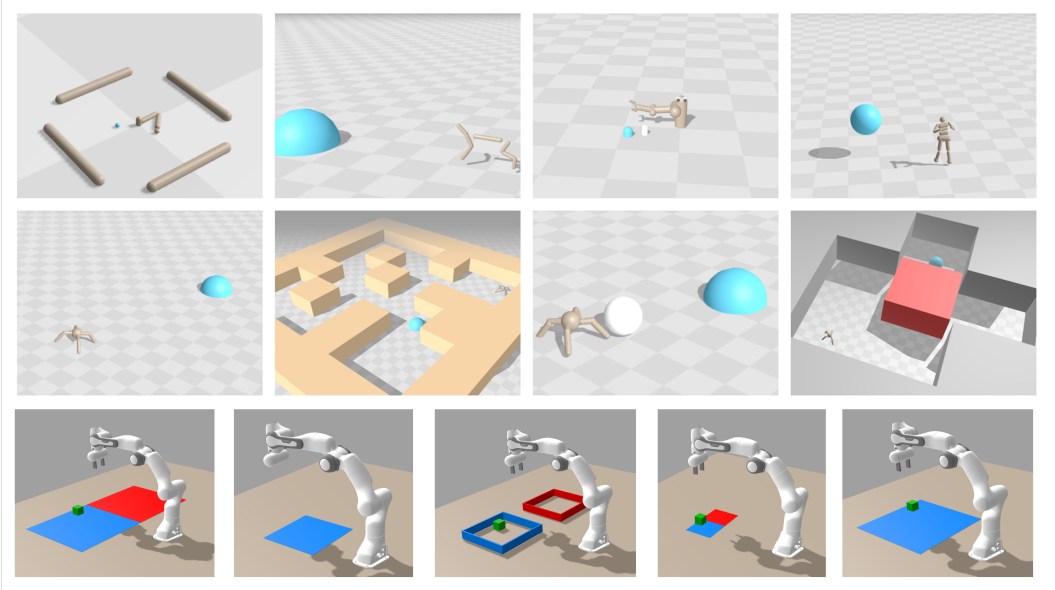


Figure 11: **Example Environments.** We use the environments implemented in JaxGCRL (Bortkiewicz et al., 2025a). Figure taken from Bortkiewicz et al. (2025a).

C Environment Setup

C.1 Online Environments

We use the online environment implementations of Bortkiewicz et al. (2025a). The full list of our 10 environments is: `arm_binpick_hard`, `arm_push_easy`, `arm_push_hard`, `humanoid`, `humanoid_big_maze`, `humanoid_u_maze`, `ant_ball`, `ant_hardest_maze`, `ant_u4_maze`, `ant_u5_maze`.

C.2 Details on the Ant U5 Maze Case Study

Measuring hyperbolicity. We quantify how tree-like the geometry of the Ant U5 maze is by evaluating its relative δ-hyperbolicity (Appendix B.2) as a function of the task horizon. Starting from its local structure (short horizons, nearby goals) we iteratively expand the environment to its global one (long horizons, far away goals). For comparison, we track both *straight-line distances* (Euclidean, ignoring walls) and *route distances* (shortest path inside the maze) across the same set of points.

To do so, we discretize the environment into a grid with a spacing of 0.5 world units, define route distances as the shortest paths between neighboring cells, and sample 300 grid points uniformly at random. From these we draw $200{,}000$ quadruples of four distinct points uniformly at random. For a horizon R, let $X_R \subseteq X$ denote the subspace of sampled points with their quadruple lying entirely in X_R. We compute their relative δ-hyperbolicity as

$$\delta_{\mathrm{rel}}(R) := \frac{2\,\delta(X_R)}{\mathrm{diam}(X_R)}.$$

A decreasing $\delta_{\mathrm{rel}}(R)$ therefore means that the environment becomes increasingly tree-like at larger task horizons. Straight-line distances are evaluated on the same subspaces and quadruples, normalized by the straight-line diameter of X_R.

As a reference, the maximum relative defect any four points in the Euclidean plane can attain is $2 - \sqrt{2} \approx 0.586$. This theoretical ceiling is reached by the vertices of a square.

Embedding distance versus route distance correlations. We combine 400 recorded states from each of the ten Euclidean and ten hyperbolic runs into a shared dataset. All critics evaluate the same

20,000 state-goal pairs, with 2,000 pairs per equal-width route-distance bin over approximately 0–51 world units. Each state's rollout action is fixed across critics and goals. For each critic k, we compute the Spearman correlation between route distance ℓ_i and embedding distance $e_{k,i} = d(\phi_k(s_i, a_i), \psi_k(g_i))$ across the same $N = 20{,}000$ state–goal pairs. Let R_i and $S_{k,i}$ denote the ranks of ℓ_i and $e_{k,i}$, respectively, assigning average ranks to ties. With $\bar{R}$ and $\bar{S}_k$ denoting their mean ranks

$$\rho_k = \frac{\sum_{i=1}^{N}(R_i - \bar{R})(S_{k,i} - \bar{S}_k)}{\sqrt{\sum_{i=1}^{N}(R_i - \bar{R})^2 \sum_{i=1}^{N}(S_{k,i} - \bar{S}_k)^2}}. \tag{3}$$

For each method, we report the mean correlation across 10 training seeds, with 95% bootstrap CIs (Agarwal et al., 2021). The probability of improvement is computed according to Agarwal et al. (2021).

Route ordering. We draw 300,000 candidate comparisons with replacement. For each comparison, we select one of the 20,000 state–goal pairs uniformly, then select a second uniformly from the same 1-world-unit straight-line-distance bin. We retain comparisons whose route lengths differ by at least 10 world units, yielding 81,007 comparisons involving 13,666 distinct state–goal pairs. All critics are evaluated on these same comparisons. We report the percentage of comparisons in which the longer route has the larger embedding distance, assigning half credit when the embedding distances are equal. Correlation and ordering results report ten-seed means and 95% bootstrap CIs from 50,000 resamples (Agarwal et al., 2021).

Gradient alignment. We collect 150 rollout states per run, with test goals comprising the maze's designated goal locations and locations visited during the collected rollouts. Actions come from each state's source policy under its original goal and remain fixed across critics and probe goals. We differentiate the critic value (negative embedding distance) with respect to torso position, holding all other inputs fixed. Cosine similarity is measured with the straight-line direction toward the goal and the direction toward the free neighboring grid node with the smallest remaining route distance. We retain state-goal pairs with a route distance of at least 24 world units and a straight-line distance below 40% of the route distance. Additionally, we require a finite and nonzero gradient. We exclude examples where no adjacent node reduces the remaining route distance. Within each seed, we average over state-goal pairs in the $[24, 48)$ and $[48, \infty)$ route-distance bins containing at least 30 valid samples. We report ten-seed means and 95% bootstrap CIs (Agarwal et al., 2021).

C.3 Humanoid Evaluation Protocol

Our Humanoid results are not directly comparable to Figure 1 of Wang et al. (2025). The discrepancy stems from how Brax stores the simulation timestep. The JaxGCRL Humanoid environment overrides the physics sub-step of its MuJoCo model from $0.003\,\mathrm{s}$ to $0.0015\,\mathrm{s}$ (Bortkiewicz et al., 2025a). Up to version 0.10.3, Brax keeps a separate copy of the timestep that is not overridden, which is invisible from the environment code. Wang et al. (2025) use Brax 0.10.1 for all their plots. This means their environment simulates at $0.003\,\mathrm{s}$, and each environment step spans $0.03\,\mathrm{s}$. Starting from version 0.10.4, Brax reads a single-timestep field. Consequently, the override takes effect, and each environment step spans $0.015\,\mathrm{s}$. The version-dependent learning curves they report are consistent with the timestep rather than with other changes in the packages. Humanoid is the only affected task because the other environments, including Humanoid U-Maze and Humanoid Big Maze, respect the timestep setting via this override. We run Humanoid at the $0.0015\,\mathrm{s}$ substep that we believe Wang et al. (2025) intended. This corresponds to the left panel of their Figure 20, and our Euclidean baseline is consistent with that curve.

D ARCHITECTURE & HYPERPARAMETERS

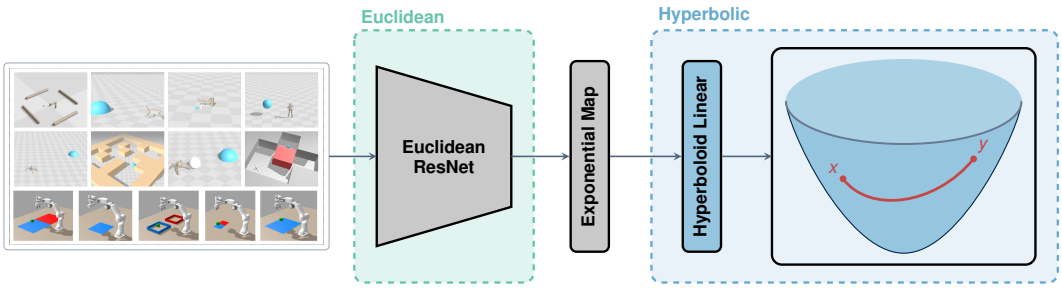


Figure 12: A Euclidean encoder maps observations to feature vectors, which are then viewed as tangent vectors at the origin of the hyperboloid, mapped to the manifold via the exponential map followed by one hyperbolic linear layer (HTC, Yang et al. (2024)). The similarity for the online goal-oriented contrastive learning is the geodesic distance between any two given points.

D.1 ONLINE CRL HYPERPARAMETERS

Table 1: **Online hyperparameters.** Unshaded rows are Wang et al. (2025, Table 7) and are shared by the Euclidean and the hybrid agents. Blue rows exist only for hyperbolic agents. Red rows differ from the released code's defaults. Total number of steps per task: ant_u4_maze, ant_ball, arm_push_easy, arm_push_hard, arm_binpick_hard 100M; ant_hardest_maze 200M; humanoid 300M; ant_u5_maze, humanoid_u_maze, humanoid_big_maze 400M.

Hyperparameter	Value
num_timesteps	100M–400M (varying across tasks)
update-to-data (UTD) ratio	1:40
max_replay_size	10,000
min_replay_size	1,000
episode_length	1,000
discounting	0.99
num_envs	512
batch_size	512
policy_lr	3e-4
critic_lr	3e-4
contrastive_loss_function	symmetric InfoNCE
energy_function	L2
logsumexp_penalty	0.1
network depth	16
network width	256
representation dimension	64
manifold	hyperboloid
curvature c	0.5
hyperbolic layers	1 (HTC (Yang et al., 2024))
hyperbolic energy_function	geodesic, $-d_c/\tau$
num_epochs (one evaluation each)	200 (humanoid: 600)
num_eval_envs	32

D.2 OFFLINE CRL HYPERPARAMETERS

We adapt our online agent in two ways for the offline setting. First, we use the Minkowski inner product as an energy function instead of the geodesic distance. This mirrors offline Euclidean agents, which also perform better with a dot-product energy than with a distance energy (Park et al., 2025). For a state-action embedding $\boldsymbol{x}$ and a goal embedding $\boldsymbol{y}$ on the hyperboloid $\mathbb{H}^d_c$, the geodesic distance $d_{\mathbb{H}^d_c}(\boldsymbol{x}, \boldsymbol{y})$ strictly decreases as the Minkowski inner product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_L$ increases (c.f. Appendix B.1). Therefore, both energies rank goals in the same order, and the critic keeps its hyperbolic geometry. Second, we use a hybrid hyperbolic actor (Salemohamed et al., 2023) and regularize it with Hyper++ (Klein et al., 2026). We hypothesize that the hyperbolic actor is stable for offline GCRL due to the absence of input nonstationarity.

Table 2: **Offline (OGBench) hyperparameters.** Unshaded rows are the OGBench reference CRL configuration Park et al. (2025) and are shared by the Euclidean and the hyperbolic agent. Blue rows exist only for the hyperbolic agent. Red rows differ from the reference.

Hyperparameter	Value
train_steps	1M
batch_size	1,024
optimizer / learning rate	Adam, 3e-4
actor MLP	(512, 512, 512), GELU, no LayerNorm
critic MLP (ϕ, ψ)	(512, 512, 512), GELU, LayerNorm
representation dimension	512
critic ensemble	2 (min over members in the actor loss)
discount	0.99 (humanoidmaze, antmaze-giant: 0.995)
contrastive loss	binary NCE
energy_function	$\phi^\top \psi / \sqrt{512}$
value goals (cur, traj, geom, rand)	(0, 0, 1, 0)
actor goals (cur, traj, geom, rand)	navigate (0, 1, 0, 0); stitch (0, 0.5, 0, 0.5)
actor loss	DDPG+BC, $\alpha = 0.1$, constant std
manifold	hyperboloid
critic curvature c	0.1
critic hyperbolic layers	1 (HTC (Yang et al., 2024)) per encoder
hyperbolic energy_function	Minkowski, $\langle \phi, \psi \rangle_{\mathcal{L}} / \tau$, $\tau = 1$
actor curvature c_π	0.5
actor hyperbolic layers	1 (HTC), width 64
eval_interval	100k steps (11 evaluations incl. step 0; score = mean of the last 3)
eval_tasks	5
eval_episodes per task	20

E Additional Results

E.1 Full Curvature Ablation

Figure 13 compares hyperbolic agents at several curvatures c against the Euclidean agent across environments, including Ant Ball. On Humanoid and Arm Push, the Euclidean agent outperforms our default $c = 0.5$ with high probability (Agarwal et al., 2021). However, on both, there exists a curvature that allows the hyperbolic agent to match the Euclidean steps-at-goal. No single curvature is best on all environments.

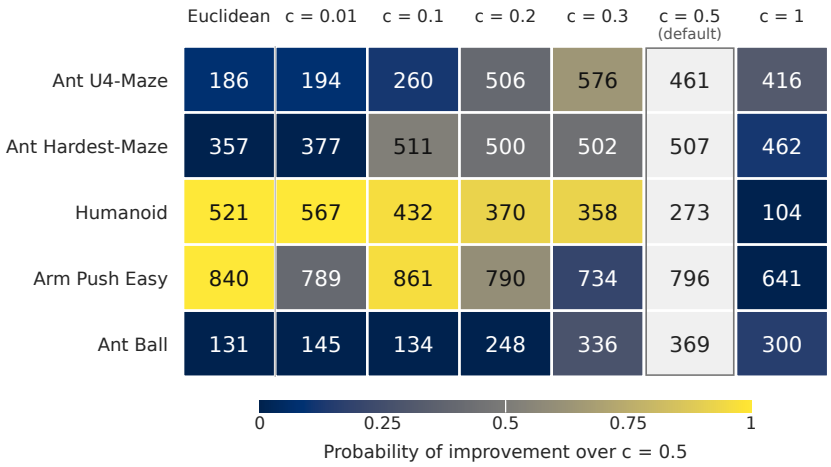


Figure 13: **Full curvature ablation.** Same Figure as Figure 7, including Ant Ball and the Euclidean agent as baseline. No single curvature setting c works best across all environments, and the best c depends on the environment. Hyperbolic agents with the curvature set per environment match Euclidean performance on Humanoid and Arm Push. Mean over the final 10 evaluation epochs over five seeds.

E.2 Behavior on Ant Ball

Figure 14 shows typical failure and success behavior on Ant Ball. The Euclidean agent (top row) reaches the ball and pushes it once in roughly the target direction, but then the ball rolls off course and comes to rest 1.5 m from the target after step 118. The agent never returns to the ball after this push. For the rest of the episode, it moves around without heading toward the ball, the target, or a pushing position, and it ends 3 m from the ball. The hyperbolic agent (bottom row) instead pushes the ball 1.0 m past the target at step 110, circles to the far side, and pushes it back. From step 264 onward, the ball stays inside the goal region. We consistently observe this pattern across 640 evaluation episodes.

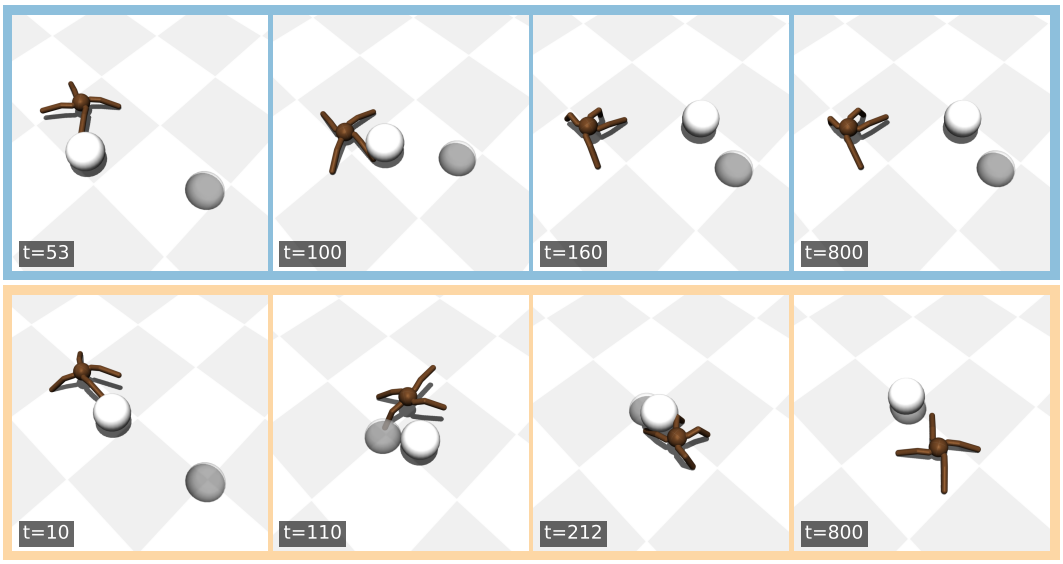
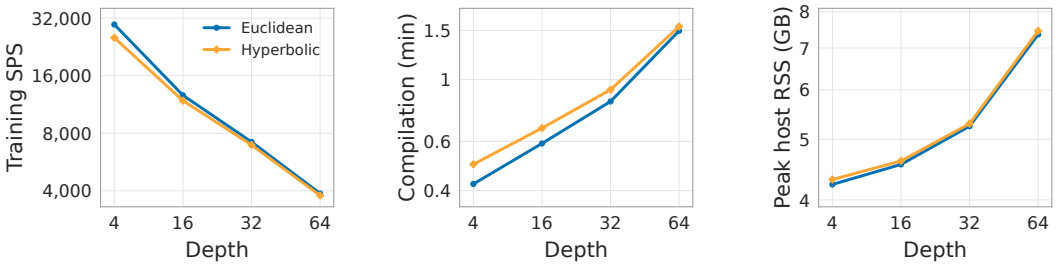


Figure 14: **Behavior on Ant Ball.**

E.3 Full Offline Results Table

Table 3: With a Euclidean actor, the hyperbolic critic reaches a higher mean success rate than Euclidean CRL on seven of eight tasks, and adding the hyperbolic actor raises it further on seven of eight. Entries are mean OGBench success rates (%) ± standard deviation over 8 seeds. **Bold** denotes hyperbolic entries whose 95% bootstrap CI does not overlap that of Euclidean CRL. The last column is the full offline hyperbolic agent, whose hybrid hyperbolic actor is regularized with Hyper++ (Klein et al., 2026).

		Hyperbolic critic	
Task	Euclidean CRL	Euclidean actor	Hyperbolic actor
AntMaze			
Giant navigate	16.9 ± 4.8	**28.0** ± 2.3	**38.7** ± 4.6
Medium stitch	43.7 ± 8.7	**80.2** ± 6.8	**77.2** ± 7.4
Large stitch	17.6 ± 3.2	21.2 ± 6.6	22.6 ± 7.5
HumanoidMaze			
Medium navigate	60.2 ± 3.0	**66.1** ± 4.2	**68.2** ± 3.9
Large navigate	23.5 ± 6.0	20.5 ± 5.4	24.9 ± 6.9
Giant navigate	5.9 ± 2.1	**10.7** ± 2.6	**16.8** ± 2.9
Medium stitch	56.0 ± 3.2	**63.9** ± 3.6	**68.0** ± 2.2
Large stitch	8.9 ± 4.2	**14.4** ± 1.4	**17.7** ± 2.0



(a) Training throughput. (b) Compilation overhead. (c) Peak host memory.

Figure 15: **Runtime on Ant U4-Maze versus network depth on one H100 GPU. 15a)** Median SPS over retained epochs. **15b)** JAX compilation time. **15c)** Peak host memory. All vertical axes are logarithmic.

F WALL-CLOCK TIMINGS

In this section, we measure training throughput, compilation overhead, and peak host memory on Ant U4-Maze, Humanoid U-Maze, and Humanoid Big-Maze. All runs use the hyperparameters specified in Appendix D.1. Throughput is measured in environment steps per second (SPS) within a training epoch. It covers environment interaction, replay sampling, parameter updates, and training diagnostics. It excludes setup, replay prefill, and evaluation. We discard the first epoch and report the median SPS over the remaining epochs. For depths up to 64, we measure nine epochs of five training steps. For deeper networks, we use four epochs of four training steps. We estimate compilation overhead as the first epoch's duration minus the median duration of later epochs, including all first-call costs, such as tracing and autotuning, not only compiler time. Peak host memory is the maximum resident set size (RSS) of the process, reported by `/usr/bin/time -v` in decimal GB.

Figure 15 reports these metrics for Ant U4-Maze, and Table 4 reports them for Humanoid U-Maze and Humanoid Big-Maze. Figures 15a and 15b show that the hyperbolic critic head lowers throughput and modestly increases compilation overhead. For SPS, the maximum overhead is 14.71% at depth 4, but this quickly drops for deeper networks. At our main evaluation setting using depth 16, hyperbolic agents achieve 6.27% lower throughput. Peak host memory is nearly unchanged, see Figure 15c. In contrast, deeper Euclidean networks are far more expensive: going from depth 64 to 512 leads to several times lower throughput and raises compilation overhead and peak host memory by more than an order of magnitude, as shown in Table 4.

For a budget of N environment steps, the estimated time in hours is

$$\hat{T}(N) = \frac{N\, t_{\text{step}}}{31{,}744 \times 3{,}600},$$

where t_{step} is the median epoch duration in seconds divided by the number of training steps per epoch, and $31{,}744$ the number of transitions collected per iteration. Thus, $\hat{T}(N)$ is the number of training steps needed to collect N transitions times the measured seconds per step, converted to hours. It excludes compilation, setup, prefill, evaluation, and checkpointing.

Table 4: **Runtime on the Humanoid mazes on one H100 GPU.** SPS are measured as median environment steps per second, excluding evaluation. We estimate the compile time as first-epoch duration minus median later-epoch duration. Peak RSS does not refer to GPU VRAM. The hyperbolic head adds 8,448 parameters irrespective of depth.

Method	Actor depth	Critic depth	Params (M)	Median SPS (↑)	Est. compile (min, ↓)	Peak RSS (GB, ↓)
Humanoid U-Maze						
Hyperbolic	64	64	12.9	3,722	1.62	7.6
Euclidean	64	64	12.9	3,788	1.54	7.6
Euclidean	512	512	102.0	512	32.52	83.2
Euclidean	512	1024	169.9	303	92.79	299.4
Humanoid Big-Maze						
Hyperbolic	64	64	12.9	3,635	1.64	7.6
Euclidean	64	64	12.9	3,686	1.54	7.6
Euclidean	512	512	102.0	510	32.44	83.1