HYPERBOLIC DEEP REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a new class of deep reinforcement learning (RL) algorithms that model latent representations in hyperbolic space. Sequential decision-making requires reasoning about the possible future consequences of current behavior. Consequently, capturing the relationship between key evolving features for a given task is conducive to recovering effective policies. To this end, hyperbolic geometry provides deep RL models with a natural basis to precisely encode this inherently hierarchical information. However, applying existing methodologies from the hyperbolic deep learning literature leads to fatal optimization instabilities due to the non-stationarity and variance characterizing RL gradient estimators. Hence, we design a new general method that counteracts such optimization challenges and enables stable end-to-end learning with deep hyperbolic representations. We empirically validate our framework by applying it to popular on-policy and off-policy RL algorithms on the Procgen and Atari 100K benchmarks, attaining near universal performance and generalization benefits. Given its natural fit, we hope future RL research will consider hyperbolic representations as a standard tool.

1 INTRODUCTION

Reinforcement Learning (RL) achieved notable milestones in several game-playing and robotics applications (Mnih et al., 2013; Vinyals et al., 2019; Kalashnikov et al., 2018; OpenAI et al., 2019; Lee et al., 2021). However, all these recent advances relied on large amounts of data and domain-specific practices, restricting their applicability in many important real-world contexts (Dulac-Arnold et al., 2019). We argue that these challenges are symptomatic of current deep RL models lacking a *proper prior* to efficiently learn *generalizable* features for control (Kirk et al., 2021). We propose to tackle this issue by introducing *hyperbolic geometry* to RL, as a new inductive bias for representation learning.

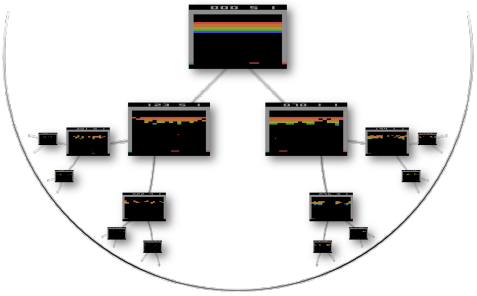

Figure 1: Hierarchical relationship between states in *breakout*, visualized in hyperbolic space.

The evolution of the state in a Markov decision process can be conceptualized as a tree, with the policy and dynamics determining the possible branches. Analogously, the same hierarchical evolution often applies to the most significant features required for decision-making (e.g., presence of bricks, location of paddle/ball in Fig. 1). These relationships tend to hold beyond individual trajectories, making hierarchy a natural basis to encode information for RL (Flet-Berliac, 2019). Consequently, we hypothesize that deep RL models should prioritize encoding precisely *hierarchically-structured* features to facilitate learning effective and generalizable policies. In contrast, we note that non-evolving features, such as the aesthetic properties of elements in the environment, are often linked with spurious correlations, hindering generalization to new states (Song et al., 2019). Similarly, human cognition also appears to learn representations of actions and elements of the environment by focusing on their underlying hierarchical relationship (Barker & Wright, 1955; Zhou et al., 2018).

Hyperbolic geometry (Beltrami, 1868; Cannon et al., 1997) provides a natural choice to efficiently encode hierarchically-structured features. A defining property of hyperbolic space is *exponential volume growth*, which enables the embedding of tree-like hierarchical data with low distortion using only a few dimensions (Sarkar, 2011). In contrast, the volume of Euclidean spaces only grows polynomially, requiring high dimensionality to precisely embed tree structures (Matoušek, 1990),

potentially leading to higher complexity, more parameters, and overfitting. We analyze the properties of learned RL representations using a measure based on the δ-*hyperbolicity* (Gromov, 1987), quantifying how close an arbitrary metric space is to a hyperbolic one. In line with our intuition, we show that performance improvements of RL algorithms correlate with the increasing *hyperbolicity* of the discrete space spanned by their latent representations. This result validates the importance of appropriately encoding hierarchical information, suggesting that the inductive bias provided by employing hyperbolic representations would facilitate recovering effective solutions.

Hyperbolic geometry has recently been exploited in other areas of machine learning showing substantial performance and efficiency benefits for learning representations of hierarchical and graph data (Nickel & Kiela, 2017; Chamberlain et al., 2017). Recent contributions further extended tools from modern deep learning to work in hyperbolic space (Ganea et al., 2018; Shimizu et al., 2020), validating their effectiveness in both supervised and unsupervised learning tasks (Khrulkov et al., 2020; Guo et al., 2022; Nagano et al., 2019; Mathieu et al., 2019; Bose et al., 2020). However, most of these approaches showed clear improvements on smaller-scale problems that failed to hold when scaling to higher-dimensional datasets and representation sizes. Many of these shortcomings are tied to the practical challenges of optimizing hyperbolic and Euclidean parameters end-to-end (Guo et al., 2022). In RL, We show the non-stationarity and high-variance characterizing common gradient estimators *exacerbates* these issues, making a naive incorporation of hyperbolic layers with existing techniques yield underwhelming results.

In this work, we overcome the aforementioned challenges and effectively train deep RL algorithms with latent hyperbolic representations end-to-end. In particular, we design *spectrally-regularized hyperbolic mappings* (S-RYM), a simple recipe combining scaling and spectral normalization (Miyato et al., 2018) that stabilizes the learned hyperbolic representations and enables their seamless integration with deep RL. We use S-RYM to build hyperbolic versions of both on-policy (Schulman et al., 2017) and off-policy algorithms (Hessel et al., 2018), and evaluate on both Procgen (Cobbe et al., 2020) and Atari 100K benchmarks (Bellemare et al., 2013). We show that our framework attains *near universal performance and generalization improvements* over established Euclidean baselines, making even general algorithms competitive with highly-tuned SotA baselines. We hope our work will set a new standard and be the first of many incorporating hyperbolic representations with RL. To this end, we share our implementation to facilitate future extensions.

2 Preliminaries

In this section, we introduce the main definitions required for the remainder of the paper. We refer to App. A and (Cannon et al., 1997) for further details about RL and hyperbolic space, respectively.

2.1 Reinforcement learning

The RL problem setting is traditionally described as a Markov Decision Process (MDP), defined by the tuple $(S, A, P, p_0, r, \gamma)$. At each timestep t, an agent interacts with the environment, observing some state from the state space $s \in S$, executing some action from its action space $a \in A$, and receiving some reward according to its reward function $r : S \times A \mapsto \mathbb{R}$. The transition dynamics $P : S \times A \times S \mapsto \mathbb{R}$ and initial state distribution $p_0 : S \mapsto \mathbb{R}$ determine the evolution of the environment's state while the discount factor $\gamma \in [0, 1)$ quantifies the agent's preference for earlier rewards. Agent behavior in RL can be represented by a parameterized distribution function π_θ, whose sequential interaction with the environment yields some trajectory $\tau = (s_0, a_0, s_1, a_1, ..., s_T, a_T)$. The agent's objective is to learn a policy maximizing its expected discounted sum of rewards over trajectories:

$$\arg\max_{\theta} \mathbb{E}_{\tau \sim \pi_\theta, P} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]. \tag{1}$$

We differentiate two main classes of RL algorithms with very different optimization procedures based on their different usage of the collected data. *On-policy* algorithms collect a new set of trajectories with the latest policy for each training iteration, discarding old data. In contrast, *off-policy* algorithms maintain a large *replay buffer* of past experiences and use it for learning useful quantities about the environment, such as world models and value functions. Two notable instances from each class are Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Rainbow DQN (Hessel et al., 2018), upon which many recent advances have been built upon.

2.2 Machine Learning in Hyperbolic Spaces

A *hyperbolic space* $\mathbb{H}^n$ is an n-dimensional Riemannian manifold with constant negative sectional curvature $-c$. Beltrami (1868) showed the equiconsistency of hyperbolic and Euclidean geometry using a model named after its re-discoverer, the *Poincaré ball model*. This model equips an n-dimensional open ball $\mathbb{B}^n = \{\mathbf{x} \in \mathbb{R}^n : c\|\mathbf{x}\| < 1\}$ of radius $1/\sqrt{c}$ with a conformal metric of the form $\mathbf{G}_\mathbf{x} = \lambda_\mathbf{x}^2 \mathbf{I}$, where $\lambda_\mathbf{x} = \frac{2}{1-c\|\mathbf{x}\|^2}$ is the *conformal factor* (we will omit the dependence on the curvature $-c$ in our definitions for notation brevity). The *geodesic* (shortest path) between two points in this metric is a circular arc perpendicular to the boundary with the length given by:

$$d(\mathbf{x}, \mathbf{y}) = \frac{1}{\sqrt{c}} \cosh^{-1}\left(1 + 2c\frac{\|\mathbf{x}-\mathbf{y}\|^2}{(1-c\|\mathbf{x}\|^2)\,(1-c\|\mathbf{y}\|^2)}\right). \tag{2}$$

From these characteristics, hyperbolic spaces can be viewed as a continuous analog of trees. In particular, the volume of a ball on $\mathbb{H}^n$ grows exponentially w.r.t. its radius. This property mirrors the exponential node growth in trees with constant branching factors. Visually, this makes geodesics between distinct points pass through some midpoint with lower magnitude, analogously to how tree geodesics between nodes (defined as the shortest path in their graph) must cross their closest shared parent (Fig. 2).

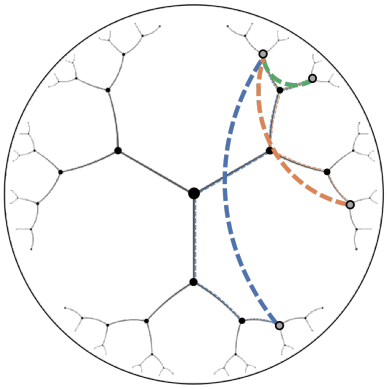

Figure 2: Geodesics on $\mathbb{H}^2$ and shortest paths connecting nodes of a tree.

Key operations for learning. On a Riemannian manifold, the *exponential map* $\exp_x(\mathbf{v})$ outputs a unit step along a geodesic starting from point x in the direction of an input velocity $\mathbf{v}$. It thus allows locally treating $\mathbb{H}^n$ as Euclidean space. We use the exponential map from the origin of the Poincaré ball to map Euclidean input vectors v into $\mathbb{H}^n$,

$$\exp_\mathbf{0}(\mathbf{v}) = \tanh\left(\sqrt{c}\|\mathbf{v}\|\right)\frac{\mathbf{v}}{\sqrt{c}\|\mathbf{v}\|}. \tag{3}$$

Following Ganea et al. (2018), we consider the framework of *gyrovector spaces* (Ungar, 2008) to extend common vector operations to non-Euclidean geometries, and in particular $\mathbb{H}^n$. The most basic such generalized operation is the *Mobius addition* $\oplus$ of two vectors,

$$\mathbf{x} \oplus \mathbf{y} = \frac{(1 + 2\mathbf{x}\langle\mathbf{x},\mathbf{y}\rangle + c\|\mathbf{y}\|^2)\mathbf{x} + (1 + c\|\mathbf{x}\|^2)\mathbf{y}}{1 + 2c\langle\mathbf{x},\mathbf{y}\rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}. \tag{4}$$

Next, consider a Euclidean affine transformation $f(\mathbf{x}) = \langle\mathbf{x},\mathbf{w}\rangle + b$ used in typical neural network layers. We can rewrite this transformation as $f(\mathbf{x}) = \langle\mathbf{x}-\mathbf{p},\mathbf{w}\rangle$ and interpret $\mathbf{w}, \mathbf{p} \in \mathbb{R}^d$ as the *normal* and *shift* parameters of a *hyperplane* $H = \{\mathbf{y} \in \mathbb{R}^d : \langle\mathbf{y}-\mathbf{p},\mathbf{w}\rangle = 0\}$ (Lebanon & Lafferty, 2004). This allows us to further rewrite $f(\mathbf{x})$ in terms of the *signed distance* to the hyperplane H, effectively acting as a weighted 'decision boundary':

$$f(\mathbf{x}) = \text{sign}\left(\langle\mathbf{x}-\mathbf{p},\mathbf{w}\rangle\right)\|\mathbf{w}\|d(\mathbf{x}, H). \tag{5}$$

This formulation allows to extend affine transformations to the Poincaré ball by considering the signed distance from a *gyroplane* in $\mathbb{B}^d$ (generalized hyperplane) $H = \{\mathbf{y} \in \mathbb{B}^d : \langle\mathbf{y}\oplus-\mathbf{p},\mathbf{w}\rangle = 0\}$,

$$f(\mathbf{x}) = \text{sign}(\langle\mathbf{x}\oplus-\mathbf{p}\mathbf{w}\rangle)\frac{2\|\mathbf{w}\|}{\sqrt{1-c\|\mathbf{p}\|^2}}d(\mathbf{x}, H);\; d(\mathbf{x}, H) = \frac{1}{\sqrt{c}}\sinh^{-1}\left(\frac{2\sqrt{c}|\langle\mathbf{x}\oplus-\mathbf{p},\mathbf{w}\rangle|}{(1-c\|\mathbf{x}\oplus-\mathbf{p}\|^2)\|\mathbf{w}\|}\right) \tag{6}$$

In line with recent use of hyperbolic geometry in supervised (Khrulkov et al., 2020; Guo et al., 2022) and unsupervised (Nagano et al., 2019; Mathieu et al., 2019) deep learning, we employ these operations to parameterize *hybrid* neural networks: we first process the input data $\mathbf{x}$ with standard layers to produce some Euclidean velocity vectors $\mathbf{x}_E = f_E(\mathbf{x})$. Then, we obtain our hyperbolic representations by applying the exponential map $\mathbf{x}_H = \exp_\mathbf{0}(\mathbf{x}_E)$. Finally, we employ affine transformations $\{f_i\}$ of the form 6 to output the set of policy and value scalars with $f_H(\mathbf{x}_H) = \{f_i(\mathbf{x}_H)\}$.

3 Hyperbolic Representations for Reinforcement Learning

In this section, we base our empirical RL analysis on Procgen (Cobbe et al., 2020). This benchmark consists of 16 visual environments, with procedurally-generated random levels. Following common practice, we *train* agents using exclusively the first 200 levels of each environment and evaluate on the full distribution of levels to assess agent performance and generalization.

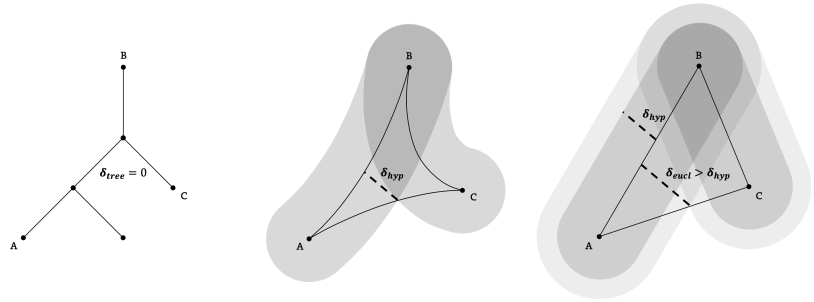

Figure 3: A geodesic space is δ-hyperbolic if every triangle is δ-slim, i.e., each of its sides is entirely contained within a δ-sized region from the other two. We illustrate the necessary δ to satisfy this property for $\triangle ABC$ in a tree triangle (**Left**), a hyperbolic triangle (**Center**) and an Euclidean triangle (**Right**); sharing vertex coordinates. In tree triangles, $\delta_{tree} = 0$ since $\overline{AC}$ always intersects both $\overline{AB}$ and $\overline{BC}$.

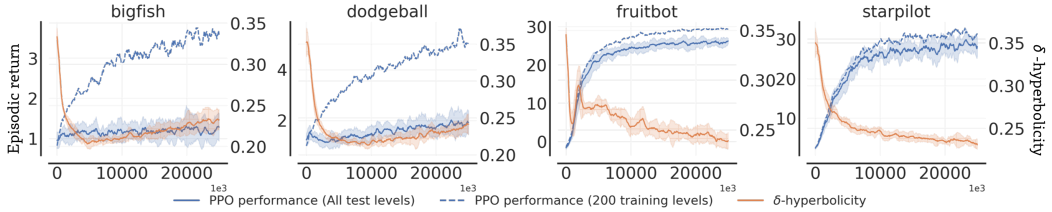

Figure 4: Performance and relative δ-hyperbolicity of the final latent representations of a PPO agent.

3.1 The inherent hyperbolicity of deep RL

Key quantities for each state, such as the value and the policy, are naturally related to its possible successors. In contrast, other fixed, non-hierarchical information about the environment such as its general appearance, can often be safely ignored. This divide becomes particularly relevant when considering the problem of RL generalization. For instance, Raileanu & Fergus (2021) found that agents' can overfit to spurious correlations between the value and non-hierarchical features (e.g., background color) in the observed states. Hence, we hypothesize that *effective representations* should encode features directly related to the hierarchical state relationships of MDPs.

δ-hyperbolicity. We analyze the representation spaces learned by RL agents, testing whether they preserve and reflect this hierarchical structure. We use the δ-hyperbolicity of a metric space (X, d) (Gromov, 1987; Bonk & Schramm, 2011), which we formally describe in App. A.2. For our use-case, X is δ-hyperbolic if every possible geodesic triangle $\triangle xyz \in X$ is δ-slim. This means that for every point on any side of $\triangle xyz$ there exists some point on one of the other sides whose distance is at most δ. In trees, *every point belongs to at least two of its sides* yielding $\delta = 0$ (Figure 3). Thus, we can interpret δ-hyperbolicity as measuring the deviation of a given metric from an exact tree metric.

The representations learned by an RL agent from encoding the collected states span some finite subset of Euclidean space $\mathbf{x}_E \in X_E \subset \mathbb{R}^n$, yielding a discrete metric space X_E. To test our hypothesis, we compute the δ-hyperbolicity of X_E and analyze how it relates to *agent performance*. Similarly to (Khrulkov et al., 2020), we compute δ using the efficient algorithm proposed by Fournier et al. (2015). To account for the scale of the representations, we normalize δ by $\text{diam}(X_E)$, yielding a *relative* hyperbolicity measure $\delta_{rel} = 2\delta/\text{diam}(X_E)$ (Borassi et al., 2015), which can span values between 0 (hyperbolic hierarchical tree-like structure) and 1 (perfectly non-hyperbolic spaces).

Results. We train an agent with PPO (Schulman et al., 2017) on four Procgen environments, encoding states from the latest rollouts using the representations before the final linear policy and value heads, $x_E = f_E(s)$. Hence, we estimate δ_{rel} from the space spanned by these latent encodings as training progresses. As shown in Figure 4, δ_{rel} quickly drops to low values $(0.22 - 0.28)$ in the first training iterations, reflecting the largest relative improvements in agent performance. Subsequently, in the *fruitbot* and *starpilot* environments, δ_{rel} further decreases throughout training as the agent recovers high performance with a low generalization gap between the training and test distribution of levels. Instead, in *bigfish* and *dodgeball*, δ_{rel} begins to increase again after the initial drop, suggesting that the latent representation space starts losing its hierarchical structure. Correspondingly, the agent starts overfitting as test levels performance stagnates while the generalization gap with the training levels performance keeps increasing. These results validate our hypothesis, empirically showing the importance of encoding hierarchical features for recovering effective solutions. Fur-

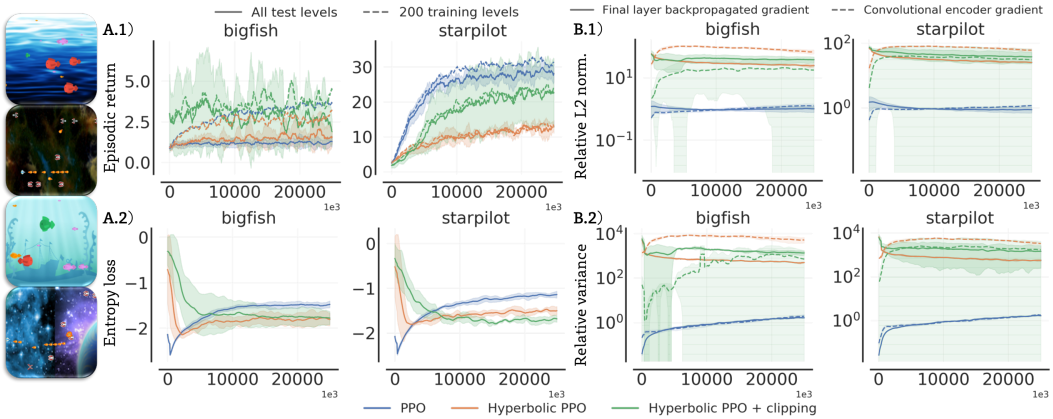

Figure 6: Analysis of key statistics for our *naive* implementations of hyperbolic PPO agents using existing practices to stabilize optimization in hyperbolic space. On the left, we display performance (**A.1**) and negative entropy (**A.2**). On the right, we display magnitudes (**B.1**) and variances (**B.2**) of the backpropagated gradients.

thermore, they suggest that PPO's poor generalization in some environments is due to the observed tendency of the Euclidean latent space to encode spurious features that hinder its hyperbolicity.

Motivated by our findings, we propose employing hyperbolic geometry to model the latent representations of deep RL models. Representing tree-metrics in Euclidean spaces incurs non-trivial worse-case distortions, growing with the number of nodes at a rate dependent on the dimensionality (Matoušek, 1990). This property suggests that it is not possible to encode complex hierarchies in Euclidean space both efficiently and accurately, explaining why some solutions learned by PPO could not maintain their hyperbolicity throughout training. In contrast, mapping the latent representations to hyperbolic spaces of any dimensionality enables encoding features exhibiting a tree-structured relation over the data with *arbitrarily low distortion* (Sarkar, 2011). Hence, hyperbolic latent representations introduce a different *inductive bias* for modeling the policy and value function, stemming from this inherent efficiency of specifically encoding hierarchical information (Tifrea et al., 2018).

3.2 Optimization Challenges

Naive integration. We test a simple extension to PPO, mapping the latent representations of states $s \in S$ before the final linear policy and value heads $x_E = f_E(s)$ to the Poincaré ball with unitary curvature. As described in Section 2, we perform this with an exponential map to produce $x_H = \exp_0^1(x_E)$, replacing the final ReLU. To output the value and policy logits, we then finally perform a set of affine transformations in hyperbolic space, $\pi(s), V(s) = f_H(x_H) = \{f_i^1(x_H)\}_{i=0}^{|A|}$. We also consider a *clipped* version of this integration, following the recent stabilization practice from Guo et al. (2022), which entails clipping the magnitude of the latent representations to not exceed unit norm. We initialize the weights of the last two linear layers in both implementations to $100\times$ smaller values to start training with low magnitude latent representations, which facilitates the network first learning appropriate angular layouts (Nickel & Kiela, 2017; Ganea et al., 2018).

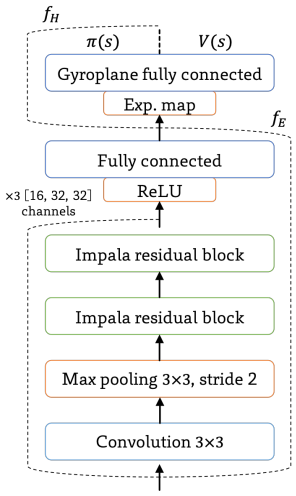

Figure 5: PPO model with an hyperbolic latent space, extending the architecture from Espeholt et al. (2018).

Results. We analyze this naive hyperbolic PPO implementation in Figure 6. As shown in part (A.1), performance is generally underwhelming, lagging considerably behind the performance of standard PPO. While applying the clipping strategy yields some improvements, its results are still considerably inferior on the tasks where Euclidean embeddings appear to already recover effective representations (e.g. *starpilot*). In part (A.2) we visualize the negated entropy of the different PPO agents. PPO's policy optimization objective includes both a reward maximization term, which requires an auxiliary estimator, and an entropy bonus term that can instead be differentiated exactly and optimized end-to-end. Its purpose

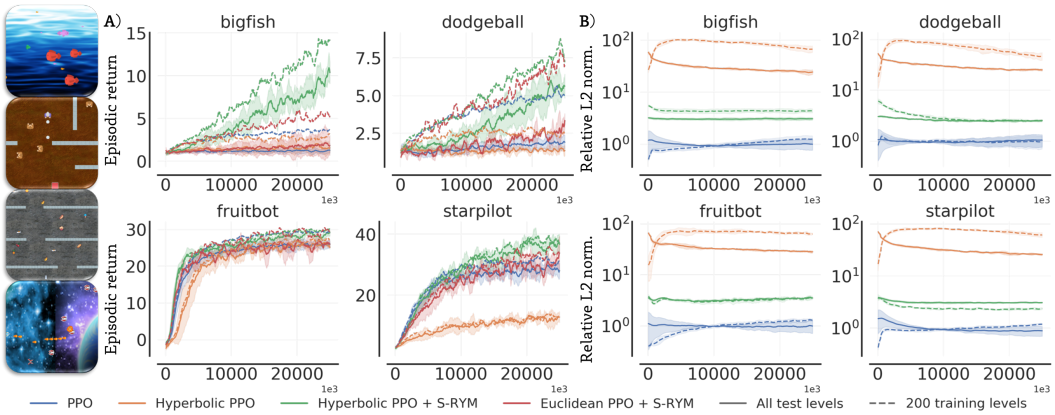

Figure 7: Analysis of hyperbolic PPO with the proposed S-RYM stabilization. We visualize performance (**A**) and gradient magnitudes (**B**) as compared to the original Euclidean and the naive hyperbolic baselines.

is to push PPO agents to explore if they struggle to optimize performance with the current data. We note that the Hyperbolic PPO agents take significantly longer to reach higher levels of entropy in the initial training phases and are also much slower to reduce their entropy as their performance improves. These results appear to indicate the presence of optimization challenges stemming from end-to-end RL training with hyperbolic representations. Therefore, we turn our attention to analyzing the *gradients* in our hyperbolic models. In part (B.1), we visualize the magnitude of the gradients both as backpropagated from the final representations and to the convolutional encoder. In part (B.2), we also visualize the variance of the same gradients with respect to the different input states in a minibatch. We find that hyperbolic PPO suffers from a severe exploding gradients problem, with both magnitudes and variances being several orders of magnitude larger than the Euclidean baseline. Similar instabilities have been documented by much recent literature, as described in App. B. Yet, in the RL case, common stabilization techniques such as careful initialization and clipping are visibly insufficient, resulting in ineffective learning and inferior agent performance.

3.3 Stabilizing hyperbolic representations

We attribute the observed optimization challenges from our naive hyperbolic PPO implementation to the high variance and non-stationarity characterizing RL. Initialization and clipping have been designed for stationary ML applications with *fixed* dataset and targets. In these settings, regularizing the initial learning iterations enables the model to find appropriate angular layouts of the representations for the underlying *fixed* loss landscape. Without appropriate angular layouts, useful representations become very hard to recover due to the highly non-convex spectrum of hyperbolic neural networks, often resulting in failure modes with low performance (Ganea et al., 2018; López & Strube, 2020). We can intuitively see why this reliance is incompatible with the RL setting, where the trajectory data and loss landscape can change significantly throughout training, making early angular layouts inevitably suboptimal. This is further exacerbated by the high variance gradients already characterizing policy gradient optimization (Sutton & Barto, 2018; Wu et al., 2018) which facilitate entering unstable learning regimes that can lead to our observed failure modes.

Spectral norm. Another sub-field of ML dealing with non-stationarity and brittle optimization is generative modeling with adversarial networks (GANs) (Goodfellow et al., 2014). In GAN training, the generated data and discriminator's parameters constantly evolve, making the loss landscape highly non-stationary as in the RL setting. Furthermore, the adversarial nature of the optimization makes it very brittle to exploding and vanishing gradients instabilities which lead to common failure modes (Arjovsky & Bottou, 2017; Brock et al., 2018). In this parallel literature, *spectral normalization (SN)* (Miyato et al., 2018) is a popular stabilization practice whose success made it ubiquitous in modern GAN implementations. Recent work (Lin et al., 2021) showed that a reason for its surprising effectiveness comes from *regulating* both the magnitude of the activations and their respective gradients very similarly to LeCun initialization (LeCun et al., 2012). Furthermore, when applied to the discriminator model, SN's effects appear to persist *throughout training*, while initialization strategies tend to only affect the initial iterations. In fact, they also show that ablating SN from GAN training empirically results in exploding gradients and degraded performance, closely resembling our same observed instabilities. We provide details about GANs and SN in App. A.3.

Table 1: Performance comparison for the considered versions of PPO full Procgen benchmark

Task\Algorithm	PPO	PPO + data aug.	PPO + S-RYM	PPO + S-RYM, 32 dim.
Levels distribution	*train/test*	*train/test*	*train/test*	*train/test*
bigfish	3.71±1 1.46±1	12.43±4 (+235%) 13.07±2 (+797%)	13.27±2 (+258%) 12.20±2 (+737%)	20.58±5 (+455%) **16.57±2 (+1037%)**
bossfight	8.18±1 7.04±2	3.38±1 (-59%) 2.96±1 (-58%)	8.61±1 (+5%) 8.14±1 (+16%)	9.26±1 (+13%) **9.02±1 (+28%)**
caveflyer	7.01±1 **5.86±1**	6.08±1 (-13%) 4.89±1 (-16%)	6.15±1 (-12%) 5.15±1 (-12%)	6.38±1 (-9%) 5.20±1 (-11%)
chaser	6.58±2 5.89±1	2.14±0 (-67%) 2.18±0 (-63%)	6.60±2 (+0%) 7.82±1 (+33%)	9.04±1 (+37%) **7.32±1 (+24%)**
climber	8.66±2 5.11±1	7.61±1 (-12%) 5.74±2 (+12%)	8.91±1 (+3%) 6.64±1 (+30%)	8.32±1 (-4%) **7.28±1 (+43%)**
coinrun	9.50±0 8.25±0	8.40±1 (-12%) 9.00±1 (+9%)	9.30±1 (-2%) 8.40±0 (+2%)	9.70±0 (+2%) **9.20±0 (+12%)**
dodgeball	5.07±1 1.87±1	3.94±1 (-22%) 3.20±1 (+71%)	7.10±1 (+40%) 6.52±1 (+248%)	7.74±2 (+53%) **7.14±1 (+281%)**
fruitbot	30.10±2 26.33±2	27.56±3 (-8%) 27.98±1 (+6%)	30.43±1 (+1%) 27.97±3 (+6%)	29.15±1 (-3%) **29.51±1 (+12%)**
heist	7.42±1 2.92±1	4.20±1 (-43%) **3.60±0 (+23%)**	5.40±1 (-27%) 2.70±1 (-7%)	6.40±1 (-14%) **3.60±1 (+23%)**
jumper	8.86±1 6.14±1	7.70±1 (-13%) 5.70±0 (-7%)	9.00±1 (+2%) **6.70±1 (+9%)**	8.50±0 (-4%) 6.10±1 (-1%)
leaper	4.86±2 4.36±2	6.80±1 (+40%) 7.00±1 (+61%)	8.00±1 (+65%) **7.30±1 (+68%)**	7.70±1 (+59%) 7.00±1 (+61%)
maze	9.25±0 6.50±0	8.50±1 (-8%) **7.10±1 (+9%)**	9.50±0 (+3%) 6.10±1 (-6%)	9.20±0 (-1%) **7.10±1 (+9%)**
miner	12.95±0 9.28±1	9.81±0 (-24%) 9.36±2 (+1%)	12.09±1 (-7%) 10.08±1 (+9%)	12.94±0 (+0%) **9.86±1 (+6%)**
ninja	7.62±1 **6.50±1**	6.90±1 (-10%) 4.50±1 (-31%)	6.50±1 (-15%) 6.10±1 (-6%)	7.50±1 (-2%) 5.60±1 (-14%)
plunder	6.92±2 6.06±3	5.13±0 (-26%) 4.96±1 (-18%)	7.26±1 (+5%) 6.87±1 (+13%)	7.35±1 (+6%) 6.68±0 (+10%)
starpilot	30.50±5 26.57±5	43.43±7 (+42%) 32.41±3 (+22%)	37.08±3 (+22%) 41.22±3 (+55%)	41.48±4 (+36%) **38.27±5 (+44%)**
Average norm. score	0.5614 0.3476	0.4451 (-21%) 0.3536 (+2%)	0.5846 (+4%) 0.4490 (+29%)	**0.6326 (+13%) 0.4730 (+36%)**
Median norm. score	0.6085 0.3457	0.5262 (-14%) 0.3312 (-4%)	0.6055 (+0%) 0.4832 (+40%)	**0.6527 (+7%) 0.4705 (+36%)**
# Env. improvements	0/16 0/16	3/16 10/16	**11/16** 12/16	8/16 **13/16**

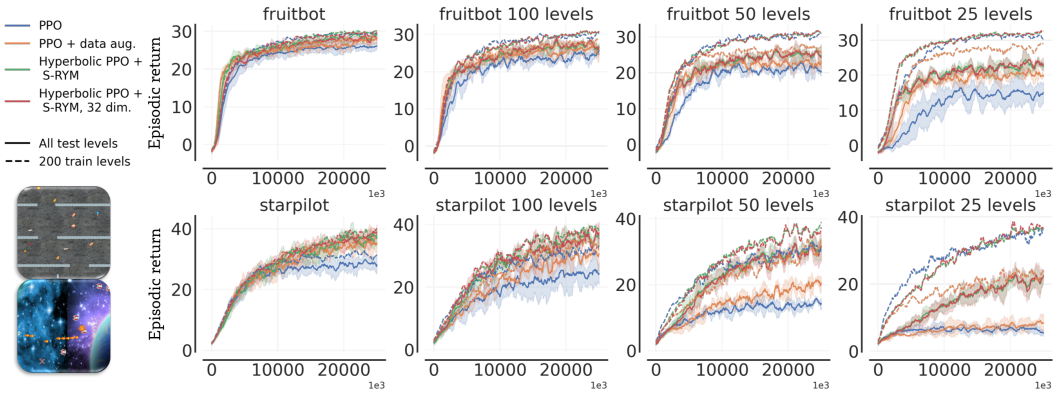

Figure 8: Performance comparison for the considered versions of PPO agents with Euclidean and hyperbolic latent representations, increasingly lowering the number of training levels.

S-RYM. Inspired by these connections, we propose to counteract the optimization challenges in RL and hyperbolic representations with SN. We make two main changes from its usual application for GAN regularization. First, we apply SN only in the Euclidean encoder sub-network (f_E), leaving the final linear transformation in hyperbolic space (f_H) unregularized since our instabilities appear to occur in the gradients *from* the hyperbolic representations. Furthermore, we add a scaling term to preserve stability for different latent representation sizes. In particular, modeling $x_E \in \mathbb{R}^n$ by an independent Gaussian, the magnitude of the representations follows some scaled *Chi distribution* $\|x_E\| \sim \chi_n$, which we can reasonably approximate with $E[\|x_E\|] = E[\chi_n] \approx \sqrt{n}$. Therefore, we propose to rescale the output of f_E by $1/\sqrt{n}$, such that modifying the dimensionality of the representations should not significantly affect their magnitude before mapping them $\mathbb{H}^n$. We call this general stabilization recipe *spectrally-regularized hyperbolic mappings* (S-RYM).

Results. As shown in Figure 7, integrating S-RYM with our hyperbolic RL agents appears to resolve their optimization challenges and considerably improve the Euclidean baseline's performance (A). To validate that these performance benefits are due to the hyperbolic geometry of the latent space, we also compare with another Euclidean ablation making use of SN, which fails to attain any improvement. Furthermore, S-RYM maintains low gradient magnitudes (B), confirming its effectiveness to stabilize training. In App. E.1, we also show that SN and rescaling are both crucial for S-RYM. Thus, in the next section we evaluate our hyperbolic deep RL framework on a large-scale, analyzing its efficacy and behavior across different benchmarks, RL algorithms, and training conditions.

4 Extensions and Evaluation

To test the generality of our hyperbolic deep RL framework, in addition to the on-policy PPO we also integrate it with the off-policy Rainbow DQN algorithm (Hessel et al., 2018). Our implementations use the same parameters and models specified in prior traditional RL literature, without any additional tuning. Furthermore, in addition to the full Procgen benchmark (16 envs.) we also evaluate on

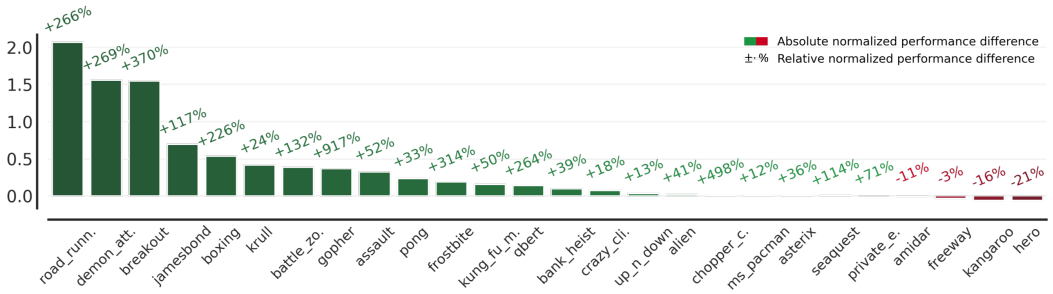

Figure 9: Absolute difference in normalized performance (**Y-axis**) and relative improvements (**Above bars**) from integrating hyperbolic representations with S-RYM onto our Rainbow implementation.

the popular Atari 100K benchmark (Bellemare et al., 2013; Kaiser et al., 2020) (26 envs.), repeating for 5 random seeds. We provide all details about benchmarks and implementations in App. C.

Generalization on Procgen. Given the documented representation efficiency of hyperbolic space, we evaluate our hyperbolic PPO implementation also reducing the dimensionality of the final representation to 32 (see App. E.2), with relative compute and parameter efficiency benefits. We compare our regularized hyperbolic PPO with using data augmentations, a more traditional way of encoding inductive biases from inducing invariances. We consider random crop augmentations from their popularity and success in modern RL. As shown in Table 1, our hyperbolic PPO implementation with S-RYM appears to yield conspicuous performance gains on most of the environments. At the same time, reducing the size of the representations provides even further benefits with significant improvements in 13/16 tasks. In contrast, applying data augmentations yields much lower and inconsistent gains, even hurting on some tasks where hyperbolic RL provides considerable improvements (e.g. *bossfight*). We also find that test performance gains do not always correlate with gains on the specific 200 training levels, yielding a significantly reduced generalization gap for the hyperbolic agents. We perform the same experiment but apply our hyperbolic deep RL framework to Rainbow DQN with similar results, also obtaining significant gains in 13/16 tasks, as reported in App. D.1.

We also evaluate the robustness of our PPO agents to encoding *spurious* features, only relevant for the training levels. In particular, we examine tasks where PPO tends to perform well and consider lowering the training levels from 200 to 100, 50, and 25. As shown in Figure 8, the performance of PPO visibly drops at each step halving the number of training levels, suggesting that the Euclidean representations overfit and lose their original efficacy. In contrast, hyperbolic PPO appears much more robust, still surpassing the original PPO results with only 100 training levels in *fruitbot* and 50 in *starpilot*. While also applying data augmentation attenuates the performance drops, its effects appear more limited and inconsistent, providing almost null improvements for *starpilot*.

Sample-efficiency on Atari 100K. We focus on the performance of our hyperbolic Rainbow DQN implementation, as the severe data limitations of this benchmark make PPO and other on-policy algorithms impractical. We show the absolute and relative per-environment performance changes from our hyperbolic RL framework in Figure 9, and provide aggregate statistics in Table 2. Also on this benchmark, the exact same hyperbolic deep RL framework provides consistent and significant benefits. In particular, we record improvements on 22/26 Atari environments over the Euclidean baseline, almost doubling the final human normalized score.

Table 2: Aggregate results on Atari 100K

Metric\Algorithm	Rainbow	Rainbow + S-RYM
Human norm. mean	0.353	**0.686** (**+93%**)
Human norm. median	0.259	**0.366** (**+41%**)
Super human scores	2	**5**

Considerations and comparisons. Our results empirically validate that introducing hyperbolic representations to shape the prior of deep RL models is both remarkably general and effective. We record almost universal improvements on two fundamentally different RL algorithms, considering both generalizations to new levels from millions of frames (Procgen) and to new experiences from only 2hrs of total play time (Atari 100K). Furthermore, our hyperbolic RL agents outperform the scores reported in most other recent advances, coming very close to the current SotA algorithms which incorporate different expensive and domain-specialized auxiliary practices (see App. D.2-D.3). Our approach is also orthogonal to many of these advances and appears to provide compatible and complementary benefits (see App. E.3). Taken together, we believe these factors show the great potential of our hyperbolic framework to become a standard way of parameterizing deep RL models.

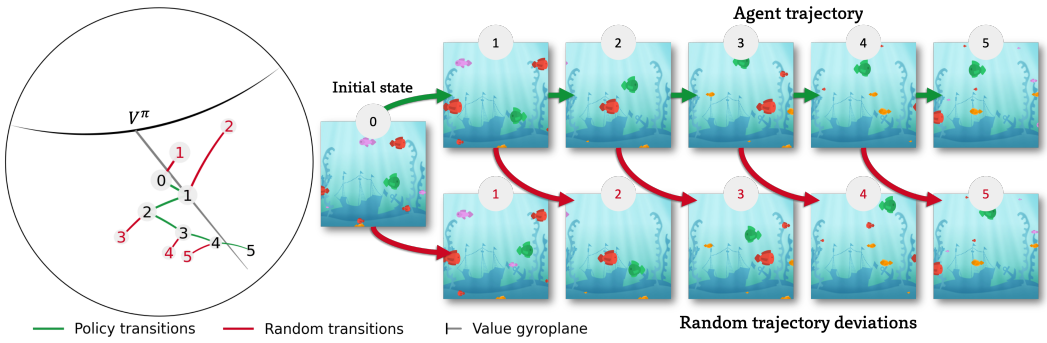

Figure 10: Visualization of 2-dimensional hyperbolic embeddings in the *bigfish* environment as we progress through a trajectory, encoding states from either policy transitions or random transitions (details in App. D.4).

Representations interpretation. We train our hyperbolic PPO agent with only *2-dimensional* representations, which still remarkably provide concrete generalization benefits over Euclidean PPO (App. D.4). Then, we analyze how these representations evolve within trajectories, mapping them on the Poincaré disk and visualizing the corresponding states. We observe a recurring *cyclical* behavior, where the magnitude of the representations monotonically increases within subsets of the trajectory as more obstacles/enemies appear. We show this in Fig. 10 and Fig. 12, comparing the representations of on-policy states sampled at constant intervals with trajectory deviations from executing random behavior. We observe the representations form tree-like structures, with the magnitudes in the on-policy states growing in the direction of the Value function's *gyroplane*'s normal. This intuitively reflects that as new elements appear the agent recognizes a larger opportunity for rewards, yet, requiring a finer level of control as distances to the policy gyroplanes will also grow exponentially, reducing entropy. Instead, following random deviations, magnitudes grow in directions orthogonal to the Value gyroplane's normal. This still reflects the higher precision required for optimal decision-making, but also the higher uncertainty to obtain future rewards from worse states.

5 Related Work

Generalization is a key open problem in RL (Kirk et al., 2021). End-to-end training of deep models with RL objectives appears has been shown prone to overfitting from spurious features only relevant in the observed transitions (Song et al., 2019; Bertran et al., 2020). To address this, prior work considered different data augmentation strategies (Laskin et al., 2020b; Yarats et al., 2021a; Cobbe et al., 2019), and online adaption methods on top to alleviate engineering burdens (Zhang & Guo, 2021; Raileanu et al., 2020). Alternative approaches have been considering problem-specific properties of the environment (Zhang et al., 2020; Raileanu & Fergus, 2021), auxiliary losses (Laskin et al., 2020a; Schwarzer et al., 2020), and frozen pre-trained layers (Yarats et al., 2021b; Stooke et al., 2021). Instead, we propose to encode a new inductive bias making use of the geometric properties of hyperbolic space, something orthogonal and likely compatible with most such prior methods.

While hyperbolic representations found recent popularity in machine learning, there have not been notable extensions for deep RL (Peng et al., 2021). Most relatedly, Tiwari & Prannoy (2018) proposed to produce hyperbolic embeddings of the state space of tabular MDPs to recover options (Sutton et al., 1999). Yet, they did not use RL for learning, but fixed data and a supervised loss based on the co-occurrence of states, similarly to the original method by Nickel & Kiela (2017).

6 Discussion and Future Work

In this work, we introduce hyperbolic geometry to deep RL. We analyze training agents using latent hyperbolic representations and propose *spectrally-regularized hyperbolic mappings*, a new stabilization strategy that overcomes the observed optimization instabilities. Hence, we apply our framework to obtain hyperbolic versions of established on-policy and off-policy RL algorithms, which we show substantially outperform their Euclidean counterparts in two popular benchmarks. We provide numerous results validating that hyperbolic representations provide deep models with a more suitable prior for control, with considerable benefits for generalization and sample-efficiency. We share our implementation to facilitate future RL advances considering hyperbolic space as a new, general tool.

REFERENCES

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gans. *arXiv preprint arXiv:1701.07875*, 2017.

Roger G Barker and Herbert F Wright. Midwest and its children: The psychological ecology of an american town. 1955.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian adaptive optimization methods. *arXiv preprint arXiv:1810.00760*, 2018.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*, pp. 449–458. PMLR, 2017.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, pp. 679–684, 1957.

Eugenio Beltrami. Teoria fondamentale degli spazii di curvatura costante. *Annali di Matematica Pura ed Applicata (1867-1897)*, 2(1):232–255, 1868.

Martin Bertran, Natalia Martinez, Mariano Phielipp, and Guillermo Sapiro. Instance-based generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 11333–11344, 2020.

Johan Bjorck, Carla P Gomes, and Kilian Q Weinberger. Towards deeper deep reinforcement learning. *arXiv preprint arXiv:2106.01151*, 2021.

Mario Bonk and Oded Schramm. Embeddings of gromov hyperbolic spaces. In *Selected Works of Oded Schramm*, pp. 243–284. Springer, 2011.

Silvere Bonnabel. Stochastic gradient descent on riemannian manifolds. *IEEE Transactions on Automatic Control*, 58(9):2217–2229, 2013.

Michele Borassi, Alessandro Chessa, and Guido Caldarelli. Hyperbolicity measures democracy in real-world networks. *Physical Review E*, 92(3):032812, 2015.

Joey Bose, Ariella Smofsky, Renjie Liao, Prakash Panangaden, and Will Hamilton. Latent variable modelling with hyperbolic normalizing flows. In *International Conference on Machine Learning*, pp. 1045–1055. PMLR, 2020.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

James W Cannon, William J Floyd, Richard Kenyon, Walter R Parry, et al. Hyperbolic geometry. *Flavors of geometry*, 31(59-115):2, 1997.

Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G. Bellemare. Dopamine: A Research Framework for Deep Reinforcement Learning. 2018. URL http://arxiv.org/abs/1812.06110.

Edoardo Cetin, Philip J Ball, Stephen Roberts, and Oya Celiktutan. Stabilizing off-policy deep reinforcement learning from pixels. In *International Conference on Machine Learning*, pp. 2784–2810. PMLR, 2022.

Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. Neural embeddings of graphs in hyperbolic space. *arXiv preprint arXiv:1705.10359*, 2017.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 1282–1289. PMLR, 2019.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pp. 2048–2056. PMLR, 2020.

Karl W Cobbe, Jacob Hilton, Oleg Klimov, and John Schulman. Phasic policy gradient. In *International Conference on Machine Learning*, pp. 2020–2027. PMLR, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pp. 1407–1416. PMLR, 2018.

Yannis Flet-Berliac. The promise of hierarchical reinforcement learning. *The Gradient*, 9, 2019.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Matteo Hessel, Ian Osband, Alex Graves, Volodymyr Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rywHCPkAW.

Hervé Fournier, Anas Ismail, and Antoine Vigneron. Computing the gromov hyperbolicity of a discrete metric space. *Information Processing Letters*, 115(6-8):576–579, 2015.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

Florin Gogianu, Tudor Berariu, Mihaela C Rosca, Claudia Clopath, Lucian Busoniu, and Razvan Pascanu. Spectral normalisation for deep reinforcement learning: an optimisation perspective. In *International Conference on Machine Learning*, pp. 3734–3744. PMLR, 2021.

Gene H Golub and Henk A Van der Vorst. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1-2):35–65, 2000.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, pp. 75–263. Springer, 1987.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Yunhui Guo, Xudong Wang, Yubei Chen, and Stella X Yu. Clipped hyperbolic classifiers are super-hyperbolic classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11–20, 2022.

Anupam Gupta. Embedding tree metrics into low dimensional euclidean spaces. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pp. 694–700, 1999.

Hado Hasselt. Double q-learning. *Advances in neural information processing systems*, 23:2613–2621, 2010.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016b.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Riashat Islam, Peter Henderson, Maziar Gomrokchi, and Doina Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *arXiv preprint arXiv:1708.04133*, 2017.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning*, pp. 4940–4950. PMLR, 2021.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for Atari. In *International Conference on Learning Representations*, 2020.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6418–6428, 2020.

Kacper Piotr Kielak. Do recent advancements in model-based deep reinforcement learning really improve data efficiency? 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of generalisation in deep reinforcement learning. *arXiv preprint arXiv:2111.09794*, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. pp. 5639–5650, 2020a.

Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33: 19884–19895, 2020b.

Guy Lebanon and John Lafferty. Hyperplane margin classifiers on the multinomial manifold. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 66, 2004.

Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–48. Springer, 2012.

Alex X Lee, Coline Manon Devin, Yuxiang Zhou, Thomas Lampe, Konstantinos Bousmalis, Jost Tobias Springenberg, Arunkumar Byravan, Abbas Abdolmaleki, Nimrod Gileadi, David Khosid, et al. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. In *5th Annual Conference on Robot Learning*, 2021.

Zinan Lin, Vyas Sekar, and Giulia Fanti. Why spectral normalization stabilizes gans: Analysis and improvements. *Advances in neural information processing systems*, 34:9625–9638, 2021.

Federico López and Michael Strube. A fully hyperbolic neural model for hierarchical multi-class classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 460–475, 2020.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

Shie Mannor, Duncan Simester, Peng Sun, and John N Tsitsiklis. Bias and variance approximation in value function estimates. *Management Science*, 53(2):308–322, 2007.

Emile Mathieu, Charline Le Lan, Chris J Maddison, Ryota Tomioka, and Yee Whye Teh. Continuous hierarchical representations with poincaré variational auto-encoders. *Advances in neural information processing systems*, 32, 2019.

Jiří Matoušek. Bi-lipschitz embeddings into low-dimensional euclidean spaces. *Commentationes Mathematicae Universitatis Carolinae*, 31(3):589–600, 1990.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Sharada Mohanty, Jyotish Poonganam, Adrien Gaidon, Andrey Kolobov, Blake Wulfe, Dipam Chakraborty, Grazvydas Semetulskis, Joao Schapke, Jonas Kubilius, Jurgis Pasukonis, et al. Measuring sample efficiency and generalization in reinforcement learning benchmarks: Neurips 2020 procgen benchmark. *arXiv preprint arXiv:2103.15332*, 2021.

Yoshihiro Nagano, Shoichiro Yamaguchi, Yasuhiro Fujita, and Masanori Koyama. A wrapped normal distribution on hyperbolic space for gradient-based learning. In *International Conference on Machine Learning*, pp. 4693–4702. PMLR, 2019.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

OpenAI, Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, Jonas Schneider, Nikolas Tezak, Jerry Tworek, Peter Welinder, Lilian Weng, Qiming Yuan, Wojciech Zaremba, and Lei Zhang. Solving rubik's cube with a robot hand. *CoRR*, abs/1910.07113, 2019.

Mark D Pendrith, Malcolm RK Ryan, et al. *Estimator variance in reinforcement learning: Theoretical problems and practical solutions*. University of New South Wales, School of Computer Science and Engineering, 1997.

Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Roberta Raileanu and Rob Fergus. Decoupling value and policy for generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 8787–8798. PMLR, 2021.

Roberta Raileanu, Max Goldstein, Denis Yarats, Ilya Kostrikov, and Rob Fergus. Automatic data augmentation for generalization in deep reinforcement learning. *arXiv preprint arXiv:2006.12862*, 2020.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pp. 355–366. Springer, 2011.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations*, 2016.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. 2020.

Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. *arXiv preprint arXiv:2006.08210*, 2020.

Xingyou Song, Yiding Jiang, Stephen Tu, Yilun Du, and Behnam Neyshabur. Observational overfitting in reinforcement learning. In *International Conference on Learning Representations*, 2019.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pp. 9870–9879. PMLR, 2021.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211, 1999.

Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. Poincare glove: Hyperbolic word embeddings. In *International Conference on Learning Representations*, 2018.

Saket Tiwari and M Prannoy. Hyperbolic embeddings for learning options in hierarchical reinforcement learning. *arXiv preprint arXiv:1812.01487*, 2018.

Abraham Albert Ungar. A gyrovector space approach to hyperbolic geometry. *Synthesis Lectures on Mathematics and Statistics*, 1(1):1–194, 2008.

Hado P Van Hasselt, Matteo Hessel, and John Aslanides. When to use parametric models in reinforcement learning? *Advances in Neural Information Processing Systems*, 32, 2019.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Kaixin Wang, Bingyi Kang, Jie Shao, and Jiashi Feng. Improving generalization in reinforcement learning with mixture regularization. *Advances in Neural Information Processing Systems*, 33: 7968–7978, 2020.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pp. 1995–2003. PMLR, 2016.

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*, 2018.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=GY6-6sTvGaf.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 10674–10681, 2021b.

Amy Zhang, Clare Lyle, Shagun Sodhani, Angelos Filos, Marta Kwiatkowska, Joelle Pineau, Yarin Gal, and Doina Precup. Invariant causal prediction for block mdps. In *International Conference on Machine Learning*, pp. 11214–11224. PMLR, 2020.

Hanping Zhang and Yuhong Guo. Generalization of reinforcement learning with policy-aware adversarial data augmentation. *arXiv preprint arXiv:2106.15587*, 2021.

Yuansheng Zhou, Brian H Smith, and Tatyana O Sharpee. Hyperbolic geometry of the olfactory space. *Science advances*, 4(8), 2018.