On the Benefit of Optimal Transport for Curriculum Reinforcement Learning

Pascal Klink, Carlo D'Eramo, Jan Peters, Joni Pajarinen

Abstract—Curriculum reinforcement learning (CRL) allows solving complex tasks by generating a tailored sequence of learning tasks, starting from easy ones and subsequently increasing their difficulty. Although the potential of curricula in RL has been clearly shown in various works, it is less clear how to generate them for a given learning environment, resulting in various methods aiming to automate this task. In this work, we focus on framing curricula as interpolations between task distributions, which has previously been shown to be a viable approach to CRL. Identifying key issues of existing methods, we frame the generation of a curriculum as a constrained optimal transport problem between task distributions. Benchmarks show that this way of curriculum generation can improve upon existing CRL methods, yielding high performance in various tasks with different characteristics.

Index Terms—Reinforcement Learning, Curriculum Learning, Optimal Transport

1 Introduction

Reinforcement learning (RL) [1] has celebrated great successes as a framework for the autonomous acquisition of desired behavior. With ever-increasing computational power, this framework and the algorithms developed under it have resulted in learning agents capable of solving non-trivial long-horizon planning [2, 3] and control tasks [4]. However, these successes have highlighted the need for certain forms of regularization, such as leagues in the context of board games [3], gradual diversification of simulated training environments for robotic manipulation [4] and -locomotion [5], or a tailored training pipeline in the context of humanoid control for soccer [6]. These regularizations can help overcome the shortcomings of modern RL agents, such as poor exploratory behavior – an active research topic [7, 8, 9].

One can view the regularizations mentioned above under the umbrella term of curriculum reinforcement learning [10], which aims to avoid the shortcomings of modern (deep) RL agents by learning on a tailored sequence of tasks. Such task sequences can materialize in various ways, and they are motivated by different perspectives in the literature, such as intrinsic motivation or regret minimization, to name some of them [11, 12, 13, 14, 15, 16].

A perspective of particular interest for this article is to interpret a curriculum as a sequence of task distributions that interpolate between an auxiliary task distribution – with the sole purpose of facilitating learning – and a distribution of target tasks [17]. We refer to these approaches as interpolation-based curricula. While algorithmic realizations of such curricula have been successfully evaluated in the literature [18, 19, 20], some evaluations indicated a relatively poor learning performance of these methods [21]. Furthermore, applications of interpolation-based curricula have been limited to scenarios with somewhat restricted distributions, such as Gaussian- or uniform ones. The observed performance gaps and lack of flexibility w.r.t. distribution parameterization call for a better understanding of these methods' inner workings to improve their performance and extend their applicability.

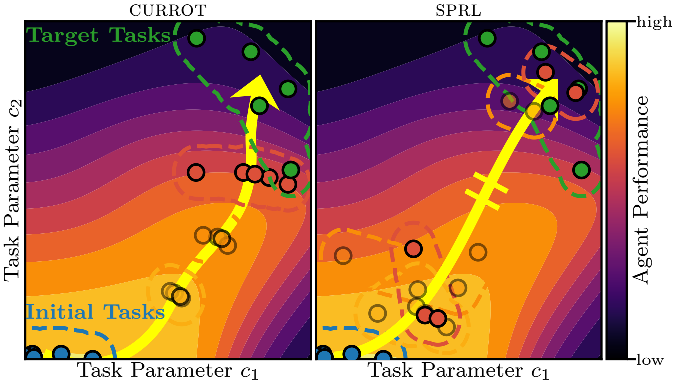


Fig. 1: Our approach (CURROT) addresses problems of existing curriculum RL methods, such as SPRL, which create curricula between a distribution of initial tasks (blue) and a distribution of target tasks (green). In this example, the curriculum can change the task via two parameters c_1 and c_2, leading to more or less challenging learning environments for an agent. Looking at the different stages of the curricula (colored points), we see that existing methods can lead to distributions that encode hard- and easy tasks, but ignore tasks of intermediate difficulty. Our method avoids such a splitting behavior, resulting in interpolations that gradually increase the task difficulty throughout the curriculum. Please see Sections 4 and 5 for a detailed description.

- *P. Klink and J. Peters are with the Technical University of Darmstadt, Germany, FG Intelligent Autonomous Systems. Correspondence to: pascal@robot-learning.de.*
- *J. Peters is also with the German Research Center for AI (DFKI), Research Department: Systems AI for Robot Learning, hessian.AI (Germany), and the Centre for Cognitive Science at Technical University of Darmstadt.*
- *C. D'Eramo is with the Center for Artificial Intelligence and Data Science at University of Würzburg (Germany), the Technical University of Darmstadt (Germany), and hessian.AI (Germany).*
- *J. Pajarinen is with the Department of Electrical Engineering and Automation, Aalto University, Finland. J. Pajarinen was supported by Academy of Finland (345521).*

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3390051

This article investigates the shortcomings of methods that realize curricula as a scheduled interpolation between task distributions based on the KL divergence and an expected performance constraint. We show how both these concepts can fail to produce meaningful curricula in simple examples. The demonstrated failure cases a) illustrate the importance of explicitly reasoning about the similarity of tasks when building a curriculum and b) show how parametric assumptions on the generated task distributions can masquerade failures of the underlying framework used to generate curricula. To resolve the observed issues, we explicitly specify the similarity of learning tasks via a distance function and use the framework of optimal transport to generate interpolating distributions that, independent of their parameterization, result in gradual task changes. Based on this explicit notion of task similarity, we propose our approach to curriculum RL (CURROT), which replaces the expected performance constraint with a more strict condition to obtain the behavior visualized in Figure 1. Furthermore, we contrast our approach with an alternative method, GRADIENT, recently proposed by Huang et al. [22]. We outline how both approaches use optimal transport to generate curricula but differ in their use of the agent performance to constrain the curriculum while avoiding the demonstrated pitfalls of expected performance constraints.

In experiments, we a) validate the correct behavior of both CURROT and GRADIENT free from approximations and parametric assumptions in a small discrete MDP and b) compare approximate implementations on a variety of tasks featuring discrete- and continuous task spaces, as well as Euclidean- and non-Euclidean measures of distance between learning tasks. In these experiments, both approaches show convincing performance with CURROT consistently matching and surpassing the performance of all other algorithms.

2 Related Work

This work generates training curricula for reinforcement learning (RL) agents. Unlike supervised learning, where there is an ongoing discussion about the mechanics and effects of curricula in different learning situations [23, 24], the mechanics seem to be more agreed upon in RL.

Curriculum Reinforcement Learning: In RL, curricula improve the learning performance of an agent by adapting the training environments to its proficiency. This adaptation of task complexity can reduce the sample complexity of RL, e.g., by bypassing poor exploratory behavior of non-proficient agents [25]. Using curricula can avoid the need for extensively engineered reward functions, which come with risks, such as failing to encode the intended behavior [26]. Applications of curricula to RL are widespread, and different terms have been established. Adaptive Domain Randomization [4] uses curricula to gradually diversify the training parameters of a simulator to facilitate sim-to-real transfer. Similarly, unsupervised environment discovery [16, 27, 28] aims to efficiently train an agent robust to variations in the environment and can be seen as a more general view of domain randomization. Automatic curriculum learning methods [12, 14, 17, 29, 30, 31, 32, 33] mainly focus on improving an agent's learning speed or performance on a set of desired tasks. Curricula are often generated as distributions that maximize a specific surrogate objective, such as learning progress [14, 34], intermediate task difficulty [30], regret [28], or disagreement between Q-functions [31]. Curriculum generation can also be interpreted as a two-player game [29]. The work by Jiang et al. [16] hints at a link between surrogate objectives and two-player games. Similar to the variety of objectives that the above algorithms optimize to build a curriculum, their implementations use drastically different approaches to approximate the training distribution for the agent, which is often defined over a continuous space of training tasks. For example, Florensa et al. [30] use a combination of GANs and a replay buffer to represent the task distribution. Portelas et al. [14] use a Gaussian mixture model to approximate the distribution of tasks that promise high learning progress. Jiang et al. [16] use a fixed-size replay buffer to realize an approximate distribution of high-regret tasks, simultaneously encouraging frequent replay of buffered tasks to keep a more accurate estimate of regret.

Interpolation-based curriculum RL algorithms formulate the generation of a curriculum as an explicit interpolation between an auxiliary task distribution and a distribution of target tasks [17, 18, 20]. This interpolation is subject to a constraint on the expected agent performance that paces its progress toward the target tasks. As highlighted by Klink et al. [17], such interpolations can be formally linked to successful curricula in supervised learning [35], the concept of annealing in statistics [36], and homotopic continuation methods in optimization [37]. As for the algorithms based on surrogate objectives, realizations of these interpolation-based curricula inevitably need to rely on approximations such as the restriction to Gaussian distributions in [17, 18, 19] or approximate update rules enabled by uniform target task distributions [20].

This article reveals shortcomings of the aforementioned interpolation-based curriculum RL methods, highlighting how approximations can masquerade issues in the conceptual algorithm formulations. One ingredient to overcome these shortcomings is an explicit notion of task similarity that we formulate as a distance function between tasks. We can then lift this distance function into the space of probability measures using *optimal transport*.

Optimal Transport: Dating back to work by Monge in the 18th century, *optimal transport* has been understood as a fundamental concept touching upon many fields in both theory and application [38, 39]. In probability theory, optimal transport translates to the so-called Wasserstein metric [40] that compares two distributions under a given metric, allowing, e.g., for the analysis of probabilistic inference algorithms as approximate gradient flows [41] and providing well-defined ways of comparing feature distributions or even graphs in computer vision and machine learning [42, 43, 44]. Gromov-Wasserstein distances [45, 46] even allow comparing distributions across metric spaces, which has been of use, e.g., in computational biology [47] or imitation learning [48]. In Reinforcement learning, optimal transport has not found widespread application, albeit some interesting works exist. Zhang et al. [49] provide a natural extension of the work by Liu et al. [41] and interpret policy optimization as Wasserstein gradient flows. Metelli et al.

[50] use Wasserstein barycenters to propagate uncertainty about value function estimates in a Q-learning approach. In more applied scenarios, optimal transport has been used to regularize RL in sequence generation- [51] or combinatorial optimization problems [52]. In goal-conditioned RL, Wasserstein distances have been previously applied to improve goal generation in the hindsight experience replay framework [53] and to realize well-performing data-driven reward functions by combining them with so-called time-step metrics [54]. Recently, Cho et al. [55] combined the data-driven reward function proposed by Durugkar et al. [54] with a curriculum that, similarly to the work by [53], improves the selection of training goals from a buffer of achieved ones. When it comes to building RL curricula over arbitrary MDPs using Optimal Transport, we are only aware of our work [56] at ICML 2022 and the work by [22] at NeurIPS 2022, which we present from a unified perspective and compare in this journal article. In addition to the aforementioned methods in goal-conditioned RL, this article emphasizes curriculum reinforcement learning as another promising application domain for optimal transport. An important issue of applied optimal transport is its computational complexity. In Appendix A, we discuss the computational aspects of optimal transport in more detail.

3 Preliminaries

This section introduces the necessary background on (contextual) RL, curriculum RL, and optimal transport.

3.1 Contextual Reinforcement Learning

Contextual reinforcement learning [57] can be seen as a conceptual extension to the (single task) reinforcement learning (RL) problem

$$\begin{aligned}\max_{\pi} J(\pi) &= \max_{\pi} \mathbb{E}_{p(\boldsymbol{\tau}|\pi)}\left[\sum_{t=0}^{\infty}\gamma^{t} r(\mathbf{s}_t,\mathbf{a}_t)\right] \qquad (1)\\ \boldsymbol{\tau} &= \{(\mathbf{s}_t,\mathbf{a}_t)|t=1,\dots\}\\ p(\boldsymbol{\tau}|\pi) &= p_0(\mathbf{s}_0)\prod_{t=1}^{\infty} p(\mathbf{s}_t|\mathbf{s}_{t-1},\mathbf{a}_{t-1})\pi(\mathbf{a}_{t-1}|\mathbf{s}_{t-1}),\end{aligned}$$

which aims to maximize the above expected discounted reward objective by finding an optimal policy $\pi{:}\mathcal{S}{\times}\mathcal{A} \mapsto \mathbb{R}_{\geq 0}$ for a given MDP $\mathcal{M}{=}\langle \mathcal{S},\mathcal{A},p,r,p_0\rangle$ with initial state distribution p_0 and transition dynamics p. Contextual RL extends this objective to a space of MDPs $\mathcal{M}(\mathbf{c}){=}\langle \mathcal{S},\mathcal{A},p_{\mathbf{c}},r_{\mathbf{c}},p_{0,\mathbf{c}}\rangle$ equipped with a distribution $\mu{:}\mathcal{C}{\mapsto}\mathbb{R}$ over contextual variables $\mathbf{c}\in\mathcal{C}$

$$\max_{\pi} J(\pi,\mu) = \max_{\pi} \mathbb{E}_{\mu(\mathbf{c})}\left[J(\pi,\mathbf{c})\right]. \tag{2}$$

The policy $\pi : \mathcal{S}{\times}\mathcal{C}{\times}\mathcal{A} \mapsto \mathbb{R}$ is conditioned on the contextual parameter $\mathbf{c}$. The distribution $\mu(\mathbf{c})$ encodes the tasks $\mathcal{M}(\mathbf{c})$ to be solved by the agent. Objective $J(\pi,\mathbf{c})$ in Eq. (2) corresponds to objective $J(\pi)$ in Eq. (1) with the initial state distribution p_0, the transition dynamics p as well as the reward function r of $\mathcal{M}$ replaced by their counterparts in $\mathcal{M}(\mathbf{c})$. This contextual model of optimal decision-making is well-suited for learning in multiple related tasks, as is the case in multi-task- [58], goal-conditioned- [59], or curriculum RL [10]. At this point, we want to emphasize that the context $\mathbf{c}$ could be readily embedded in the state space $\mathcal{S}$, resulting in a regular MDP in which the context – as part of the state – remains constant throughout an episode. The context distribution $\mu(\mathbf{c})$ would then be subsumed into the initial state distribution without losing expressiveness. We nonetheless prefer the contextual RL framework, coined by Hallak et al. [57], as it emphasizes the distribution $\mu(\mathbf{c})$, which is at the heart of curriculum RL methods, as we will see now.

3.2 Curriculum Reinforcement Learning

On an abstract level, curriculum RL methods can be understood as generating a sequence of task distributions $(p_i{:}\mathcal{C}{\mapsto}\mathbb{R})_i$ under which to train an RL agent by maximizing $J(\pi,p_i)$ w.r.t. π. When chosen appropriately, solving this sequence of optimization problems can yield a policy that performs better on the target distribution $\mu(\mathbf{c})$ than a policy found by maximizing $J(\pi,\mu)$ directly. The benefit of such mediating distributions is particularly obvious in settings where initially random agent behavior is unlikely to observe any meaningful learning signals, such as in sparse-reward learning tasks.

CRL methods differ in the specification of p_i. Often, the distribution is defined to prioritize tasks that maximize certain surrogate quantities, such as absolute learning progress [14], regret [28], or tasks of intermediate success probability [30]. This article focuses on CRL methods that model p_i as the solution to an optimization problem that aims to minimize a distance or divergence between p_i and μ. One of these approaches [17, 18, 19] defines p_i as the distribution with minimum KL divergence to μ that fulfills a constraint on the expected agent performance

$$\begin{aligned}&\min_{p} D_{\text{KL}}\left(p(\mathbf{c})\|\mu(\mathbf{c})\right) \qquad (3)\\ &\text{s.t.}\quad J(\pi,p)\geq\delta \qquad D_{\text{KL}}\left(p(\mathbf{c})\|q(\mathbf{c})\right)\leq\epsilon,\end{aligned}$$

where δ is the desired level of performance to be achieved by the agent π under $p(\mathbf{c})$ and ϵ limits the maximum KL divergence to the previous context distribution $q(\mathbf{c}){=}p_{i-1}(\mathbf{c})$. The optimizer of (3) balances between tasks likely under the (target) distribution $\mu(\mathbf{c})$ and tasks in which the agent currently obtains large rewards. The KL divergence constraint w.r.t. the previous context distribution $q(\mathbf{c})$ prevents large changes in $p(\mathbf{c})$ during subsequent iterations, avoiding the exploitation of faulty estimates of the agent performance $J(\pi,p)$ from a limited amount of samples. Objective (3) performs an interpolation between the distributions $p_{\eta}(\mathbf{c}){\propto}\mu(\mathbf{c})\exp(\eta J(\pi,\mathbf{c}))$ and $q(\mathbf{c})$, given by

$$p_{\alpha,\eta}(\mathbf{c}) \propto \left(\mu(\mathbf{c})\exp(J(\pi,\mathbf{c}))^{\eta}\right)^{\alpha} q(\mathbf{c})^{1-\alpha}. \tag{4}$$

The two parameters α and η that control the interpolation are the Lagrangian multipliers of the two constraints in objective (3). We will later investigate the behavior of this interpolating distribution.

3.3 Optimal Transport

The problem of optimally transporting density between two distributions has been initially investigated by Monge [60]. As of today, generalizations established by Kantorovich [40] have led to so-called **Wasserstein distances** as metrics between probability distributions defined on a metric space $M{=}(d,\mathcal{C})$ with metric $d:\mathcal{C}\times\mathcal{C}\mapsto\mathbb{R}_{\geq 0}$

$$\mathcal{W}_p(p_1,p_2)=\left(\inf_{\phi\in\Phi(p_1,p_2)}\mathbb{E}_\phi\left[d(\mathbf{c}_1,\mathbf{c}_2)^p\right]\right)^{1/p}, \quad p\geq 1$$

$$\Phi(p_1,p_2)=\{\phi:\mathcal{C}\times\mathcal{C}\mapsto\mathbb{R}_{\geq 0}|p_i=P_{i\#}\phi,\ i\in\{1,2\}\},$$

where $P_{i\#}$ are the push-forwards of the maps $P_1(\mathbf{c}_1,\mathbf{c}_2){=}\mathbf{c}_1$ and $P_2(\mathbf{c}_1,\mathbf{c}_2){=}\mathbf{c}_2$. We refer to [38, Chapter 2] for an excellent and intuitive introduction to these concepts. The distance between p_1 and p_2 is obtained via the solution to an optimization problem that finds a so-called plan, or coupling, ϕ. This coupling encodes how to equalize p_1 and p_2, considering the cost of moving density between parts of the space $\mathcal{C}$. The metric d encodes this cost. In the following, we will always assume to work with 2-Wasserstein distances, i.e., $p{=}2$, due to their suitedness for interpolating measures [see 38, Chapter 6 and Remark 2.24].

Similar to how (weighted) means can be defined as solutions to optimization problems on a metric space $M{=}(d,\mathcal{C})$, Wasserstein distances allow us to define what is referred to as Wasserstein barycenters [61]

$$\mathcal{B}_2(W,P)=\arg\min_p\sum_{k=1}^{K}w_k\mathcal{W}_2(p,p_k), \tag{5}$$

which represent the (weighted) mean of the distributions $P{=}\{p_k|k{\in}[1,K]\}$ with weights $W{=}\{w_k|k{\in}[1,K]\}$.

4 Curriculum Reinforcement Learning as Constrained Optimal Transport

At this point, we can motivate our approach to curriculum RL by looking at the limitations of Objective 3 caused by a) measuring similarity between context distributions via the KL divergence and b) the expected performance constraint used to control the progression towards $\mu(\mathbf{c})$.

4.1 Limitations of the KL Divergence

Given the complexity of computing $D_{\text{KL}}\left(p(\mathbf{c})\|\mu(\mathbf{c})\right)$ for arbitrary distributions, previous work restricts $\mu(\mathbf{c})$ either to a Gaussian distribution [17, 18, 19] or to be uniform over $\mathcal{C}$ to ease computation and optimization of a weighted KL divergence objective [20]. While empirically successful, these design choices masquerade the pitfalls of the KL divergence to measure distributional similarity in a CRL setting, particularly when dealing with a target distribution that does not assign uniform density over all of $\mathcal{C}$.

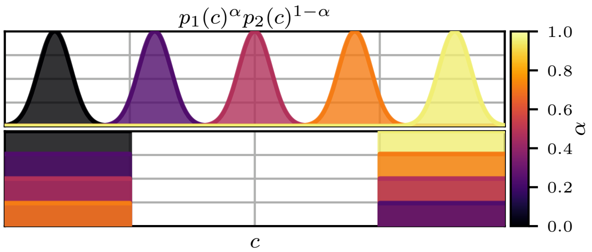


Fig. 2: Interpolations generated by optimizing Objective (6) for different values of ϵ (and with that α). In the top row, $p_1(c)$ and $p_2(c)$ are Gaussian, while in the bottom row, they assign uniform density over different parts of $\mathcal{C}$.

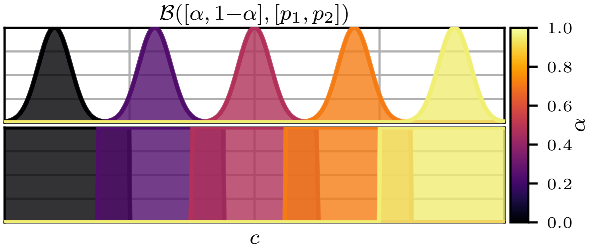


Fig. 3: Wasserstein barycenters $\mathcal{B}([\alpha,1-\alpha],[p_1,p_2])$ between the distributions shown in Figure 2. In the top row, $p_1(c)$ and $p_2(c)$ are Gaussian while in the bottom row, they assign uniform density over different parts of $\mathcal{C}$.

To demonstrate this issue, we focus on an interpolation task between two distributions

$$p_1(\mathbf{c})^{\alpha(\epsilon)}p_2(\mathbf{c})^{1-\alpha(\epsilon)}=\arg\min_{p\in\{q|D_{\text{KL}}(q\|p_2)\leq\epsilon\}}D_{\text{KL}}\left(p\|p_1\right), \tag{6}$$

corresponding to a version of Objective (3) with no constraint on the expected agent performance. Figure 2 demonstrates the sensibility of this interpolation to the parametric representation of the distributions $\mu(\mathbf{c})$ and $q(\mathbf{c})$. While for Gaussian distributions, interpolations of the form $p_1(\mathbf{c})^\alpha p_2(\mathbf{c})^{1-\alpha}$ gradually shift density in a metric sense, this behavior is all but guaranteed for non-Gaussian distributions. The interpolation between two uniform distributions with quasi-limited support [1] in the bottom row of Figure 2 displaces density from contexts $\mathbf{c}$ to contexts $\mathbf{c}'$ with large Euclidean distance $\|\mathbf{c}-\mathbf{c}'\|_2$. In settings in which the Euclidean distance between contexts $\mathbf{c}_1$ and $\mathbf{c}_2$ is a good indicator for the similarity between $\mathcal{M}(\mathbf{c}_1)$ and $\mathcal{M}(\mathbf{c}_2)$, the observed ignorance of the KL divergence w.r.t. the underlying geometry of the context space leads to curricula with "jumps" in task similarity. We can easily convince ourselves that such jumps are not a hypothetical problem by recalling that neural network-based policies $\pi(\mathbf{a}|\mathbf{s},\mathbf{c}){=}\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s},\mathbf{c})$ tend to gradually change their behavior with increasing Euclidean distance to $\mathbf{c}$.

At this point, we can leverage the notion of optimal transport to explicitly encode the similarity of two tasks, $\mathcal{M}(\mathbf{c})$ and $\mathcal{M}(\mathbf{c}')$, via a metric $d(\mathbf{c},\mathbf{c}')$ and realize the interpolation between distributions on the resulting metric space as Wasserstein barycenters (Eq. 5). As we see in Figure 3, this explicit notion of task similarity allows to generate interpolations that are stable across changes in the parameterization of context distributions and interpolate between arbitrary distributions that are not absolutely continuous w.r.t. each other. Consequently, the optimization problem

$$\min_p\mathcal{W}_2(p,\mu)\ \text{ s.t. }J(\pi,p)\geq\delta \tag{7}$$

is a promising approach to leverage optimal transport in curriculum RL. We iterate on this candidate in the next section by investigating the role of the expected performance constraint when generating curricula for reinforcement learning agents.

[1] We ensure a negligible positive probability density across all of $\mathcal{C}$ to allow for the computation of KL divergences.

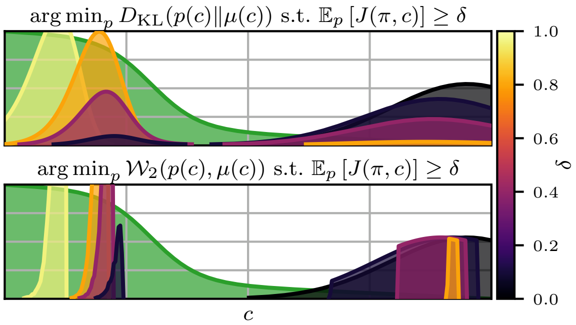


Fig. 4: Interpolations using KL divergence (top) and Wasserstein distance (bottom) subject to an expected performance constraint with different threshold values δ. The performance $J(\pi, c)$ is visualized in green.

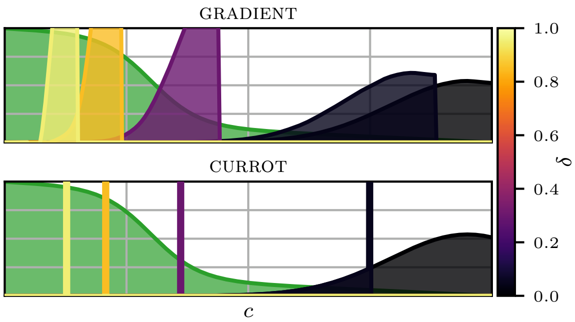


Fig. 5: Interpolations generated by GRADIENT (Eq. 9, top) and CURROT (Eq. 8, bottom) for different threshold values δ. The performance $J(\pi, c)$ is visualized in green.

4.2 Challenges of Expected Performance Constraints

The SPRL objective (3) controls the interpolation speed between the initial- and target task distribution by the expected performance of the current agent under the chosen context distribution $J(\pi, p)$. As detailed in [17], this expected performance constraint allows for establishing a connection to self-paced learning for supervised learning tasks [35, 62]. While this formal connection is interesting in its own right, we show in Figure 4 that the expected performance constraint in SPRL can lead to encoding both too simple and too complex tasks, given the current agent capabilities. Furthermore, using Wasserstein distances in Objective (7) does not resolve this issue. In Figure 4, both methods encode tasks with very high and very low agent returns to fulfill the expected performance constraint, sidestepping the goal of encoding tasks of intermediate difficulty. At this point, we can propose our algorithm CURROT and introduce a recent algorithm proposed by Huang et al. [22] – called GRADIENT– as two ways of resolving the observed interpolation issue:

1) **CURROT** restricts the support of $p(\mathbf{c})$ to those contexts $\mathbf{c} \in \mathcal{C}$ that fulfill the performance constraint $J(\pi, \mathbf{c}) \geq \delta$. We refer to this set as $\mathcal{V}(\pi, \delta) = \{\mathbf{c} | \mathbf{c} \in \mathcal{C}, J(\pi, \mathbf{c}) \geq \delta\}$. With this notation in place, we frame the restricted optimization as

$$\min_p \mathcal{W}_2(p, \mu) \quad \text{s.t.} \quad p(\mathcal{V}(\pi, \delta)) = 1. \tag{8}$$

Putting the constraint in words, we require that the curriculum assigns all probability density of p to contexts that satisfy the performance constraint.

2) **GRADIENT** restricts the interpolation to the barycentric interpolation (5) between the initial- and target context distribution, i.e. $p_\alpha(\mathbf{c}) = \mathcal{B}_2([1-\alpha, \alpha], [p_0(\mathbf{c}), \mu(\mathbf{c})])$. This restriction prevents the problematic behavior shown in Figure 4 while still allowing to adjust α using an expected performance constraint

$$\max_{\alpha \in [0,1]} \alpha \quad \text{s.t.} \quad J(\pi, p_\alpha) \geq \delta. \tag{9}$$

As shown in Figure 5, both of these methods avoid the behavior generated by Objective (7), resulting in an interpolation that gradually deforms the distribution in a metric sense with changing agent competence. In the remainder of this article, we will investigate exact and approximate versions of these algorithms to understand their behavior better. The first observation in this regard is that the curriculum of GRADIENT is entirely predetermined by the given metric $d(\mathbf{c}_1, \mathbf{c}_2)$ as well as the target- and initial distribution $\mu(\mathbf{c})$ and $p_0(\mathbf{c})$. The agent performance only influences how fast the curriculum proceeds towards $\mu(\mathbf{c})$. On the other hand, CURROT reshapes the curriculum based on the current agent performance to avoid sampling contexts with a performance lower than the threshold δ. Figure 5 shows that this reshaping results in a tendency of CURROT to place all probability density on the border of the desired agent performance δ until reaching regions of non-zero probability density under $\mu(\mathbf{c})$. At this point, the curriculum matches the target density in those parts of $\mathcal{C}$, in which the performance constraint is fulfilled, and continues to concentrate all remaining density on the boundaries of agent capability. This behavior is similar to those CRL methods that combine task-prioritization with a replay buffer of, e.g., previously solved tasks to prevent catastrophic forgetting, such as GOALGAN or PLR [28, 30]. To the best of our knowledge, such behavior has not yet been motivated by a first-principle optimization objective in the context of curriculum RL.

5 Approximate Algorithms for Discrete- and Continuous Context Spaces

Objectives (8) and (9) face challenges in more realistic application scenarios with either large discrete- or continuous context spaces due to two reasons:

1) We do not have access to the expected performance $J(\pi, \mathbf{c})$ of an agent π in context $\mathbf{c}$ but can only estimate it from observed training episodes.
2) Computing Wasserstein barycenters for arbitrary continuous- or discrete distributions in non-Euclidean spaces can quickly become intractably expensive.

The following sections address the above problems to benchmark CURROT and GRADIENT in non-trivial experimental settings.

5.1 Approximate Wasserstein Barycenters

Before branching into the description of the two algorithms, we first describe a particle-based approximation to the computation of Wasserstein Barycenters, which allows us to cheaply approximate Barycenters for the GRADIENT algorithm in large discrete state-spaces and is essential for the

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3390051

approximate implementation of the CURROT algorithm.

For approximating a Barycenter $p_\alpha{=}\mathcal{B}([1{-}\alpha,\alpha],[p_0,\mu])$, we first sample a set of N particles from $\mu(\mathbf{c})$ and $p_0(\mathbf{c})$ to form the empirical distributions

$$\hat{\mu}(\mathbf{c}) = \frac{1}{N}\sum_{n=1}^{N}\delta_{\mathbf{c}_{\mu,n}}(\mathbf{c}), \quad \mathbf{c}_{\mu,n}\sim\mu(\mathbf{c}) \tag{10}$$

$$\hat{p}_0(\mathbf{c}) = \frac{1}{N}\sum_{n=1}^{N}\delta_{\mathbf{c}_{p_0,n}}(\mathbf{c}), \quad \mathbf{c}_{p_0,n}\sim p_0(\mathbf{c}),$$

where $\delta_{\mathbf{c}_{\text{ref}}}(\mathbf{c})$ represents a Dirac distribution centered at $\mathbf{c}_{\text{ref}}$. Due to the discrete nature of $\hat{\mu}(\mathbf{c})$ and $\hat{p}_0(\mathbf{c})$, the coupling $\phi(\mathbf{c}_1,\mathbf{c}_2)$ reduces to a permutation $\phi{\in}\text{Perm}(N)$, which assigns the particles between $\hat{p}_0$ and $\hat{\mu}$ [38, Section 2.3]. With that, the computation of $\mathcal{W}_2(\hat{p}_0,\hat{\mu})$ reduces to

$$\min_{\phi\in\text{Perm}(N)}\left(\frac{1}{N}\sum_{n=1}^{N}d(\mathbf{c}_{p_0,n},\mathbf{c}_{\mu,\phi(n)})^2\right)^{\frac{1}{2}}. \tag{11}$$

Since a permutation is a particular case of a coupling [38, Section 2.3], we overload the meaning of ϕ to be either a permutation or coupling, depending on the number of arguments. With today's computing hardware, assignment problems like (11) can be solved on a single CPU core in less than a second for N in the hundreds, which is typically enough to represent the context distributions [2]. Given this optimal assignment, we then compute the Fréchet mean for each particle pair

$$\mathbf{c}_{\alpha,n}{=}\arg\min_{\mathbf{c}\in\mathcal{C}}(1{-}\alpha)d(\mathbf{c},\mathbf{c}_{p_0,n})^2+\alpha d(\mathbf{c},\mathbf{c}_{\mu,\phi(n)})^2 \tag{12}$$

to form the barycenter $\hat{p}_\alpha(\mathbf{c}){=}\frac{1}{N}\sum_{n=1}^{N}\delta_{\mathbf{c}_{\alpha,n}}(\mathbf{c})$. While certainly less efficient than specialized routines for Barycenter computations in Euclidean Spaces, such as e.g., the GeomLoss library [63], the presented approach is useful when dealing with large discrete spaces. In this case, faithful Barycenter computations must work with the full distance matrix. Assuming a discrete context space of size S and neglecting the cost of computing the optimal assignment, the approximate barycenter computation requires $O(N^2{+}2NS)$ evaluations of the distance function. Hence for $S\gg N$, even computing the $\frac{S(S+1)}{2}$ entries of the entire distance matrix required for a single step in the Sinkhorn algorithm becomes more expensive than the presented approximate method. Additionally, reducing the Barycenter computation to an optimization problem over individual particles easily allows to incorporate additional constraints that are required by the CURROT optimization objective (8).

5.2 Approximate GRADIENT

Huang et al. [22] propose to compute barycenters between $p_0(\mathbf{c})$ and $\mu(\mathbf{c})$ for discrete steps of size ϵ. Starting from $\alpha{=}0$, the agent trains for M episodes on tasks sampled from the current distribution. If the average episodic return $\frac{1}{M}\sum_{m=1}^{M}R_m$ is greater or equal to δ, α is increased by ϵ and the distribution is set to be the Wasserstein barycenter for the updated value of α.

This step-wise increase of α avoids the explicit optimization over α and, with that, the need to estimate the performance of the current policy π for a given context $\mathbf{c}$. Having laid out a way of computing approximate Barycenters in the previous section, we can summarize our implementation of GRADIENT in Algorithm 1.

Algorithm 1 Approximate GRADIENT

```
Input: Initial context dist. $p_0(\mathbf{c})$, target context dist. $\mu(\mathbf{c})$, metric $d(\mathbf{c}_1,\mathbf{c}_2)$, performance bound $\delta$, step size $\epsilon$
Initialize: $\alpha = 0$
while True do
    Compute $\hat{p}_\alpha(\mathbf{c}) = \frac{1}{N}\sum_{n=1}^{N}\delta_{\mathbf{c}_{\alpha,n}}(\mathbf{c})$ (Eq. (11) and (12))
    Agent Improvement:
    Sample contexts $\mathbf{c}_m \sim \hat{p}_\alpha(\mathbf{c}),\ m\in[1,M]$
    Train policy $\pi$ under $\mathbf{c}_m$ and observe episodic rewards $R_m = \sum_{t=0}^{\infty}\gamma^t r_{\mathbf{c}_m}(\mathbf{s}_t,\mathbf{a}_t),\ m\in[1,M]$
    Context Distribution Update:
    if $\frac{1}{M}\sum_{m=1}^{M}R_m{\geq}\delta$ then
        Advance interpolation $\alpha = \min(\alpha+\epsilon,1)$
    end if
end while
```

5.3 Approximate CURROT

As for the GRADIENT algorithm, we make use of an empirical distribution $\hat{p}(\mathbf{c})$ to represent the context distribution $p(\mathbf{c})$ (see Eq. 10). Unlike for GRADIENT, there is no possibility to side-step the estimation of $J(\pi,\mathbf{c})$ for CURROT, and any estimator of $J(\pi,\mathbf{c})$ will inevitably make mistakes. The mistakes will be particularly big for contexts $\mathbf{c}$ with a considerable distance to those sampled under the current training distribution $p(\mathbf{c})$. To avoid exploiting such erroneous performance predictions, we introduce a trust region constraint similar to the seminal SPRL objective (3) into CURROT

$$\begin{aligned}&\min_p \mathcal{W}_2(p,\mu) &(13)\\ &\text{s.t.}\ \ p(\mathcal{V}(\pi,\delta)){=}1 \qquad \mathcal{W}_2(p,q)\leq\epsilon,\end{aligned}$$

which limits the Wasserstein distance between the current- and next context distribution $q(\mathbf{c})$ and $p(\mathbf{c})$. Please note that we overload the meaning of the symbol ϵ with step size for GRADIENT and the trust region for CURROT, as both concepts limit the change in sampling distribution between updates. We realize the performance estimator using Nadaraya-Watson kernel regression [64, 65] with a squared exponential kernel

$$\hat{J}(\pi,\mathbf{c}){=}\frac{\sum_{l=1}^{L}K_h(\mathbf{c},\mathbf{c}_l)R_l}{\sum_{l=1}^{L}K_h(\mathbf{c},\mathbf{c}_l)},\ K_h(\mathbf{c},\mathbf{c}_l){=}\exp\left(-\frac{d(\mathbf{c},\mathbf{c}_l)^2}{2h^2}\right).$$

This estimator does not rely on gradient-based updates and requires no architectural choices except for the lengthscale h, consequently not complicating the application of the overall algorithm. We postpone the discussion of this lengthscale parameter h until after we have discussed the approximate optimization of Objective (13) and first focus on the choice of dataset $\mathcal{D}{=}\{(\mathbf{c}_l,R_l)|l\in[1,L]\}$ used to build the kernel regressor.

We create the dataset from two buffers, $\mathcal{D}_+$ and $\mathcal{D}_-$, of size N. We update the buffers with the results

[2] In our experiments, we use less than a thousand particles in all experiments

Algorithm 2 Approximate CURROT

```
Input: Initial context dist. $p_0(\mathbf{c})$, target context dist. $\mu(\mathbf{c})$, metric $d(\mathbf{c}_1,\mathbf{c}_2)$, performance bound $\delta$, distance bound $\epsilon$
Initialize: $\hat{p}(\mathbf{c}) = \frac{1}{N}\sum_{n=1}^{N}\delta_{\mathbf{c}_{p_0,n}}(\mathbf{c}),\ \mathbf{c}_{p_0,n}\sim p_0(\mathbf{c})$
while True do
    Agent Improvement:
    Sample contexts $\mathbf{c}_m \sim \hat{p}(\mathbf{c}),\ m\in[1,M]$
    Train policy $\pi$ under $\mathbf{c}_m$ and observe episodic rewards $R_m=\sum_{t=0}^{\infty}\gamma^t r_{\mathbf{c}_m}(\mathbf{s}_t,\mathbf{a}_t),\ m\in[1,M]$
    Context Distribution Update:
    Update buffers $\mathcal{D}_+$ and $\mathcal{D}_-$ with $\{(\mathbf{c}_m,R_m)|m\in[1,M]\}$

    Estimate $\hat{J}(\pi,\mathbf{c})\approx J(\pi,\mathbf{c})$ from $\mathcal{D}_+$ and $\mathcal{D}_-$
    Update $\hat{p}(\mathbf{c})$ via Eq. (14) and $\hat{J}(\pi,\mathbf{c})$, $\hat{p}(\mathbf{c})$, $\hat{\mu}(\mathbf{c})$
end while
```

of policy rollouts $(\mathbf{c},R_\mathbf{c})$ during agent training, where $R_\mathbf{c}=\sum_{t=0}^{\infty}\gamma^t r_\mathbf{c}(\mathbf{s}_t,\mathbf{a}_t)$. While $\mathcal{D}_-$ is simply a circular buffer that keeps the most recent N rollouts with $R_\mathbf{c}$ below the performance threshold δ, $\mathcal{D}_+$ contains contexts $\mathbf{c}$ for which $R_\mathbf{c}\geq\delta$. However, $\mathcal{D}_+$ is updated differently if full. Once full, we interpret the samples in $\mathcal{D}_+$ as an empirical distribution $\hat{p}_+(\mathbf{c})$ and select rollouts from the union of $\mathcal{D}_+$ and the set of new rollouts above the performance threshold δ to minimize $\mathcal{W}_2(\hat{p}_+,\hat{\mu})$. This optimal selection can be computed with a generalized version of the optimal assignment problem (11), where $\hat{p}_+$ is represented by N_+ particles and $\hat{\mu}$ is represented by N particles with $N_+\geq N$. The generalized problem then produces a selection of N particles to represent $\hat{p}_+$, which minimizes the resulting distance $\mathcal{W}(\hat{p}_+,\hat{\mu})$. We can hence interpret $\hat{p}_+(\mathbf{c})$ as a conservative solution to the CURROT objective (8). The solution is conservative since the particles are obtained from past iterations and may exceed the performance threshold δ by some margin, hence not targeting the exact border of the performance threshold. To more precisely target this border of agent competence, we proceed as follows: First, we solve an assignment problem between $\hat{p}(\mathbf{c})$ and $\hat{p}_+(\mathbf{c})$ to obtain pairs $(\mathbf{c}_{p,n},\mathbf{c}_{p_+,\phi(n)})$. We then reset $\mathbf{c}_{p,n}{=}\mathbf{c}_{p_+,\phi(n)}$ for those contexts $\mathbf{c}_{p,n}$ with $\hat{J}(\pi,\mathbf{c}_{p,n}){<}\delta$. Next, we again sample an empirical target distribution $\hat{\mu}(\mathbf{c})$ and solve an assignment problem between the updated empirical distribution $\hat{p}(\mathbf{c})$ and $\hat{\mu}(\mathbf{c})$ to obtain context pairs $(\mathbf{c}_{p,n},\mathbf{c}_{\mu,\phi(n)})$. We then solve an optimization problem for each pair to obtain the particles for the new empirical context distribution

$$\begin{aligned}&\operatorname*{arg\,min}_{\mathbf{c}\in\mathcal{C}}\ d(\mathbf{c},\mathbf{c}_{\mu,\phi(n)})\\&\quad\text{s.t. } \hat{J}(\pi,\mathbf{c})\geq\delta \qquad d(\mathbf{c},\mathbf{c}_{p,n})\leq\epsilon.\end{aligned}\tag{14}$$

Note that the restriction $d(\mathbf{c},\mathbf{c}_{p,n}){\leq}\epsilon$ ensures that $\mathcal{W}_2(\hat{p},\hat{q}){\leq}\epsilon$, while de-coupling the optimization for the individual particles. We use a simple approximate optimization scheme that samples a set of candidate contexts around $\mathbf{c}_{p,n}$ and selects the candidate that minimizes the distance to $\mathbf{c}_{\mu,\phi(n)}$ while fulfilling the performance constraint. In the continuous Euclidean settings, we uniformly sample candidates in the half ball of contexts that make an angle of less than 90 degrees with the descent direction $\mathbf{c}_{p,n}-\mathbf{c}_{\mu,\phi(n)}$. In discrete context spaces, we evaluate all contexts in the trust region. If even after resetting $\mathbf{c}_{p,n}{=}\mathbf{c}_{p_+,\phi(n)}$, no candidate satisfies the performance threshold, and hence Objective (14) is infeasible, we set $\mathbf{c}_{p,n}$ to the candidate with maximum performance in the ϵ-ball.

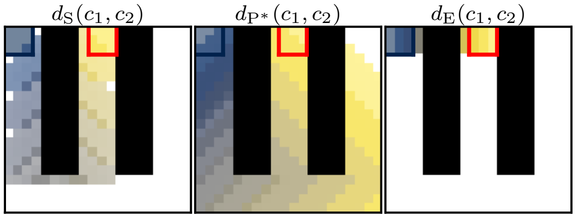


Fig. 6: E-Maze environment and visualizations of barycenters between initial- and target task distribution for the shortest-path distance d_S, performance pseudo-distance d_{P^*} and Euclidean distance d_E. Brighter colors correspond to distributions generated at later stages of the interpolation. The states covered by initial- and target task distributions are highlighted by the blue and red lines.

Having defined Objective (14), we can discuss the length-scale parameter h of the Nadaraya-Watson estimator. Given that the purpose of the estimator is to capture the trend in the ϵ-ball around a particle $\mathbf{c}_{p,n}$, we simply set the length-scale to 0.3ϵ. This choice ensures that the two-times standard deviation interval of the squared-exponential kernel K_h centered on $\mathbf{c}_{p,n}$ covers the trust region.

Like for GRADIENT, we train on $p_0(\mathbf{c})$ until reaching an average performance of at least δ, at which point we update the distribution according to Algorithm 2.

6 Experiments

To demonstrate the behavior of the introduced algorithms CURROT and GRADIENT, we benchmark the algorithms in different environments that feature discrete- and continuous context spaces with Euclidean- and non-Euclidean distance metrics. We furthermore evaluate both the exact approaches as well as their approximate implementations. To highlight the benefits of the proposed approach over currently popular CRL methods, we compare against a range of baselines. More precisely, we evaluate ALP-GMM [14], GOALGAN [30], PLR [28], VDS [31] and ACL [66] in addition to a random curriculum and training directly on $\mu(\mathbf{c})$ (referred to as Default). Details of the experiments, such as hyperparameters and employed RL algorithms, can be found in Appendix C. The code for running the experiments will be made publicly available upon acceptance.

6.1 E-Maze Environment

To investigate CURROT and GRADIENT without relying on approximations and highlight the effect of the chosen distance metric, we start the experiments with the environment shown in Figure 6. In this sparse-reward environment that is represented by a 20 × 20 grid, an agent is tasked to reach a goal position by moving around an elongated wall (black tiles in Figure 6). The curricula for this task control the goal position to be reached via the context c. We investigate three different distance functions of $\mathcal{C}$ in this environment:

- A Euclidean distance $d_{\mathrm{E}}(c_1, c_2){=}\|\mathbf{r}(c_1) - \mathbf{r}(c_2)\|_2$ based on representations $\mathbf{r}(c) \in \mathbb{R}^3$ of the discrete contexts which encode the two-dimensional goal position as well as the height (walls have a height of 200 and regular tiles a height of zero).
- A shortest-path distance $d_{\mathrm{S}}(c_1, c_2)$ computed using the Dijkstra algorithm. The search graph for the Dijkstra algorithm is built by connecting neighboring contexts using the previously defined Euclidean distance.
- A pseudo-metric investigated by Huang et al. [22] that is based on the optimal policy's absolute difference in expected return $d_{\mathrm{P}^*}(c_1, c_2){=}|J_{\pi^*}(c_1) - J_{\pi^*}(c_2)|$. Opposed to the metrics d_{E} and d_{S}, this pseudo-metric can assign $d_{\mathrm{P}^*}(c_1, c_2) = 0$ for $c_1 \neq c_2$.

While the definition of Wasserstein barycenters is not entirely rigorous for the pseudo-metric d_{P^*}, the introduced approximate algorithms can still operate on it without problems. Huang et al. [22] also investigated this pseudo-metric for the current policy π, leading to a different metric in each algorithm iteration. We investigate this interesting concept in Appendix C.2 to keep the main article short and consistent with the previous sections that assumed a fixed distance. Figure 6 visualizes the barycentric interpolations generated by d_{E}, d_{S}, and d_{P^*}. Looking at Figure 6, we can already anticipate a detrimental effect of the Euclidean metric d_{E} on the generation of the curriculum. The visualization of d_{P^*} indicates a weakness of purely performance-based metrics since a similar expected return for c_1 and c_2 does not guarantee similar outcomes of actions in the two contexts.

We visualize the expected return for different curricula in Figure 7. As we can see, CURROT and GRADIENT can significantly improve performance over both a purely random- as well as no curriculum. However, the performance gains are highly dependent on an appropriate choice of metric. While both CURROT and GRADIENT show strong performance for d_{S}, CURROT's performance diminishes for d_{P^*}, and none of the two methods can make the agent proficient on $\mu(c)$ when using d_{E}.

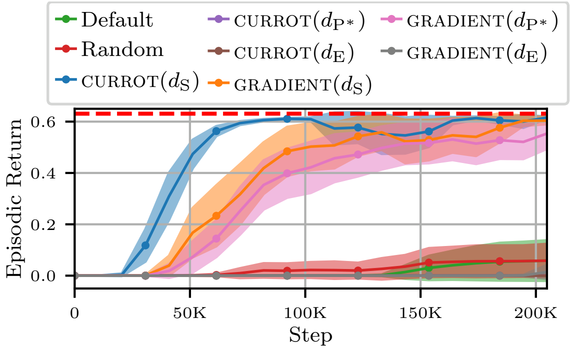


Fig. 7: Expected return on the target task distribution $\mu(c)$ in the E-Maze environment achieved by CURROT and GRADIENT under varying distance metrics. The shaded area corresponds to two times the standard error (computed from 20 seeds). The red dotted line represents the maximum possible reward achievable on $\mu(c)$.

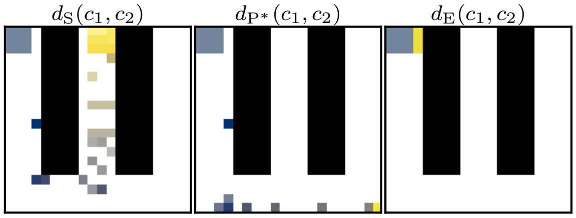


Fig. 8: CURROT sampling distribution without entropy regularization for varying distance measures. Brighter colors correspond to later training iterations.

TABLE 1: Final agent performance of CURROT and GRADIENT on $\mu(c)$ in the E-Maze environment for varying amounts of entropy regularization (λ and H_{LB}). Mean and standard error are computed from 20 seeds.

CURROT

H_{LB}	0.	0.5	1.0	2.0
d_{S}	0.62±0	0.61±0	0.53±0.04	0.58±0.03
d_{P^*}	0±0	0.45±0.06	0.38±0.06	0.42±0.06
d_{E}	0±0	0±0	0±0	0±0

GRADIENT

λ	0.	10^{-8}	10^{-4}	10^{-2}
d_{S}	0.60±0.01	0.56±0.04	0.62±0.00	0.60±0.01
d_{P^*}	0.55±0.03	0.48±0.05	0.45±0.05	0.30±0.06
d_{E}	0.01±0.01	0.03±0.03	0.03±0.03	0.01±0.01

Figure 8 shows interpolations generated by CURROT for the investigated metrics. We see that the interpolating distributions of CURROT can collapse to a Dirac distribution for d_{S} and d_{P^*}. As discussed in Section 5, Huang et al. [22] proposed using an entropy-regularized version of optimal transport due to its computational speed. Given that we solve Objectives (8) and (9) analytically, we can investigate the effect of entropy-regularization not with respect to computational speed but to performance. In Table 1, we show the final agent performance when using entropy-regularized transport plans for GRADIENT as well as a lower bound H_{LB} on the entropy of the generated task distributions for CURROT. The detailed formulations of these variants are provided in Appendix C.2. As the results show, entropy regularization can benefit CURROT. The visualizations in Figure 9 indicate that this benefit arises from avoiding the aggressive targeting of contexts right at the edge of the performance constraint that we can see in Figures 1, 5, and 8. In the case of the pseudo distance d_{P^*}, the more diverse tasks sampled from $p(c)$ sometimes allowed the agent to generalize enough to solve tasks sampled from $\mu(c)$. For

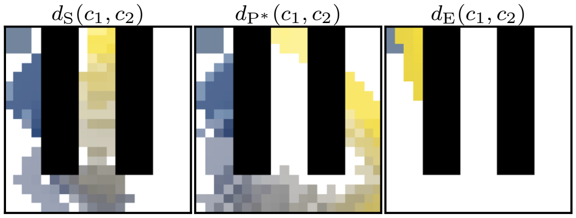


Fig. 9: CURROT sampling distribution for $H_{\mathrm{LB}}{=}2$ and varying distance measures. Brighter colors correspond to later training iterations.

GRADIENT, we cannot see significant performance gains but can observe that a too-high entropy regularization in combination with d_{P^*} diminished performance. Given that for an adequate metric (i.e., d_{S}), the observed performance is stable across different amounts of entropy regularization, we do not further explore this avenue in the following experiments.

6.2 Unlock-Pickup Environment

In the following environment, we aim to benchmark approximate implementations of CURROT and GRADIENT for large discrete context spaces and demonstrate that appropriate distances for non-trivial context spaces can be designed by hand. In Figure 10, we visualize the unlock-pickup environment from the Minigrid environment collection [67] that we chose for this investigation. To master this environment, the agent must pick up a key, unlock a door and eventually pick up a box in the room that has just been unlocked.

We define a curriculum by controlling the starting state of an episode via the context c, i.e., controlling the position of the box, key, agent, and door, as well as the state of the door (whether closed or open). As detailed in Appendix C.3, this task parameterization results in 81.920 tasks to compile a curriculum from. The initial context distribution is defined to encode states in which the agent is directly in front of the box, similar to the bottom-right image in Figure 10. Starting from this initial distribution, the learning algorithm needs to generate a curriculum that ultimately allows the agent to reach and pick up the box from a random position in the left room with a closed door. As we show in Appendix C.3, it is possible to define a so-called highway distance function [68] between contexts that properly takes the role of the door and its interaction with the key into account, without relying on a planning algorithm like in the previous environment. We use this distance function in the following evaluations.

In addition to the approximate versions of CURROT and GRADIENT, we evaluate PLR, VDS, and ACL on this task. We do not evaluate SPRL, ALP-GMM, and GOALGAN since those algorithms have been designed for continuous and Euclidean context spaces by, e.g., leveraging Gaussian distributions, kd-trees, or Gaussian sampling noise. The evaluation results in Figure 11 show that CURROT and GRADIENT consistently allow mastering the target tasks (a cumulative discounted return of $0.75 \approx 0.99^{28}$ is obtained by solving a task in 28 steps). For both CURROT and GRADIENT, each of the 20 runs led to a well-performing policy, and we can barely see any difference in learning speed between the approaches. Learning directly on the target task distribution allows mastering the environment in some runs while failing to do so in others due to the high dependence on collecting enough positive reward signals at the beginning of learning. These two outcomes lead, on average, to a lower performance compared to CURROT and GRADIENT. Finally, we see that all baseline curriculum methods learn slower than directly learning on the target task distribution $\mu(c)$, with ACL not producing policies that collect any reward on the target tasks. Given the successful application of PLR in the Procgen benchmark, which features a diverse set of Arcade game levels with highly distinct visual observations, we wish to discuss the observed low performance of PLR here in more detail. As we show in Appendix C.3, PLR indeed samples contexts occurring under $\mu(c)$ with at least 7% in each run. Furthermore, in about half of the runs, the agent also learns to solve those target tasks that are replayed by PLR at some point in the curriculum. However, these replayed target tasks only make up a small fraction of the total number of target tasks, resulting in low performance on all of $\mu(c)$. The absence of a notion of target distribution for PLR seems to lead to ineffective use of samples w.r.t improving performance on the target. This lack of target distribution causing problems will be a re-occurring theme for the subsequent experiments.

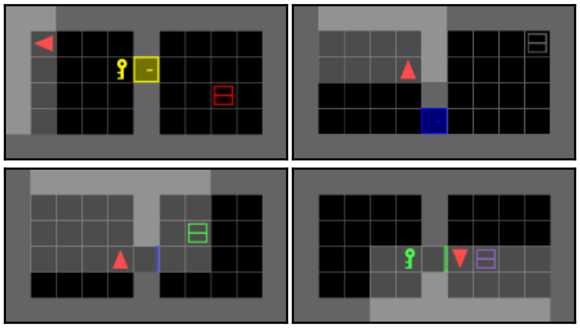


Fig. 10: The Unlock-Pickup environment, in which an agent needs to pick up the box in the right room by unlocking the door. After reset, the agent is randomly placed in the left room not carrying the key (top left image). After picking up the key (top right), the door can be unlocked (bottom left) to move to the box (bottom right). The door-, box- and key positions as well as their colors vary across environment resets. The agent receives a partial view of the world (highlighted rectangle) that is blocked by walls and closed doors.

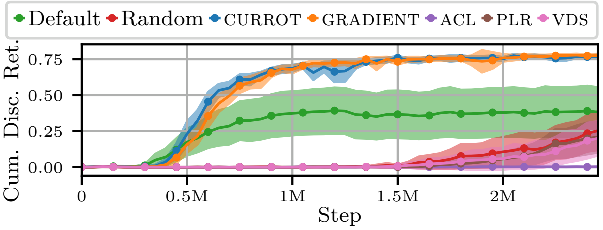


Fig. 11: Episodic return on the target task distribution $\mu(c)$ in the Unlock-Pickup environment for different curricula. The shaded area corresponds to two times the standard error computed from 20 seeds.

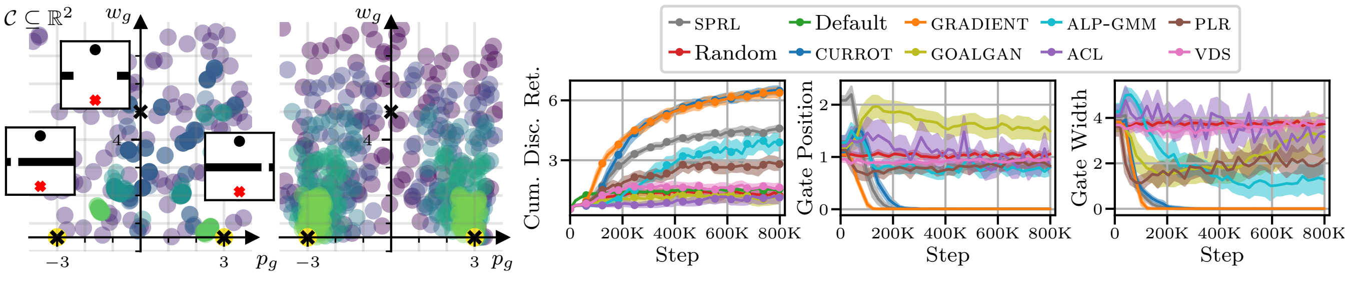
6.3 Point-Mass Environment

In this environment, in which a point-mass agent must pass through a narrow gate to reach a goal position opposite a wall (Figure 12), we benchmark our approximate implementations of CURROT and GRADIENT in continuous settings. The context $\mathbf{c} \in \mathbb{R}^2$ controls the position and width of the gate that the agent needs to pass. This environment has been introduced with the SPRL algorithm by Klink et al. [18] with a Gaussian target distribution that essentially encodes one narrow gate requiring the agent to detour before reaching the target position. Combined with a dense reward based on the Euclidean distance to the goal, the target task is subject to a prominent local minimum that simply moves the agent close to the wall without passing through. We extend this task with a bi-modal target distribution that challenges SPRL's Gaussian restriction that – as we discussed – is required for it to work properly. As seen in Figure 12, CURROT and GRADIENT generate curricula that target both modes of the distribution and allow learning a proficient policy on all of $\mu(\mathbf{c})$. As we show in Appendix C.4, the Gaussian restriction of SPRL's context distribution leads to $p(\mathbf{c})$ matching only one of the modes of $\mu(\mathbf{c})$, resulting in a lower average reward on $\mu(\mathbf{c})$ compared to CURROT and GRADIENT. We additionally visualize summary statistics for the other CRL methods in Figure 12, showing that they result in a less targeted sampling of contexts likely under $\mu(\mathbf{c})$. This observation, in combination with the lower performance compared to CURROT and GRADIENT, once more emphasizes the importance of embedding a notion of target distribution in CRL algorithms.

We additionally benchmark CURROT and GRADIENT in versions of the point-mass environment with increasing context spaces dimensions. The results in Appendix D show that both approaches can scale to higher dimensions (we investigated up to 30-dimensional context spaces) for this environment. However, they also emphasize the importance of certain algorithmic choices such as the choice of initial context distribution $p_0(\mathbf{c})$ for both methods and the choice of the trust region as well as the sampling schemes of candidates for Objective (14) for CURROT. To keep the main article short, we refer the interested reader to Appendix D.

6.4 Sparse Goal-Reaching Environment

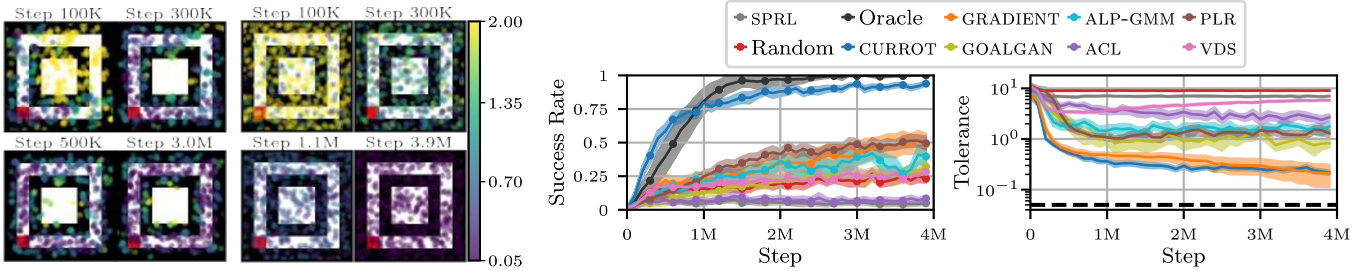
We next turn to a sparse-reward, goal-reaching environment in which an agent needs to reach a desired position with high precision (Figure 13). Such environments have, e.g., been investigated by Florensa et al. [30]. The context $\mathbf{c}{\in}\mathcal{C} \subseteq \mathbb{R}^3$ of this environment encodes the 2D goal position as well as the allowed tolerance for reaching the goal. This parameterization results in both infeasible tasks being part of $\mathcal{C}$ (unreachable regions) as well as tasks that are solely meant to be stepping stones to more complicated ones (low-precision tasks). Given that the agent is ultimately tasked to reach as many goals as possible with the highest precision, i.e., the lowest tolerance, the target distribution $\mu(\mathbf{c})$ is a uniform distribution on a 2D slice of $\mathcal{C}$ with minimal task tolerance. The walls in the environment (Figure 13) render many target tasks infeasible, requiring the curriculum to identify the feasible subspace of tasks to achieve a good learning performance. Figure 13 shows that CURROT results in the best learning performance across all evaluated CRL methods. Only an oracle, which trains the learning agent only on the feasible subspace of high-precision tasks, can reach higher performance. The evolution of the task tolerances shown in Figure 13 highlights that CURROT and GRADIENT continuously reduce the task tolerance. The baseline CRL methods lack focus on the tasks encoded by $\mu(c)$, sampling tasks with comparatively high tolerance even towards the end of training. Interestingly, SPRL samples high-tolerance tasks throughout all training epochs since its Gaussian context distribution converges to a quasi-uniform distribution over $\mathcal{C}$. Otherwise, SPRL would not be able to cover the non-Gaussian distribution of feasible high-precision target tasks without encoding many infeasible tasks. Figure 13 shows the particle evolution for runs of CURROT and GRADIENT. CURROT gradually decreases the goal tolerance over epochs, starting from contexts that are close to the initial position of the agent. Interestingly, it retains higher tolerance contexts located in the walls of the environment even in later epochs due to the trade-off between sampling high-precision tasks and covering all goal positions. The pre-determined interpolation of GRADIENT cannot adjust to infeasible parts of the context space and reduces to a curriculum that shrinks the upper-bound t_{ub} of the tolerance interval $[0.05, t_{\text{ub}}]$. Consequently, a decrease in t_{ub} increases the number of infeasible tasks on which the agent is trained, slowing down learning and resulting in a significant performance gap between CURROT and GRADIENT in this environment. We additionally evaluate CURROT and GRADIENT with Hindsight Experience Replay (HER) [11] in Appendix C.5, showing that HER can serve as a drop-in replacement for SAC in this task.



(a) CURROT (b) GRADIENT (c) Performance Comparison

Fig. 12: a + b) The point-mass environment with its two-dimensional context space. The target distribution $\mu(\mathbf{c})$ encodes the two gates with width $w_g{=}0.5$, in which the agent (black dot) is required to navigate through a narrow gate at different positions to reach the goal (red cross). The colored dots visualize a curriculum generated by CURROT and GRADIENT for this environment. c) Left: Discounted cumulative return over learning epochs obtained in the point mass environment under different curricula as well as baselines that sample tasks uniformly from all of $\mathcal{C}$ (Random) or $\mu(\mathbf{c})$ (Default). Middle and Right: Median minimum distance to the target contexts of $\mu(\mathbf{c})$ for the two dimensions of the context space (i.e., gate position and -width). Mean and two-times standard error intervals are computed from 20 seeds.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3390051



(a) CURROT (b) GRADIENT (c) Performance Comparison

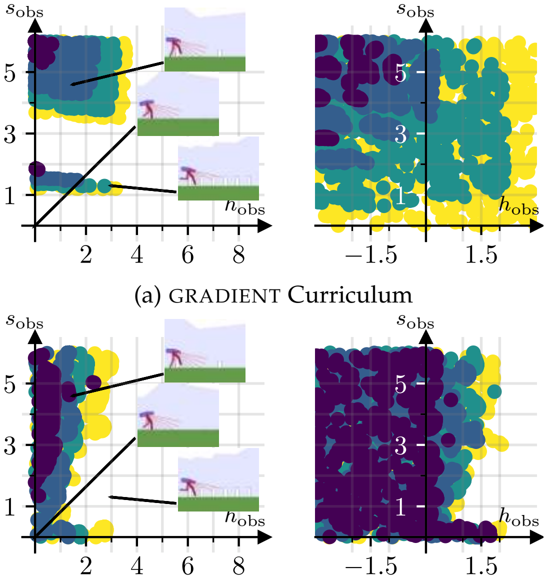
Fig. 13: a + b) Curricula generated by CURROT and GRADIENT in the spare goal-reaching (SGR) environment at different epochs. The starting area of the agent is highlighted in red. The walls are shown in black. The position of the samples encodes the goal to be reached while the color encodes the goal tolerance. c) Success rate on the feasible subspace of $\mathcal{C}$ (left) and median goal tolerance (right) for different CRL methods in the SGR environment. We also include an oracle baseline that only samples the feasible tasks in the context space $\mathcal{C}$. For both plots, mean and two-times standard error intervals are computed from 20 runs.

6.5 Teach My Agent

In this final evaluation environment, a bipedal agent must learn to maneuver over a track of evenly spaced obstacles of a specified height (see Figure 14). The environment is a modified bipedal walker environment introduced by Portelas et al. [14] and extended by Romac et al. [21] in which the spacing and height of obstacles is controlled by the context $\mathbf{c} \in \mathbb{R}^2$. The evaluations by Romac et al. [21] demonstrated poor performance of SPRL, often performing statistically significantly worse than a random curriculum. Given that both CURROT and GRADIENT can be seen as improved versions of SPRL that – among other improvements – explicitly take the geometry of the context space into account, we are interested in whether they can improve upon SPRL.

We hence revisit two learning scenarios investigated by Romac et al. [21], in which CRL methods demonstrated a substantial benefit over random sampling: a setting in which most tasks of the context space are infeasible due to large obstacles and a setting in which most tasks of the context space are trivially solvable. Both scenarios lead to slow learning progress when choosing tasks randomly due to frequently encountering too complex or too simple learning tasks. Given that the uniform initial- and target distribution over the context space lead to poor learning performance, we extend the CURROT and GRADIENT method with a simple randomized search to find areas of $\mathcal{C}$ where the agent achieves returns above δ, similar in spirit to SPRL. We describe this method in Appendix B.

Figure 15 visualizes the performance of CURROT and GRADIENT in comparison to other CRL methods that were already evaluated by Romac et al. [21]. We see that CURROT achieves the best performance in all environments, in one case performing statistically significantly better than ALP-GMM, the best method evaluated in [21]. We also see that the extended version of GRADIENT can improve upon a random curriculum in the "mostly infeasible" scenario while performing insignificantly worse than a random curriculum in the "mostly trivial" scenario. Figure 14 can help shed some light on the observed performance difference between CURROT and GRADIENT. For the "mostly trivial" scenario, GRADIENT consistently arrives at sampling from the uniform $\mu(\mathbf{c})$, whereas CURROT focuses on the contexts at the border of agent competence. For the "mostly infeasible" scenario, the pre-determined interpolation of GRADIENT can fail to encode feasible learning tasks, ultimately leading to a lower overall performance than CURROT.



(a) GRADIENT Curriculum

(b) CURROT Curriculum

Fig. 14: Sampling distribution of GRADIENT and CURROT on the *teach my agent* benchmark in the *no expert knowledge* setting in task spaces with *mostly infeasible-* (left) and *mostly trivial* (right) tasks. The small images visualize the obstacles encoded by the corresponding contexts. For environment details, please see [21]. Brighter colors indicate tasks at later epochs of training. The yellow dots represent the samples from the last generated distribution.

Summarizing, the experimental results underline that empirically successful curricula can be generated by framing CRL as an interpolation between context distributions. The leap in performance between GRADIENT and CURROT compared to SPRL and the performance differences between GRADIENT and CURROT underline the tremendous impact of design choices, such as the distributional measure of similarity and the way of incorporating performance constraints,

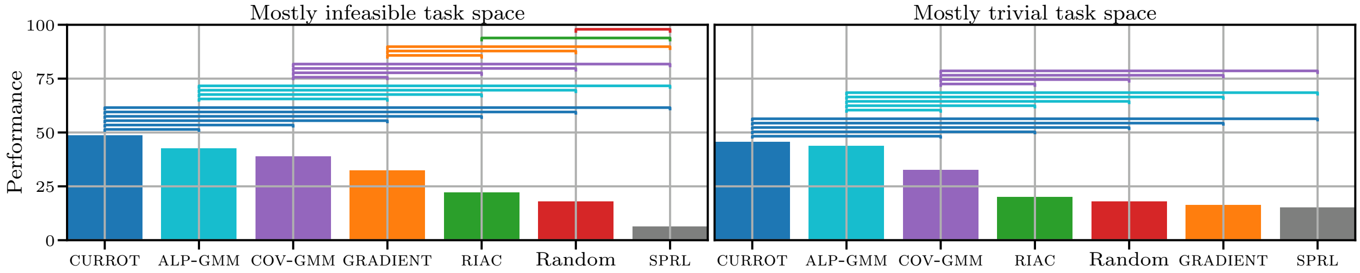


Fig. 15: Performance (in percentage of solved tasks) in the *Teach My Agent* benchmark in the *no expert knowledge* setting. The baseline results are taken from [21], and only CURROT and GRADIENT are evaluated by us. Statistics have been computed from 32 seeds. Horizontal lines between connecting two methods indicate statistically significant different performances according to Welch's t-test with $p < 0.05$.

on the final algorithm performance. However, when chosen correctly, these curricula exhibit strong performance and allow for guiding training towards tasks of interest specified via $\mu(\mathbf{c})$. Especially this last aspect can allow for more flexibility in the curriculum design, as it is possible to define auxiliary task parameterizations without jeopardizing learning progress toward tasks of interest. We saw an example of this trade-off in the sparse goal-reaching environment, where the additional precision parameter boosted the performance of CURROT while diminishing the performance of other CRL methods.

7 Conclusion

In this article, we framed curriculum reinforcement learning as an interpolation between distributions of initial- and target tasks. We demonstrated that the lack of an explicit notion of task similarity in combination with an expected performance constraint makes existing approaches highly dependent on the parameterization of the interpolating task distribution. We avoided these pitfalls by explicitly encoding task similarity via an optimal transport formulation, and by restricting the generated task distributions to only encode tasks that satisfy a specified performance threshold. The resulting method called CURROT led to good performance in experiments due to its focus on tasks at the performance threshold and the adaptive nature of the curriculum. Contrasting our approach to a recently proposed method that generates curricula via Wasserstein barycenters between initial- and target task distributions [22], we saw that the more adaptive nature of our formulation resulted in better performance when facing learning settings with infeasible target tasks. In tasks, in which infeasibility is not a concern, both methods performed similar. In Appendix D, we saw that both methods can scale to higher dimensional tasks although the conceptually more simple GRADIENT algorithm requires less adaptations of its approximations to do so. Together, both methods demonstrate the benefit of using optimal transport for curriculum RL and we believe that this benefit can be maximized by developing algorithms that combine the adaptivity of CURROT with the simpler algorithmic realization of GRADIENT. Additionally, we believe that the precise notion of task similarity via the distance $d(\mathbf{c}_1, \mathbf{c}_2)$ can prove beneficial in advancing the understanding of curriculum RL. We already saw that an appropriate definition of task similarity is key to successful curriculum learning. We believe that distances learned from experience, which encode a form of intrinsic motivation, will significantly advance these methods by merging the strong empirical results of intrinsic motivation in open-ended learning scenarios [13] with the targeted learning achieved by CURROT and GRADIENT.

Acknowledgments

This project has received funding from the DFG project PA3179/1-1 (ROBOLEAP) and the German Federal Ministry of Education and Research (BMBF) (Project: 01IS22078). This work was also funded by Hessian.ai through the project 'The Third Wave of Artificial Intelligence – 3AI' by the Ministry for Science and Arts of the state of Hessen. Finally, the authors gratefully acknowledge the computing time provided to them on the high-performance computer Lichtenberg at the NHR Centers NHR4CES at TU Darmstadt. This is funded by the Federal Ministry of Education and Research and the state governments participating on the basis of the resolutions of the GWK for national high-performance computing at universities. Joni Pajarinen was supported by Research Council of Finland (formerly Academy of Finland) (decision 345521).

References

[1] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. MIT Press, 1998.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[3] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[4] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[5] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*, 2022.

[6] S. Liu, G. Lever, Z. Wang, J. Merel, S. Eslami, D. Hennes, W. M. Czarnecki, Y. Tassa, S. Omidshafiei, A. Abdolmaleki *et al.*, "From motor control to team play in simulated humanoid football," *arXiv preprint arXiv:2105.12196*, 2021.

[7] M. Bellemare, S. Srinivasan, G. Ostrovski, T. Schaul, D. Saxton, and R. Munos, "Unifying count-based exploration and intrinsic motivation," in *Neural Information Processing Systems (NeurIPS)*, 2016.

[8] M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar, "Bayesian reinforcement learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 8, no. 5-6, pp. 359–483, 2015.

[9] M. C. Machado, M. G. Bellemare, and M. Bowling, "Count-based exploration with the successor representation," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[10] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, "Curriculum learning for reinforcement learning domains: A framework and survey," *Journal of Machine Learning Research (JMLR)*, vol. 21, no. 181, pp. 1–50, 2020.

[11] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Neural Information Processing Systems (NeurIPS)*, 2017.

[12] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel, "Reverse curriculum generation for reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2017.

[13] R. Wang, J. Lehman, J. Clune, and K. O. Stanley, "Poet: open-ended coevolution of environments and their optimized solutions," in *Genetic and Evolutionary Computation Conference (GECCO)*, 2019, pp. 142–151.

[14] R. Portelas, C. Colas, K. Hofmann, and P.-Y. Oudeyer, "Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments," in *Conference on Robot Learning (CoRL)*, 2019.

[15] J. Wöhlke, F. Schmitt, and H. van Hoof, "A performance-based start state curriculum framework for reinforcement learning," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2020, pp. 1503–1511.

[16] M. Jiang, M. Dennis, J. Parker-Holder, J. Foerster, E. Grefenstette, and T. Rocktäschel, "Replay-guided adversarial environment design," in *Neural Information Processing Systems (NeurIPS)*, 2021.

[17] P. Klink, H. Abdulsamad, B. Belousov, C. D'Eramo, J. Peters, and J. Pajarinen, "A probabilistic interpretation of self-paced learning with applications to reinforcement learning," *Journal of Machine Learning Research (JMLR)*, vol. 22, no. 182, pp. 1–52, 2021.

[18] P. Klink, H. Abdulsamad, B. Belousov, and J. Peters, "Self-paced contextual reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2020.

[19] P. Klink, C. D' Eramo, J. R. Peters, and J. Pajarinen, "Self-paced deep reinforcement learning," in *Neural Information Processing Systems (NeurIPS)*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., 2020.

[20] J. Chen, Y. Zhang, Y. Xu, H. Ma, H. Yang, J. Song, Y. Wang, and Y. Wu, "Variational automatic curriculum learning for sparse-reward cooperative multi-agent problems," *Neural Information Processing Systems (NeurIPS)*, 2021.

[21] C. Romac, R. Portelas, K. Hofmann, and P.-Y. Oudeyer, "Teachmyagent: a benchmark for automatic curriculum learning in deep rl," *International Conference on Machine Learning (ICML)*, 2021.

[22] P. Huang, M. Xu, J. Zhu, L. Shi, F. Fang, and D. Zhao, "Curriculum reinforcement learning using optimal transport via gradual domain adaptation," in *Neural Information Processing Systems (NeurIPS)*, 2022.

[23] D. Weinshall and D. Amir, "Theory of curriculum learning, with convex loss functions," *Journal of Machine Learning Research (JMLR)*, vol. 21, no. 222, pp. 1–19, 2020.

[24] X. Wu, E. Dyer, and B. Neyshabur, "When do curricula work?" in *International Conference on Learning Representations (ICLR)*, 2021.

[25] Q. Li, Y. Zhai, Y. Ma, and S. Levine, "Understanding the complexity gains of single-task rl with a curriculum," in *International Conference on Machine Learning (ICML)*, 2023.

[26] A. Allievi, P. Stone, S. Niekum, S. Booth, and W. B. Knox, "The perils of trial-and-error reward design: Misdesign through overfitting and invalid task specifications," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

[27] M. Dennis, N. Jaques, E. Vinitsky, A. Bayen, S. Russell, A. Critch, and S. Levine, "Emergent complexity and zero-shot transfer via unsupervised environment design," in *Neural Information Processing Systems (NeurIPS)*, 2020.

[28] M. Jiang, E. Grefenstette, and T. Rocktäschel, "Prioritized level replay," in *International Conference on Machine Learning (ICML)*, 2021.

[29] S. Sukhbaatar, Z. Lin, I. Kostrikov, G. Synnaeve, A. Szlam, and R. Fergus, "Intrinsic motivation and automatic curricula via asymmetric self-play," in *International Conference on Learning Representations (ICLR)*, 2018.

[30] C. Florensa, D. Held, X. Geng, and P. Abbeel, "Automatic goal generation for reinforcement learning agents," in *International Conference on Machine Learning (ICML)*, 2018.

[31] Y. Zhang, P. Abbeel, and L. Pinto, "Automatic curriculum learning through value disagreement," in *Neural Information Processing Systems (NeurIPS)*, 2020.

[32] S. Racaniere, A. K. Lampinen, A. Santoro, D. P. Reichert, V. Firoiu, and T. P. Lillicrap, "Automated curricula through setter-solver interactions," in *International Conference on Learning Representations (ICLR)*, 2020.

[33] T. Eimer, A. Biedenkapp, F. Hutter, and M. Lindauer, "Self-paced context evaluation for contextual reinforcement learning," in *International Conference on Machine Learning (ICML)*, 2021.

[34] A. Baranes and P.-Y. Oudeyer, "Intrinsically motivated goal exploration for active motor learning in robots: A case study," in *International Conference on Intelligent Robots and Systems (IROS)*, 2010.

[35] M. P. Kumar, B. Packer, and D. Koller, "Self-paced

learning for latent variable models," in *Neural Information Processing Systems (NeurIPS)*, 2010.

[36] R. M. Neal, "Annealed importance sampling," *Statistics and Computing*, vol. 11, no. 2, pp. 125–139, 2001.

[37] E. L. Allgower and K. Georg, *Introduction to numerical continuation methods*. SIAM, 2003.

[38] G. Peyré, M. Cuturi *et al.*, "Computational optimal transport: With applications to data science," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[39] Y. Chen, T. T. Georgiou, and M. Pavon, "Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a schrödinger bridge," *SIAM Review (SIREV)*, vol. 63, no. 2, pp. 249–313, 2021.

[40] L. Kantorovich, "On the transfer of masses (in russian)," *Doklady Akademii Nauk*, vol. 37, no. 2, pp. 227–229, 1942.

[41] C. Liu, J. Zhuo, P. Cheng, R. Zhang, J. Zhu, and L. Carin, "Understanding and accelerating particle-based variational inference," in *International Conference on Machine Learning (ICML)*, 2019.

[42] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde, "Optimal mass transport: Signal processing and machine-learning applications," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 43–59, 2017.

[43] K. Kandasamy, W. Neiswanger, J. Schneider, B. Poczos, and E. P. Xing, "Neural architecture search with bayesian optimisation and optimal transport," in *Neural information processing systems (NeurIPS)*, 2018.

[44] M. Togninalli, E. Ghisu, F. Llinares-López, B. Rieck, and K. Borgwardt, "Wasserstein weisfeiler-lehman graph kernels," in *Neural Information Processing Systems (NeurIPS)*, 2019.

[45] F. Mémoli, "Gromov–wasserstein distances and the metric approach to object matching," *Foundations of computational mathematics*, vol. 11, no. 4, pp. 417–487, 2011.

[46] C. Vincent-Cuaz, R. Flamary, M. Corneli, T. Vayer, and N. Courty, "Semi-relaxed gromov-wasserstein divergence and applications on graphs," in *International Conference on Learning Representations (ICLR)*, 2022.

[47] P. Demetci, R. Santorella, B. Sandstede, W. S. Noble, and R. Singh, "Gromov-wasserstein optimal transport to align single-cell multi-omics data," in *ICML 2020 Workshop on Computational Biology*, 2020.

[48] A. Fickinger, S. Cohen, S. Russell, and B. Amos, "Cross-domain imitation learning via optimal transport," in *International Conference on Learning Representations (ICLR)*, 2022.

[49] R. Zhang, C. Chen, C. Li, and L. Carin, "Policy optimization as wasserstein gradient flows," in *International Conference on Machine Learning (ICML)*, 2018.

[50] A. M. Metelli, A. Likmeta, and M. Restelli, "Propagating uncertainty in reinforcement learning via wasserstein barycenters," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[51] L. Chen, K. Bai, C. Tao, Y. Zhang, G. Wang, W. Wang, R. Henao, and L. Carin, "Sequence generation with optimal-transport-enhanced reinforcement learning," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[52] Y. L. Goh, W. S. Lee, X. Bresson, T. Laurent, and N. Lim, "Combining reinforcement learning and optimal transport for the traveling salesman problem," in *1st International Workshop on Optimal Transport and Structured Data Modeling*, 2022.

[53] Z. Ren, K. Dong, Y. Zhou, Q. Liu, and J. Peng, "Exploration via hindsight goal generation," *Neural Information Processing Systems (NeurIPS)*, 2019.

[54] I. Durugkar, M. Tec, S. Niekum, and P. Stone, "Adversarial intrinsic motivation for reinforcement learning," 2021.

[55] D. Cho, S. Lee, and H. J. Kim, "Outcome-directed reinforcement learning by uncertainty & temporal distance-aware curriculum goal generation," *International Conference on Learning Representations (ICLR)*, 2023.

[56] P. Klink, H. Yang, C. D'Eramo, J. Peters, and J. Pajarinen, "Curriculum reinforcement learning via constrained optimal transport," in *International Conference on Machine Learning*. PMLR, 2022, pp. 11 341–11 358.

[57] A. Hallak, D. Di Castro, and S. Mannor, "Contextual markov decision processes," *arXiv preprint arXiv:1502.02259*, 2015.

[58] A. Wilson, A. Fern, S. Ray, and P. Tadepalli, "Multi-task reinforcement learning: a hierarchical bayesian approach," in *International Conference on Machine Learning (ICML)*, 2007.

[59] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal value function approximators," in *International Conference on Machine Learning (ICML)*, 2015.

[60] G. Monge, "Mémoire sur la théorie des déblais et des remblais," *De l'Imprimerie Royale*, 1781.

[61] M. Agueh and G. Carlier, "Barycenters in the wasserstein space," *SIAM Journal on Mathematical Analysis*, vol. 43, no. 2, pp. 904–924, 2011.

[62] D. Meng, Q. Zhao, and L. Jiang, "A theoretical understanding of self-paced learning," *Information Sciences*, vol. 414, pp. 319–328, 2017.

[63] J. Feydy, T. Séjourné, F.-X. Vialard, S.-i. Amari, A. Trouve, and G. Peyré, "Interpolating between optimal transport and mmd using sinkhorn divergences," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[64] E. A. Nadaraya, "On estimating regression," *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[65] G. S. Watson, "Smooth regression analysis," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372, 1964.

[66] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu, "Automated curriculum learning for neural networks," in *International Conference on Machine Learning (ICML)*, 2017.

[67] M. Chevalier-Boisvert, L. Willems, and S. Pal, "Minimalistic gridworld environment for gymnasium," 2018. [Online]. Available: https://github.com/Farama-Foundation/Minigrid

[68] E. Baikousi, G. Rogkakos, and P. Vassiliadis, "Similarity measures for multidimensional data," in *International Conference on Data Engineering (ICDE)*, 2011.

[69] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment

problems," *Computing*, vol. 38, no. 4, pp. 325–340, 1987.

[70] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[71] J. Feydy and P. Roussillon, "Geomloss," 2019. [Online]. Available: https://www.kernel-operations.io/geomloss/index.html

[72] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, "Sliced and radon wasserstein barycenters of measures," *Journal of Mathematical Imaging and Vision*, vol. 51, no. 1, pp. 22–45, 2015.

[73] S. Kolouri, K. Nadjahi, U. Simsekli, R. Badeau, and G. Rohde, "Generalized sliced wasserstein distances," *Neural Information Processing Systems (NeurIPS)*, 2019.

[74] N. Courty, R. Flamary, and M. Ducoffe, "Learning wasserstein embeddings," in *International Conference on Learning Representations (ICLR)*, 2018.

[75] L. Li, A. Genevay, M. Yurochkin, and J. M. Solomon, "Continuous regularized wasserstein barycenters," *Neural Information Processing Systems (NeurIPS)*, 2020.

[76] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[77] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, "Iterative bregman projections for regularized transportation problems," *SIAM Journal on Scientific Computing*, vol. 37, no. 2, pp. A1111–A1138, 2015.

[78] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Neural Information Processing Systems (NeurIPS)*, 2013.

[79] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

Pascal Klink is a Ph.D. student with Jan Peters and Joni Pajarinen at the Institute for Intelligent Autonomous Systems (IAS) at the Techincal University of Darmstadt since May 2019. In his Ph.D., Pascal focuses on improving the learning performance of reinforcement learning agents by leveraging experience across learning tasks via curricula. He completed a research internship at Amazon Robotics and received the AI newcomer award (2021) from the German computer science foundation. Before this, Pascal received his M.Sc. from the Technical University of Darmstadt, where he also worked as a student assistant.

Carlo D'Eramo is an Associate Professor for Reinforcement Learning and Computational Decision-Making at the Center for Artificial Intelligence and Data Science of Julius-Maximilians-Universität Würzburg. He is also an independent group leader of hessian.AI. The research of his LiteRL group revolves around the problem of how agents can efficiently acquire expert skills that account for the complexity of the real world. To answer this question, his group investigates lightweight methods to obtain adaptive autonomous agents, focusing on several RL topics, including multi-task, curriculum, adversarial, options, and multi-agent RL.

Jan Peters is a full professor (W3) for Intelligent Autonomous Systems at the Computer Science Department of the Technical University of Darmstadt since 2011 and, at the same time, he is the dept head of the research department on Systems AI for Robot Learning (SAIROL) at the German Research Center for Artificial Intelligence (Deutsches Forschungszentrum für Künstliche Intelligenz, DFKI) since 2022. He is also a founding research faculty member of The Hessian Center for Artificial Intelligence. He has received the Dick Volz Best 2007 US Ph.D. Thesis Runner-Up Award, Robotics: Science & Systems - Early Career Spotlight, INNS Young Investigator Award, and IEEE Robotics & Automation Society's Early Career Award, as well as numerous best paper awards. He received an ERC Starting Grant and was appointed an IEEE fellow, AIAA fellow and ELLIS fellow.

Joni Pajarinen is an Assistant Professor at Aalto University, where he leads the Aalto Robot Learning research group. The research group focuses on making robots capable of operating autonomously alongside humans by helping them understand what they need to learn in order to perform their assigned tasks. To this end, the group focuses on developing novel decision-making methods in reinforcement learning, planning under uncertainty, and decision-making in multi-agent systems.

Appendix A
Computational Cost of Optimal Transport

The benefits of optimal transport (OT), such as explicitly incorporating a ground distance on the sample space, come at the price of a relatively high computational burden caused by the need to solve an optimization problem to compute the Wasserstein distance between two distributions. In practice, OT problems in continuous spaces (such as some of the context spaces investigated in this article) are often reduced to linear assignment problems between sets of particles. Such assignment problems can be exactly solved with variations of the Hungarian algorithm with a time complexity of $\mathcal{O}(n^3)$ [69]. While this polynomial complexity ultimately leads to prohibitive runtimes for large n, we can typically avoid this problem for curriculum RL. Given the often moderate dimensionality of the chosen context spaces, a few hundred particles are typically sufficient to represent the context distributions. In our experiments, we used less than 500 particles in the continuous environments and 640 particles for the discrete unlock-pickup environment, leading to observed solving times of less than 200ms with the `linear_sum_assignment` function of the SciPy library [70] on an AMD Ryzen 9 3900X. Since the CURROT and GRADIENT algorithms solve, at most, three OT problems per context distribution update, the computational costs of OT are relatively small for the investigated environments.

Furthermore, approximations have emerged to tackle problems that require a large number of particles. For example, the GeomLoss library [71], which we use in the GRADIENT implementations for continuous Euclidean spaces, implements a variant of entropy-regularized OT that has brought down the computation time of OT for sets of hundreds of thousands of samples to seconds on high-end GPUs [63]. So-called sliced Wasserstein distances [72, 73] approximately solve the given OT problem by solving M OT problems in 1-D subspaces, reducing the time complexity to $\mathcal{O}(Mn\log(n))$, where typically $M \ll n$. Finally, neural function approximators have been employed e.g. to speed up the computation of Wasserstein distances by learning a metric embedding from data [74] or enable to computation of regularized free-support Wasserstein barycenters by approximating the dual potentials [75]. Consequently, we see opportunities to significantly increase the number of particles via such approximate approaches, even though our experiments did not indicate a need for that so far.

Appendix B
Search for Feasible Contexts

As detailed in Section 6.5, the initial context distribution $p_0(\mathbf{c})$ may be uninformed and consequently lead to sampling many learning tasks for which the agent performance is below δ. In such scenarios, we can initiate a search procedure for tasks in which the current agent achieves a performance at least δ of as long as $\bar{R} = \frac{1}{M}\sum_{m=1}^{M} R_m < \delta$. We terminate this search procedure as soon as $\bar{R} \geq \delta$. During this search, $\mathcal{D}_+$ contains the best-encountered samples, and $\mathcal{D}_-$ is empty. When a batch of M new episodes arrives, we add those episodes whose return is at least as large as the median return in $\mathcal{D}_+$ to the buffer – and for each new episode added, remove the worst performing episode. The search distribution is a (truncated) Gaussian mixture model

$$p_{\text{search}}(\mathbf{c}) = \sum_{i=1}^{N_{\mathcal{D}}} w_i \mathcal{N}\left(\mathbf{c} \middle| \mathbf{c}_i, \sigma_i^2 \mathbf{I}\right)$$

with weights w_i and variances σ_i^2 defined via the minimum return observed over all episodes R_{min} and the median performance of the buffered episodes R_{med}

$$w_i \propto \max(0, R_{\mathbf{c}_i} - R_{\text{med}}), \quad \sigma_i = \max\left(10^{-3}, 2\frac{\delta - R_{\mathbf{c}_i}}{\delta - R_{\text{min}}}\right).$$

For simplicity of exposition, we assume that $\mathcal{C} = [0, 1]^d$, i.e., that the context space is a d-dimensional hyper-cube of edge-length one. Consequently, a context $\mathbf{c}$ with a return of R_{min} will have a standard deviation of two in each dimension, which, in combination with the Gaussian being truncated, leads to spread-out sampling across the hypercube. If the dimensions of $\mathcal{C}$ are scaled differently, a simple re-scaling is sufficient to use the above sampling procedure. As detailed in the main article, we only required the search procedure in the teach my agent environments, as in the other environments $p_0(\mathbf{c})$ provided enough successful initial episodes. For discrete context spaces, the search distribution would need to be adapted, e.g., by defining a uniform distribution over all contexts $\mathbf{c}$ with a distance $d(\mathbf{c}, \mathbf{c}_i)$ less that or equal to a threshold that is similarly scaled as the variance σ_i^2.

Appendix C
Experimental Details

This section discusses hyperparameters and additional details of the conducted experiments that could not be provided in the main text due to space limitations. For all experiments except the *teach my agent* benchmark, we used RL algorithms from the `Stable Baselines 3` library [76]. For *teach my agent*, we use the SAC implementation provided with the benchmark.

C.1 Algorithm Hyperparameters

The main parameters of SPRL, CURROT, and GRADIENT all factor into one parameter δ corresponding to the performance constraint and one parameter ϵ controlling the interpolation speed. We did not perform an extensive hyperparameter search for these parameters but used their interpretability to select appropriate parameter regions to search in. The performance parameter δ was chosen by evaluating values around 50% of the maximum reward. This approach resulted in a search over $\delta \in \{3, 4, 5\}$ for the point-mass environment and $\delta \in \{0.4, 0.6, 0.8\}$ for the sparse goal-reaching and unlock-pickup environment. For the *teach my agent* experiments, we evaluated $\delta \in \{140, 160, 180\}$ for CURROT and GRADIENT. We did not evaluate SPRL in the *teach my agent* experiment since we took the results from Romac et al. [21]. We evaluated GRADIENT for $\epsilon \in [0.05, 0.1, 0.2]$. For SPRL, we initialized ϵ with a value of 0.05 used in the initial experiments by Klink et al. However, we realized that larger values slightly improved performance. For CURROT, the value of ϵ depends on the magnitude of the distances d and hence changes per experiment. In the

	SPRL				CURROT		GRADIENT	
ENV.	δ	ϵ	$\boldsymbol{\sigma}_{\text{LB}}$	$D_{\text{KL}_{\text{LB}}}$	δ	ϵ	δ	ϵ
SPARSE GOAL-REACHING	0.6	.25	-	-	0.8	1.2	0.6	0.05
POINT MASS	4	.25	[.2 .1875]	8000	4	0.7	3.0	0.2
UNLOCK-PICKUP	-	-	-	-	0.6	3	0.6	0.05
TEACH MY AGENT	-	-	-	-	180	0.5|0.4	180	0.05

TABLE 2: Hyperparameters of SPRL, CURROT, and GRADIENT in the different learning environments. The ϵ parameter of CURROT is computed according to the procedure described in appendix C. We do not provide *teach my agent* parameters for SPRL as we rely on the results reported by [21]. We also do not evaluate SPRL in the unlock-pickup environment since SPRL is designed for continuous context spaces.

conducted experiments, we set the parameter ϵ to around 5% of the maximum distance between any two points in the context space, also evaluating a slightly larger and smaller value. However, we refer to Appendix D for a detailed discussion of how to chose ϵ particularly when dealing with higher dimensional context spaces. When targeting narrow target distributions, Klink et al. introduce a lower bound on the standard deviation $\boldsymbol{\sigma}_{\text{lb}}$ of the context distribution of SPRL. This lower bound needs to be respected until the KL divergence w.r.t. $\mu(\mathbf{c})$ falls below a threshold D_{KL}, as otherwise, the variance of the context distribution may collapse too early, causing the KL divergence constraint on subsequent distributions to only allow for minimal changes to the context distribution. This detail again highlights the benefit of Wasserstein distances, as they are not subject to such subtleties due to their reliance on a chosen metric. Table 2 shows the parameters of CURROT, GRADIENT, and SPRL for the different environments.

For ALP-GMM, the relevant hyperparameters are the percentage of random samples drawn from the context space p_{rand}, the number of completed learning episodes between the update of the context distribution n_{rollout}, and the maximum buffer size of past trajectories to keep s_{buffer}. Similar to Klink et al. [17], we chose them by a grid-search over $(p_{\text{rand}}, n_{\text{rollout}}, s_{\text{buffer}}) \in \{0.1, 0.2, 0.3\} \times \{50, 100, 200\} \times \{500, 1000, 2000\}$.

For GOALGAN, we tuned the amount of random noise that is added on top of each sample δ_{noise}, the number of policy rollouts between the update of the context distribution n_{rollout} as well as the percentage of samples drawn from the success buffer p_{success} via a grid search over $(\delta_{\text{noise}}, n_{\text{rollout}}, p_{\text{success}}) \in \{0.025, 0.05, 0.1\} \times \{50, 100, 200\} \times \{0.1, 0.2, 0.3\}$.

For ACL, the continuous context spaces of the environments need to be discretized, as the algorithm is formulated as a bandit problem. The Exp3.S bandit algorithm that ultimately realizes the curriculum requires two hyperparameters to be chosen: the scale factor for updating the arm probabilities η and the ϵ parameter of the ϵ-greedy exploration strategy. We combine ACL with the absolute learning progress (ALP) metric also used in ALP-GMM and conducted a hyperparameter search over $(\eta, \epsilon) \in \{0.05, 0.1, 0.2\} \times \{0.01, 0.025, 0.05\}$. Hence, contrasting ACL and ALP-GMM sheds light on the importance of exploiting the continuity of the context space. For ACL, the absolute learning progress in a context $\mathbf{c}$ can be estimated by keeping track of the last reward obtained in the bin of $\mathbf{c}$ (note that we discretize the context space) and then computing the absolute difference between the return obtained from the current policy execution and the stored last reward. We had numerical issues when implementing the ACL algorithm by Graves et al. [66] due to the normalization of the ALPs via quantiles. Consequently, we normalized via the maximum and minimum ALP seen over the entire history of tasks.

For PLR, the staleness coefficient ρ, the score temperature β, and the replay probability p need to be chosen. We did a grid search over $(\rho, \beta, p) \in \{0.15, 0.3, 0.45\} \times \{0.15, 0.3, 0.45\} \times \{0.55, 0.7, 0.85\}$ and chose the best configuration for each environment.

For VDS, the parameters for the training of the Q-function ensemble, i.e., the learning rate lr, the number of epochs

	ALP-GMM			GOALGAN			ACL	
ENV.	p_{RAND}	n_{ROLLOUT}	s_{BUFFER}	δ_{NOISE}	n_{ROLLOUT}	p_{SUCCESS}	η	ϵ
SPARSE GOAL-REACHING	.2	200	500	.1	200	.2	0.05	0.2
POINT MASS	.1	100	500	.1	200	.2	0.025	0.2
UNLOCK-PICKUP	-	-	-	-	-	-	0.025	0.1

	PLR			VDS		
ENV.	ρ	β	p	LR	n_{EP}	n_{BATCH}
SPARSE GOAL-REACHING	.45	.15	.55	$5{\times}10^{-4}$	10	80
POINT MASS	.15	.45	.85	10^{-3}	3	20
UNLOCK-PICKUP	.45	.45	.55	10^{-3}	5	20

TABLE 3: Hyperparameters of the investigated baseline algorithms in the different learning environments, as described in Appendix C.

n_{ep} and the number of mini-batches n_{batch}, need to be chosen. Just as for PLR, we conducted a grid search over $(\text{lr}, n_{\text{ep}}, n_{\text{batch}}) \in \{10^{-4}, 5{\times}10^{-4}, 10^{-3}\} \times \{3, 5, 10\} \times \{20, 40, 80\}$. The parameters of all employed baselines are given in Table 3. We now continue with the description of experimental details for each environment.

C.2 E-Maze Environment

The xy-coordinates of the representatives

$$\mathbf{r}(c){=}[x,\ y,\ z]^T \in \mathbb{R}^3$$

of a context c form a grid on $[-1, 1] \times [-1, 1]$ and, as mentioned in the main article, $z{=}200$ for walls and $z{=}0$ for all other cells. The four actions $\{\text{up}, \text{down}, \text{left}, \text{right}\}$ lead to a transition to the corresponding neighboring cell with a probability of 0.9, if the neighboring cell has the same height, and 0 if not. Upon reaching the desired state (controlled by the context c), the agent observes a reward of value one, and the episode terminates. In this environment, we use PPO with $\lambda = 0.99$ and all other parameters left to the implementation defaults of the `Stable Baselines 3` library.

For solving objectives (8) and (9), we make use of the `linprog` function from the `SciPy` library [70].

Current Agent Performance as a Distance: In the main text, we have investigated the pseudo-distance

$$d_{\text{P}^*}(c_1, c_2) = |J(\pi^*, c_1) - J(\pi^*, c_2)| \tag{15}$$

that defines the similarity of contexts based on the absolute performance difference of the optimal policy in the contexts c_1 and c_2. While d_{P^*} only performed slightly worse than the more informed distance d_{S} for GRADIENT, it could only provide meaningful performance for CURROT if combined with entropy regularization. However, Huang et al. [22] also investigated a pseudo-distance function that computes the similarity of two contexts based on the *current* policy π

$$d_{\text{P}}(c_1, c_2) = |J(\pi, c_1) - J(\pi, c_2)|, \tag{16}$$

leading to a distance function that changes in each iteration. As we show in Figure 16, this distance, while still leading to slower learning for CURROT compared to d_{S}, leads to stable learning across different levels of entropy regularization without any prior environment knowledge. Figure 17 shows multiple curricula that have been generated by CURROT and GRADIENT. Particularly for CURROT, we can see fairly diverse curricula, which sometimes target all three corridors at once (top middle) and sometimes even back track out of the right-most corridor into the remaining two (top right). We see the good performance and the diverse behavior as indicators for the potential of general purpose distance metrics that encode some form of implicit exploration, calling for future investigations to better understand their mechanics. Furthermore, computational aspects arise with the use of such metrics, since for the case of d_{P}, robust and efficient versions for estimating $J(\pi, c)$ need to be devised.

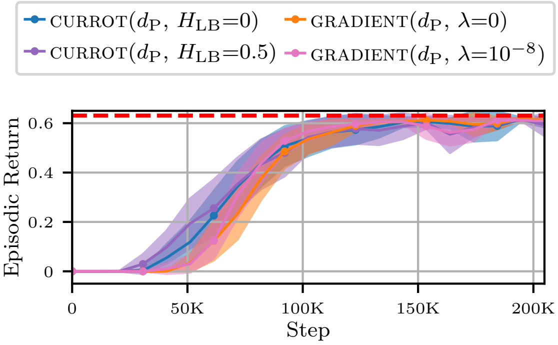


Fig. 16: Expected return on the target task distribution $\mu(c)$ in the E-Maze environment achieved by CURROT and GRADIENT under varying entropy regularizations for the current performance-based distance d_{P}. The shaded area corresponds to two times the standard error (computed from 20 seeds). The red dotted line represents the maximum possible reward achievable on $\mu(c)$.

Entropy-Regularized CURROT and GRADIENT: As discussed in Section 6.1, we benchmark versions of GRADIENT and CURROT in which we introduce different forms of entropy regularization. For GRADIENT, we recreate the implementation by Huang et al. [22] by using optimal transport formulations that regularize the entropy of the transport plan ϕ [77, 78]

$$\mathcal{W}_{p,\lambda}(p_1, p_2){=}\left(\inf_{\phi \in \Phi(p_1, p_2)} \mathbb{E}_\phi\left[d(\mathbf{c}_1, \mathbf{c}_2)^p\right] - \lambda H(\phi)\right)^{1/p}, \tag{17}$$

with the constraint set $\Phi(p_1, p_2)$ defined as in Section 3.3 and the entropy $H(p)$ of a distribution p over a sample space $\mathcal{X}$ defined as $H(p) = -\int_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}) \log(p(\mathbf{x}))$. Note that Huang et al. [22] chose these formulations for computational speed rather than curriculum performance. This formulation allows for a straightforward adaptation of the GRADIENT objective to incorporate entropy-regularization

$$\max_{\alpha \in [0,1]} \ \alpha \quad \text{s.t. } J(\pi, p_{\alpha,\lambda}) \geq \delta \tag{18}$$

$$p_{\alpha,\lambda}(c) = \arg\min_p \ \alpha \mathcal{W}_{2,\lambda}(p, \mu) + (1-\alpha)\mathcal{W}_{2,\lambda}(p, p_0). \tag{19}$$

For the CURROT algorithm, we choose a more direct form of regularization and directly constrain the entropy of the

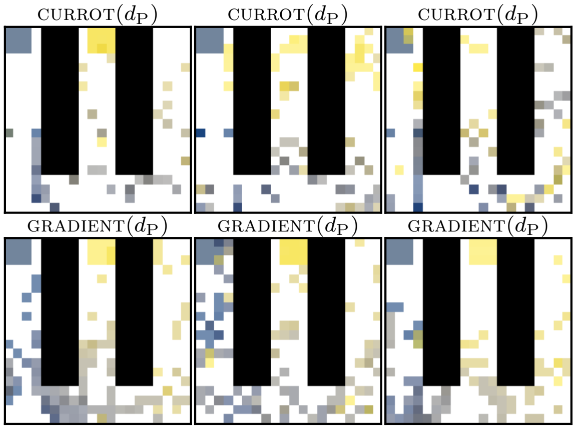


Fig. 17: Interpolations generated by CURROT and GRADIENT in different runs for the current performance-based distance $d_{\text{P}}(c_1, c_2)$. Brighter colors indicate later iterations.

interpolating distribution p

$$\begin{aligned}&\min_{p} \mathcal{W}_2(p, \mu) \\ &\text{s.t. } p(\mathcal{V}(\pi, \delta)) = 1 \quad H(p) \geq H_{\text{LB}}.\end{aligned} \tag{20}$$

The above entropy regularized objectives are not linear programs anymore, and we hence solve the (convex) objectives with the CVXPY library [79].

C.3 Unlock-Pickup Environment

We use the Unlock-Pickup environment from the Minigrid library [67]. We do not change the behavior of the environment and only remove the additional discounting that occurs within the environment, as the environment does not reveal the current timestep to the agent, which, combined with an internally discounted reward, leads to non-Markovian behavior. As stated in the main article, the context c controls the initial state of the environment by specifying the position of the agent, key, and box as well as the position and state of the door (i.e., open or closed). We use the DQN algorithm since the extremely sparse nature of the environment favors RL algorithms with a replay buffer. Compared to the default parameters of the DQN algorithm, we only increase the exploration rate from 0.05 to 0.1 and also increase the batch size to 256. We train the Q-network every fourth step, updating the target network with a Polyak update with $\tau = 0.005$ in each step.

The Q-network is realized by encoding the image observation with a convolutional neural network with three convolutions of kernel size $(2, 2)$, ReLU activations after each convolution, and a max-pool operation with kernel size $(2, 2)$ after the first convolution and ReLU operation. We do not use information about the agent orientation or the textual task description, as both are not strictly necessary for our environment. The convolutional network has 32-dimensional hidden layers. The output of the convolutional encoder is 64-dimensional, which is then further processed by two fully connected layers with 64 dimensions and ReLU activations before being reduced to the Q-values for the seven actions available in the environment.

As briefly mentioned in the main article, the target distribution $\mu(c)$ is a uniform distribution over all those contexts in which the agent is in the left room with a closed door and does not hold the key. The initial state distribution contains one context for each box position in the right room in which the agent is positioned directly next to the box.

Distance Function As discussed in Section 6.2, a context c controls the starting state of the environment, which is defined by

- the agent position $\text{ap} : \mathcal{C} \mapsto [1, 9] \times [1, 4]$
- the key position $\text{kp} : \mathcal{C} \mapsto [1, 9] \times [1, 4]$
- the box position $\text{bp} : \mathcal{C} \mapsto [6, 9] \times [1, 4]$
- the position of the door in the wall $\text{dp} : \mathcal{C} \mapsto [1, 4]$
- the state of the door $\text{ds} : \mathcal{C} \mapsto \{\text{open}, \text{closed}\}$.

The images of the individual functions that access the state information of a context are motivated by the two rooms $R_1 = [1, 4] \times [1, 4]$ and $R_2 = [6, 9] \times [1, 4]$ that make up the environment. Consequently, the agent and the key can be placed in both rooms, whereas the box can only be placed in R_2. The wall that separates the rooms occupies tiles in $W(c) = \{(5, y) \mid y{\in}[1, 4], y{\neq}\text{dp}(c)\}$. Due to this wall, we restrict the context space $\mathcal{C}$ such that it does not contain contexts in which the agent or key is located in the wall, i.e., $\text{ap}(c) \notin W(c)$ and $\text{kp}(c) \notin W(c)$. Additionally, we only allow placing the agent and key in R_2 if the door is open. Formally, this requires $\text{ap}(c){\geq}4 \Rightarrow \text{ds}(c){=}\text{open}$ and $\text{kp}(c){\geq}4 \Rightarrow \text{ds}(c){=}\text{open}$. Finally, neither key nor agent can be at the same position as the box, i.e., $\text{ap}(c) \neq \text{bp}(c)$ and $\text{kp}(c) \neq \text{bp}(c)$. With these restrictions, we arrive at the 81.920 individual contexts mentioned in Section 6.2.

Note that the distance function between contexts reasons both about state changes that can be achieved in an episode, such as moving between agent positions, and ones that can't, such as moving the box. Moving boxes is impossible since the episode terminates successfully when the agent picks up the box. Hence, a distance function that is purely based on state transitions would neglect certain similarities between contexts in this environment.

We define the distance function $d_{\text{base}}(c_1, c_2)$ function via representatives $r(c)$, i.e.

$$d(c_1, c_2){=}\begin{cases} d_{\text{base}}(c_1, r(c_1)){+}d_{\text{base}}(r(c_1), r(c_2)) \\ \quad +d_{\text{base}}(r(c_2), c_2), \text{ if } \text{ds}(c_1){\neq}\text{ds}(c_2) \\ d_{\text{base}}(c_1, c_2), \text{ else.} \end{cases} \tag{21}$$

Such distances are also known as highway distances [68]. The mapping $r : \mathcal{C} \mapsto \mathcal{C}$ from a context c to its representative $r(c)$ ensures that the agent is standing right in front of the open door with the key in its hand, i.e., $\text{ds}(r(c)){=}\text{open}$, and $\text{ap}(r(c)){=}\text{kp}(r(c)){=}[4, \text{dp}(c)]$, while ensuring that $\text{dp}(r(c)){=}\text{dp}(c)$ and $\text{bp}(r(c)){=}\text{bp}(c)$.

The base distance $d_{\text{base}}(c_1, c_2)$ encodes the cost of moving both key and agent from their positions in c_1 to those in c_2 (via d_{ka}) as well as the cost of equalizing the box positions between the contexts (via the L1 distance)

$$d_{\text{base}}(c_1, c_2) = \begin{cases} d_{\text{ka}}(c_1, c_2) + \|\text{bp}(c_1) - \text{bp}(c_2)\|_1, \\ \quad \text{if } \text{dp}(c_1){=}\text{dp}(c_2) \\ \infty, \text{ else.} \end{cases} \tag{22}$$

We see that we render contexts with different door positions incomparable to ease the definition of the distance function. The key-agent distance is defined on top of an object distance $d_{\text{obj,dp}}$ that is conditioned on a door position dp

$$d_{\text{ka}}(c_1, c_2) = \begin{cases} d_{\text{obj,dp}(c1)}(\text{ap}(c_1), \text{ap}(c_2)), \text{ if } \text{kp}(c_1){=}\text{kp}(c_2) \\ d_{\text{obj,dp}(c1)}(\text{ap}(c_1), \text{kp}(c_1)) \\ \quad +d_{\text{obj,dp}(c1)}(\text{kp}(c_1), \text{kp}(c_2)) \\ \quad +d_{\text{obj,dp}(c1)}(\text{ap}(c_2), \text{kp}(c_2)), \text{ else.} \end{cases} \tag{23}$$

Note that we can simply take $\text{dp}(c_1)$ since we know that $\text{dp}(c_1){=}\text{dp}(c_2)$. The object distance is defined as the L1 distance between the two objects if they are in the same room and incorporates the detour caused by passing through the door in the wall if not

$$d_{\text{obj,dp}}(\mathbf{p}_1, \mathbf{p}_2) = \begin{cases} \|\mathbf{p}_1 - \mathbf{p}_2\|_1, \text{ if } p_{1,0}{\leq}4 \Leftrightarrow p_{2,0}{\leq}4 \\ \|\mathbf{p}_1 - [5, \text{dp}]\|_1 + \|[5, \text{dp}] - \mathbf{p}_2\|_1, \text{ else.} \end{cases} \tag{24}$$

We ensured that the resulting distance $d(c_1, c_2)$ fulfills all axioms of a valid distance function, i.e. $d(c_1, c_2) \geq 0$,

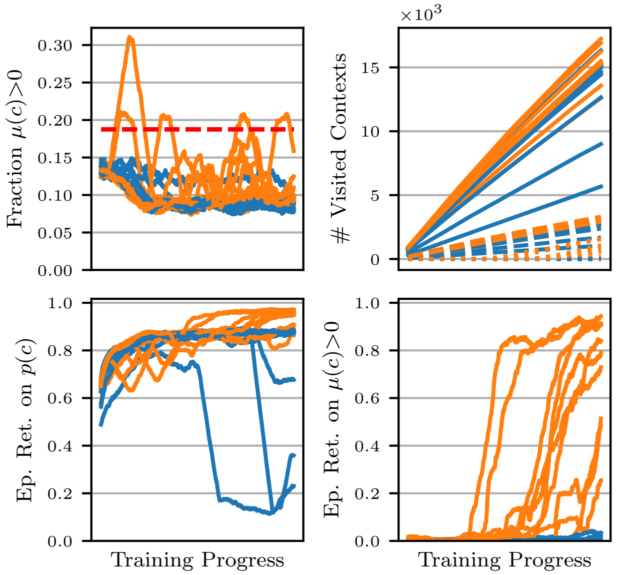


Fig. 18: Statistics of the PLR curricula in the unlock-pickup environment over training progress. The top left plot shows the fraction of contexts sampled by PLR that are also sampled by the target context distribution $\mu(c)$. The red dashed line indicates the fraction of target samples generated by a random curriculum. The top right plot shows the number of unique contexts (solid lines), unique target contexts (dashed lines), and unique solved target contexts (dotted lines) sampled by PLR at least once. The bottom left plot indicates the performance on the PLR curriculum. The performance in those contexts of the curriculum, which are also sampled by the target context distribution $\mu(c)$ (i.e., on the fraction indicated in the top left), are shown in the bottom right.

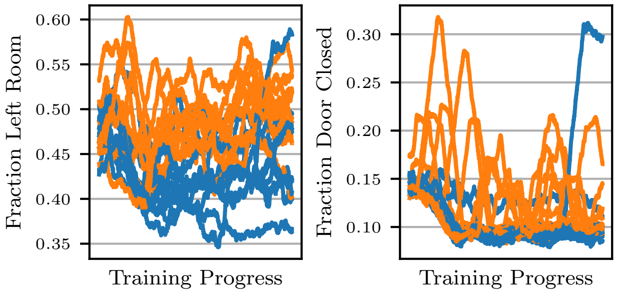


Fig. 19: Fraction of contexts in the PLR curricula in which the agent is placed in the left room (left) and in which the door is closed (right) at the start of the episode. A closed door implies that the agent is located in the left room, hence a more strict condition. Note that the color coding corresponds to the one in Figure 18, indicating runs with high- (orange) and low performance (blue) on $p(c)$.

$d(c_1, c_2){=}0 \Leftrightarrow c_1{=}c_2$, $d(c_1, c_2) = d(c_2, c_1)$, and $d(c_1, c_3) \leq d(c_1, c_2) + d(c_2, c_3)$ via brute-force computations. Note that the in-comparability of contexts with different door positions effectively splits the context space into four disjoint sets (for the four different door positions) that cannot be compared. Hence, we must only ensure these axioms within the four disjoint sets separately.

PLR Performance: As mentioned in Section 6.2, Figure 18 shows statistics of the PLR curricula. We can see that throughout most PLR curricula, the chance of sampling a target context stays relatively constant, even though the number of distinct sampled contexts and the number of distinct sampled target contexts continuously grows. We also see that the agent receives a positive learning signal on $p(c)$ in all runs of PLR. Additionally, we see that the prioritization by PLR suppresses contexts from $\mu(c)$ since a purely random curriculum would sample a target context 18.75% of the time. In about half of the runs (orange lines), the agent learned to solve some of the target tasks, although this fraction is rather low (there are 15.360 target tasks). Interestingly, this increase in proficiency on tasks from $\mu(c)$ does not go hand-in-hand with a consistently increased sampling rate of target tasks. However, as we see in Figure 19 there seems to be a tendency of PLR runs that are more successful on $\mu(c)$ to sample more contexts in which the agent is located in the left room at the beginning of the episode. Generally speaking, Figures 18 and 19 show that PLR prioritized specific contexts over others. However, either due to the missing notion of a target distribution or the dependence of PLR on the agent's internal value function (which may be biased and incorrect), the generated curricula did not consistently progress to the most challenging, long-sequence tasks encoded by $\mu(c)$.

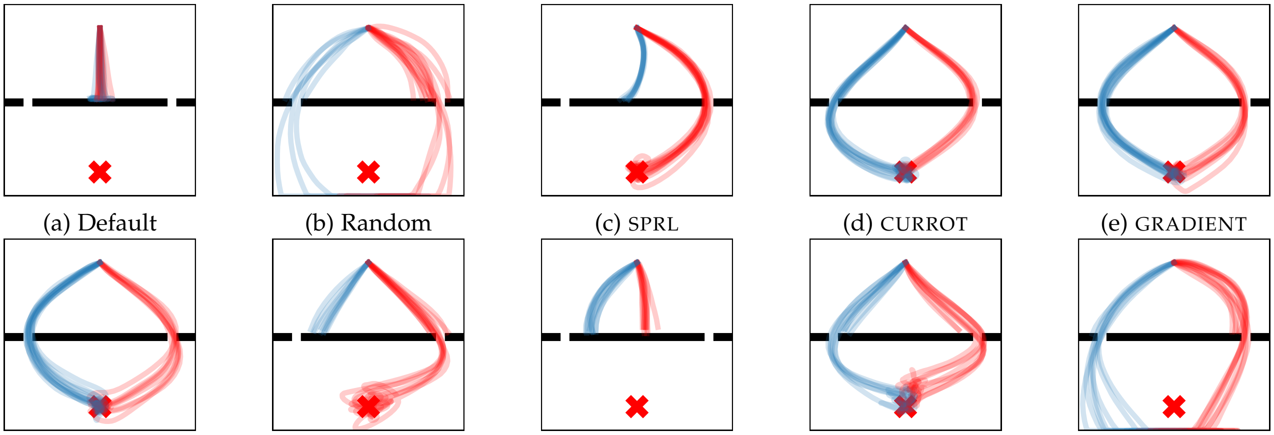
C.4 Point-Mass Environment

The environment setup is the same as the one investigated by Klink et al. [17, 19] with the only difference in the target context distributions, which is now defined as a Gaussian mixture

$$\mu(\mathbf{c}) = \frac{1}{2}\mathcal{N}\left(\mathbf{c}_1, 10^{-4}\mathbf{I}\right) + \frac{1}{2}\mathcal{N}\left(\mathbf{c}_2, 10^{-4}\mathbf{I}\right)$$
$$\mathbf{c}_1 = [-3\ 0.5]^T, \mathbf{c}_2 = [3\ 0.5]^T.$$

In this environment, we use PPO with 4.096 steps per policy update, a batch size of 128, and $\lambda{=}0.99$. All other parameters are left to the implementation defaults of the `Stable Baselines 3` implementation.

Figure 20 shows trajectories generated by agents trained with different curricula in the point-mass environment. We see that directly learning on the two target tasks (Default) prevents the agent from finding the gates in the wall to pass through. Consequently, the agent minimizes the distance to the goal by moving right in front of the wall (but not crashing into it) to accumulate reward over time. We see that random learning indeed generates meaningful behavior. This behavior is, however, not precise enough to pass reliably through the wall. As mentioned in the main article, SPRL only learns to pass through one of the gates, as its uni-modal Gaussian distribution can only encode one of the modes of $\mu(\mathbf{c})$ (see Figure 21 for a visualization). CURROT and GRADIENT learn policies that can pass through both gates reliably, showing that the gradual interpolation towards both target tasks allowed the agent to learn both. ALP-GMM and PLR also learn good policies. The generated trajectories are, however, not as precise as the ones learned with CURROT and GRADIENT and sometimes only solve one of the two tasks reliably. ACL, GOALGAN, and VDS partly



(a) Default (b) Random (c) SPRL (d) CURROT (e) GRADIENT

(f) ALP-GMM (g) GOALGAN (h) ACL (i) PLR (j) VDS

Fig. 20: Final trajectories generated by the different investigated curricula in the point mass environment. The color encodes the context: Blue represents gates positioned at the left and red at the right.

create meaningful behavior. However, this behavior is unreliable, leading to low returns due to the agent frequently crashing into the wall.

C.5 Sparse Goal-Reaching Environment

For the sparse goal-reaching task, the goal can be chosen within $[-9, 9] \times [-9, 9]$, and the allowed tolerance can be chosen from $[0.05, 18]$. Hence, the context space is a three-dimensional cube $\mathcal{C} = [-9, 9] \times [-9, 9] \times [0.05, 18]$. The actually reachable space of positions (and with that goals) is a subset of $[-7, 7] \times [-7, 7]$ due to the “hole” caused by the inner walls of the environment. The target context distribution is a uniform distribution over tasks with a tolerance of 0.05

$$\mu(\mathbf{c}) \propto \begin{cases} 1, & \text{if } c_3 = 0.05, \\ 0, & \text{else.} \end{cases}$$

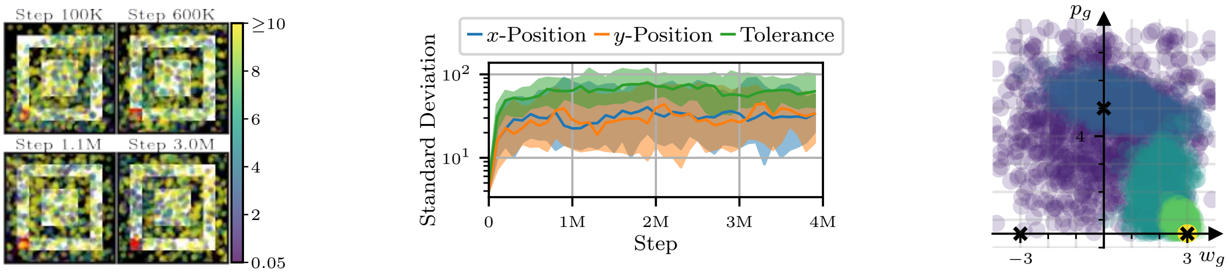
The state s of the environment is given by the agent’s x- and y-position. The reward is sparse, only rewarding the agent if the goal is reached. A goal is considered reached if the Euclidean distance between the goal and position of the point mass falls below the tolerance

$$\|\mathbf{s} - [\mathbf{c}_1 \ \mathbf{c}_2]^T\|_2 \leq c_3.$$

The two-dimensional action of the agent corresponds to its displacement in the $x-$ and $y-$ direction. The action is clipped such that the Euclidean displacement per step is no larger than 0.3.

Given the sparse reward of the task, we again use an RL algorithm that utilizes a replay buffer. Since the actions are continuous in this environment, we use SAC instead of DQN. Compared to the default algorithm parameters of `Stable Baselines 3`, we only changed the policy update frequency to 5 environment steps, increased the batch size to 512, and reduced the buffer size to 200.000 steps.

Figure 21 visualizes the behavior of SPRL in the sparse goal-reaching (SGR). We see that for the SGR environment, SPRL increases the variance of the Gaussian context distribution to assign probability density to the target contexts while fulfilling the expected performance constraint by encoding trivial tasks with high tolerance (Figures 21a and 21b). The inferior performance of an agent trained with SPRL



(a) SPRL Curriculum (SGR) (b) SPRL Sampling Distribution Stds. (SGR) (c) SPRL Curriculum (Point Mass)

Fig. 21: a) Visualization of the sampling distribution of SPRL in the sparse goal-reaching (SGR) task. The color of the dots encodes the tolerance of the corresponding contexts, and the position represents the goal to be reached under that tolerance. The walls are shown in black, and the red area visualizes the starting area of the agent. b) 10-, 50- and 90-percentile of the standard deviation of SPRL’s sampling distribution on the sparse goal-reaching task. The statistics have been computed from 20 seeds. c) Sampling distribution of SPRL in the point mass environment for a given seed. The color indicates the iteration, where brighter colors correspond to later iterations.

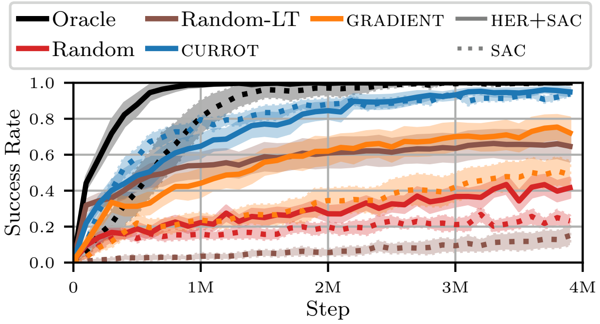


Fig. 22: Comparison of Hindsight Experience Replay (HER, solid lines) and SAC (dotted lines). Across all curricula, pairing HER and SAC achieves similar or better final success rate compared to SAC alone. The final success rate improves the most when training on random tasks with the target tolerance of 0.05 (**Random-LT**). When training on random tasks with randomized tolerance (**Random**), performance improvements are less pronounced. Mean and two-times standard error intervals are computed from 20 seeds.

compared to one trained with a random curriculum shows that the Gaussian approximation to a uniform distribution is a poor choice for this environment. While it may be possible to find other parametric distributions that are better suited to the particular problem, CURROT flexibly adapts the shape of the distribution without requiring any prior choices.

Hindsight Experience Replay (HER): Given the success of HER for sparse-reward goal-reaching tasks, we evaluated its performance in our sparse goal-reaching environment. A difference to the environments evaluated by Andrychowicz et al. [11] is the varying tolerance encoded by the contexts $\mathbf{c} \in \mathbb{C} \subseteq \mathbb{R}^3$. Andrychowicz et al. [11] assumed a fixed tolerance for their investigations of HER. We consequently train HER by uniformly sampling $\mathcal{C}$, corresponding to the Random strategy in Figure 13, and sampling from $\mu(\mathbf{c})$, i.e., only sampling high-precision tasks. We refer to the latter sampling strategy as **Random-LT**, where LT is short for low tolerance. HER only influences experience replay and can be easily combined with arbitrary task sampling strategies. Figure 22 shows the results of training HER with the aforementioned task-sampling strategies and in combination with GRADIENT and CURROT. We used the HER implementation in the `Stable Baselines 3` library [76] with the *future* strategy. We tuned the number of additional goals to maximize HER's performance, finding that $k{=}2$ additional goals for each real goal delivered the best results. Looking at Figure 22, we see that HER is well-compatible with all curricula, either matching or improving upon the success rate of SAC alone. HER drastically improves the performance when directly sampling high-precision tasks of $\mu(\mathbf{c})$. Training on random tasks of $\mathcal{C}$ and with GRADIENT benefit from HER, whereas the performance of CURROT does not improve with the replay of hindsight goals. Finally, when training only on the feasible tasks of $\mu(\mathbf{c})$ (Oracle), HER significantly improves learning speed. The results indicate that for this task, HER's implicit curriculum has a somewhat orthogonal effect than the explicit curricula realized by the different investigated sampling strategies.

C.6 Teach My Agent

As mentioned in the main text, we used the environment and SAC learning agent implementation provided by Romac et al. [21]. We only interfaced CURROT and GRADIENT to the setup they provided, allowing us to reuse the baseline evaluations provided by Romac et al. [21]. The two settings (*mostly infeasible* and *mostly trivial*) differ in the boundaries of their respective context spaces. The *mostly infeasible* setting encodes tasks with a stump height in $[0, 9]$ and -spacing in $[0, 6]$. The *mostly trivial* setting keeps the same boundaries for the stump spacing while encoding stumps with a height in $[-3, 3]$. Since a stump with negative height is considered not present, half of the context space of the *mostly trivial* setting does not encode any obstacles for the bipedal walker to master. The initial- and target context distribution $\mu(\mathbf{c})$ is uniform over the respective context space $\mathcal{C}$ for both settings.

Appendix D
Higher Dimensional Context Experiments

In addition to the low-dimensional context parametrizations of the tasks in the main article, we create a higher-dimensional version of the point-mass environment in which we essentially over-parameterize the environment. We do this by keeping the position of the gate $p_g \in [-4, 4]$ as a parameter but splitting the gate width into a left- and right width parameter $w_{g,l} \in [0.25, 4]$ and $w_{g,r} \in [0.25, 4]$. Note that we multiplied the range of two width parameters by a factor of 0.5 compared to the regular point mass environment from the main article. The actual context for this environment consists of multiple instances of these three parameters, i.e.

$$\mathbf{c} {=} [p_{g_1} \dots p_{g_N} \, w_{g_1,l} \dots w_{g_N,l} \, w_{g_1,r} \dots w_{g_N,r}] \in \mathcal{C} {\subseteq} \mathbb{R}^{3N}.$$

We instantiate the point-mass environment from this over-parameterized context using two different reductions

$$\begin{aligned} \mathbf{c}_{\text{min}} &= \left[p_{g_{n^*}} \ \min_{n \in [1,N]} w_{g_n,l} \ \min_{n \in [1,N]} w_{g_n,r} \right] \\ \mathbf{c}_{\text{max}} &= \left[p_{g_{n^*}} \ \max_{n \in [1,N]} w_{g_n,l} \ \max_{n \in [1,N]} w_{g_n,r} \right] \\ n^* &= \arg\max_{n \in [1,N]} |p_{g_n}| \, . \end{aligned} \tag{25}$$

The only difference between the environment in Section 6.3 and the one investigated in this section is that we separately parameterize the width of the left- and right gate half. We chose these two reductions to highlight that not only the dimensionality of the context space $\mathcal{C}$ is important but also its underlying structure. When using $\mathbf{c}_{\text{min}}$, the chance of sampling tasks with a narrow gate far away from the center increases with N. For $\mathbf{c}_{\text{max}}$, the chance of sampling wide gates increases. Most importantly, the learning task does not get more complex with an increasing value of N since the agent always faces the same learning task and observation space. We can hence be sure that observed performance drops are not due to an inherently more complex learning- or approximation task on the level of the RL agent but

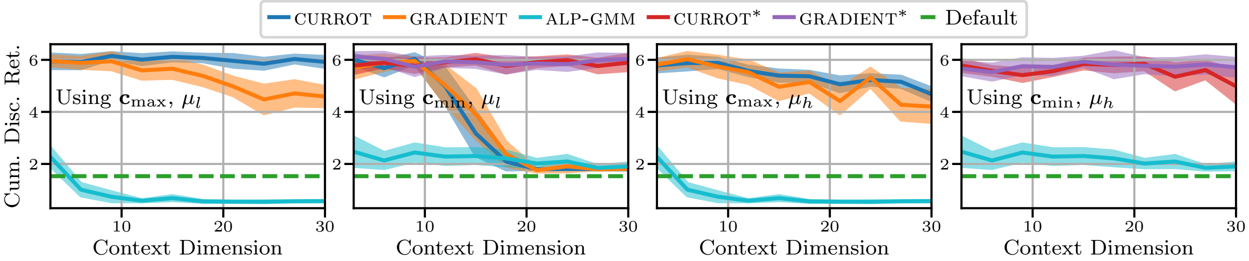


Fig. 23: Performance of CURROT and GRADIENT in high-dimensional context space versions of the point mass environment. The two left plots show the final agent performance when training for the low-entropy target distribution $\mu_l(\mathbf{c})$ (Eq. 26) for different reductions $\mathbf{c}_{\text{max}}$ and $\mathbf{c}_{\text{min}}$. The two right plots shows the same results when training for the high-entropy target distribution $\mu_h(\mathbf{c})$. Note that the performance of ALP-GMM is not affected by a change in target distribution since it generates the curriculum without this information. The green line indicates the average final performance of regular training on $\mu(\mathbf{c})$. Means (thick lines) and two-times standard errors (shaded areas) are computed from 20 seeds. CURROT* and GRADIENT* refer to versions of CURROT and GRADIENT that use the less adversarial initial task distribution for the $\mathbf{c}_{\text{min}}$ reduction (please see Appendix D for a description).

are due to the curriculum generation. We first investigate a narrow Gaussian mixture model as the target distribution

$$\mu_l(\mathbf{c}) = \frac{1}{2}\mathcal{N}\left(\mathbf{c}_1, 10^{-4}\mathbf{I}\right) + \frac{1}{2}\mathcal{N}\left(\mathbf{c}_2, 10^{-4}\mathbf{I}\right) \tag{26}$$

$$\mathbf{c}_1 = [\underbrace{-3\ldots-3}_{N-\text{times}}\ \underbrace{0.25\ldots0.25}_{N-\text{times}}\ \underbrace{0.25\ldots0.25}_{N-\text{times}}] \tag{27}$$

$$\mathbf{c}_2 = [\underbrace{3\ldots3}_{N-\text{times}}\ \underbrace{0.25\ldots0.25}_{N-\text{times}}\ \underbrace{0.25\ldots0.25}_{N-\text{times}}].$$

We benchmarked CURROT, GRADIENT, and ALP-GMM in this task, keeping all algorithm parameters the same as in the point-mass environment from the main article and only adjusting the trust region parameter ϵ of CURROT according to the rule described in Appendix C since the effective distances between points in $\mathcal{C}$ increase with N. Importantly, we always represent the curricula for CURROT and GRADIENT using 100 particles. Figure 23 shows the obtained results. As we see, CURROT and GRADIENT generate good curricula even for high-dimensional context spaces when using the $\mathbf{c}_{\text{max}}$ reduction but fail for higher-dimensional scenarios when using $\mathbf{c}_{\text{min}}$. However, this failure does not arise from a failing interpolation but due to the increasing likeliness of sampling complex tasks under the initial uniform distribution over $\mathcal{C}$, leading to CURROT and GRADIENT not reaching the performance threshold δ on the initial distribution $p_0(\mathbf{c})$. We first tested the feasible context search from the TeachMyAgent benchmark to remedy this issue. However, this search also failed since, just like for uniform noise, uninformed Gaussian noise increases the chance of sampling small gates for $\mathbf{c}_{\text{min}}$ in high dimensions. To benchmark the algorithms for the $\mathbf{c}_{\text{min}}$ reduction, we consequently generate an initial distribution with the same distribution of gate positions and -widths as for $N{=}1$, regardless of the choice of N. We do this by sampling contexts for $N{=}1$ and then projecting them to the required dimension by sampling appropriate random values for the remaining entries in $\mathbf{c}$. Starting from this initial distribution makes the agent proficient on $\mu(\mathbf{c})$ across all dimensions, as shown in Figure 23 (we denote the resulting approaches as CURROT* and GRADIENT*).

We additionally investigate a setting where $\mu_h(\mathbf{c})$ encodes all high-dimensional contexts $\mathbf{c}$ that result in the same reduced target contexts $\mathbf{c}_1{=}[-3\;0.25\;0.25]$ and $\mathbf{c}_2{=}[3\;0.25\;0.25]$. When evaluating the CURROT method in this scenario, we saw that our approximate optimization of Objective (14) via uniform samples in a half-sphere did not lead to good progression to the target samples. Adding samples along the direction $\mathbf{c}_{\mu,\phi(n)}{-}\mathbf{c}_{p,n}$ was enough to solve this issue in our approximate optimization and ensure good progression. While performing these experiments, we saw that our rule of choosing the ϵ parameter for CURROT, i.e., setting it to 0.05 of the maximum distance d_{max} between any two contexts in $\mathcal{C}$, can prevent Objective (14) from sampling high-dimensional high-entropy distributions. This problem occurs if the Wasserstein distance between two particle-based representations $\hat{\mu}_1(\mathbf{c})$ and $\hat{\mu}_2(\mathbf{c})$ of the target distribution $\mu(\mathbf{c})$ is larger than $\epsilon{=}0.05d_{\text{max}}$. Consequently, we adapted our rule of choosing the trust region size for CURROT to $\epsilon{=}\max(0.05d_{\text{max}}, 1.2\mathcal{W}_2(\hat{\mu}_1, \hat{\mu}_2))$. Figure 23 shows the results for the high-entropy target distributions, where we again see that both CURROT and GRADIENT can solve these tasks.

Choosing Particles in Higher Dimensions: The findings in this section provided a better understanding of the role of the number of particles to represent $\hat{p}(\mathbf{c})$ that we would like to summarize here.

For CURROT and GRADIENT, the particles serve two objectives: Approximating the sampling- and target distribution and estimating the agent performance. By restricting the curriculum to the barycentric interpolation, GRADIENT can provide unbiased samples from the interpolation and the target distribution even when using a few particles in high dimensions. Consequently, the need for more particles in higher dimensions only arises from counteracting a potentially higher variance of the expected performance estimate. However, the effect of more noisy expected performance estimates can also be counteracted by smaller step sizes ϵ with which to advance the Barycentric interpolation.

For CURROT, we saw that a small number of particles in combination with a too-small trust region ϵ can lead to biased sampling of the target distribution. However, we also saw that setting $\epsilon{=}\max(0.05d_{\text{max}}, 1.2\mathcal{W}_2(\hat{\mu}_1, \hat{\mu}_2))$ for a given number of particles N ensures good sampling of the target distribution. With this automated choice of ϵ, the appropriate number of N should, as for the GRADIENT algorithm, be guided by the complexity of the performance estimate. If the performance estimate is not of sufficient quality, increasing the number of particles will decrease the minimum required trust region to sample unbiasedly from $\mu(\mathbf{c})$ and yield more samples for the kernel regression. It is also possible to only increase the number of particles in the buffers for the kernel regression while keeping the number of particles representing the context distribution fixed.

Finally, a more specific feature of CURROT is the optimization of Objective (14), which may become more delicate in higher dimensions and require more sophisticated approaches than the simple sampling scheme used in this article. One option could be to use parallelized gradient-based optimization schemes, which should be easy to implement given the rather simple nature of the constraints.

Initial Distribution in Higher Dimensions: We also saw that using a uniform initial distribution $p_0(\mathbf{c})$ for GRADIENT and CURROT can be problematic if easy tasks are unlikely under this distribution. In this case, CURROT and GRADIENT will not achieve the expected performance threshold to progress the curriculum. Furthermore, simple search approaches for feasible contexts like the one detailed in Appendix B may fail. At this point, it may either be required to implement a more problem-specific search for feasible contexts or provide a more informed initial distribution $p_0(\mathbf{c})$ that does not require a search for feasible contexts. Both of these approaches can be used for GRADIENT and CURROT.