Aalto University

OPEN ACCESS

This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Klink, Pascal; Yang, Haoyi; D'Eramo, Carlo; Pajarinen, Joni; Peters, Jan

Curriculum reinforcement learning via constrained optimal transport

Published in:
Proceedings of the 39th International Conference on Machine Learning

Published: 01/01/2022

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Klink, P., Yang, H., D'Eramo, C., Pajarinen, J., & Peters, J. (2022). Curriculum reinforcement learning via constrained optimal transport. In *Proceedings of the 39th International Conference on Machine Learning* (Proceedings of Machine Learning Research; Vol. 162). JMLR.
https://proceedings.mlr.press/v162/klink22a/klink22a.pdf

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Curriculum Reinforcement Learning via Constrained Optimal Transport

Pascal Klink [1] **Haoyi Yang** [1] **Carlo D'Eramo** [1] **Joni Pajarinen** [1 2] **Jan Peters** [1]

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 a variety of works, it is less clear how to generate them for a given learning environment, resulting in a variety of methods aiming to automate this task. In this work, we focus on the idea of 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 a variety of tasks with different characteristics.

1. Introduction

Reinforcement learning (RL) (Sutton & Barto, 1998) has celebrated great successes as a framework for autonomous acquisition of desired behavior. With ever-increasing computational power, this framework and the algorithms developed under it have allowed to create learning agents capable of solving non-trivial long-horizon planning (Mnih et al., 2015; Silver et al., 2017) and control tasks (Akkaya et al., 2019). However, these successes have highlighted the need for certain forms of regularization, such as leagues in the context of board games (Silver et al., 2017), a gradual diversification of simulated training environments for robotic manipulation (Akkaya et al., 2019) or a tailored training pipeline in the context of humanoid control for soccer (Liu et al., 2021). These regularizations can overcome shortcomings of modern RL agents, like poor exploratory behavior – an active topic of research (Bellemare et al., 2016; Ghavamzadeh et al., 2015; Machado et al., 2020).

One can view the aforementioned regularizations under the umbrella term of curriculum reinforcement learning (Narvekar et al., 2020), 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 a variety of ways and they are motivated from many perspectives in the literature (Andrychowicz et al., 2017; Florensa et al., 2017; Portelas et al., 2019; Wöhlke et al., 2020).

A perspective of particular interest for this paper 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 (Klink et al., 2021). While algorithmic realizations of this perspective have been successfully evaluated in the literature (Klink et al., 2020a;b; Chen et al., 2021a), there also exist evaluations attesting those methods a rather poor learning performance (Romac et al., 2021). This discrepancy calls for a better understanding of why these methods perform well in some scenarios - and why they do not in others.

In this paper, we investigate the shortcomings of methods that realize curricula as an interpolation between task distributions using the KL divergence as a measure of distributional similarity. Based on the resulting insights, we propose to realize curricula as an optimal transport problem in a parameterized task space subject to a performance constraint on the tasks probable under the curriculum. As we discuss in the paper, this high-level concept implicitly introduces empirically successful concepts observed in existing curriculum RL algorithms. Nevertheless, we also show how it differs from these existing algorithms.

In experiments, we show that the proposed method matches or surpasses the performance of existing baselines on a variety of tasks. To summarize, this paper

1) highlights shortcomings of current curriculum RL algorithms that realize curricula as interpolations between task distributions;
2) proposes to generate curricula by solving a constrained optimal transport problem;
3) discusses and evaluates the resulting algorithm in a variety of experiments.

[1] Intelligent Autonomous Systems, Technical University of Darmstadt, Germany [2] Department of Electrical Engineering and Automation, Aalto University, Finland. Correspondence to: Pascal Klink <pascal.klink@tu-darmstadt.de>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

2. Related Work

The focus of this work is on the generation of training curricula for reinforcement learning (RL) agents. Opposed to supervised learning, where there is an ongoing discussion about the mechanics and effects of curricula in different learning situations (Weinshall & Amir, 2020; Wu et al., 2021), 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, and with that e.g. bypass poor exploratory behavior of non-proficient agents. Applications are by now widespread and different terms have been established. Adaptive Domain Randomization (Akkaya et al., 2019) uses curricula to gradually diversify the training parameters of a simulator to facilitate sim-to-real transfer. Unsupervised environment discovery (Dennis et al., 2020; Jiang et al., 2021b;a) aims to efficiently train an agent which is robust to variations in the environment. Automatic curriculum learning methods (Florensa et al., 2017; Sukhbaatar et al., 2018; Florensa et al., 2018; Portelas et al., 2019; Zhang et al., 2020; Racaniere et al., 2020; Eimer et al., 2021; Klink et al., 2021) particularly focus on improving the learning speed and/or performance of an agent on a set of desired tasks. Curricula are often generated as distributions that maximize a certain surrogate objective, such as learning progress (Baranes & Oudeyer, 2010; Portelas et al., 2019), intermediate task difficulty (Florensa et al., 2018), regret (Jiang et al., 2021b), or disagreement between Q-functions (Zhang et al., 2020). Curriculum generation can also be interpreted as a two-player game (Sukhbaatar et al., 2018). The work by Jiang et al. (2021a) even hints at a link between surrogate objectives and two-player games.
Opposed to these interpretations, other algorithms formulate the generation of a curriculum as an explicit interpolation between an auxiliary task distribution and a distribution of target tasks (Klink et al., 2020a; 2021; Chen et al., 2021a). As shown by Klink et al. (2021), such interpolations can be formally linked to successful curricula in supervised learning (Kumar et al., 2010), the concept of annealing in statistics (Neal, 2001), and homotopic continuation methods in optimization (Allgower & Georg, 2003). In this paper, we reveal shortcomings of such interpolation-based curriculum RL methods caused by the KL divergence as a similarity measure between distributions over learning tasks. Based on these insights, we propose a novel formulation of interpolation-based curricula as a constrained *optimal transport* problem.

Optimal Transport: Dating back to the work by Monge in the 18th century, *optimal transport* has been understood as an important fundamental concept touching upon many fields in both theory and application (Peyré et al., 2019; Chen et al., 2021b). In probability theory, optimal transport translates to the so-called Wasserstein metric (Kantorovich, 1942) that compares two distributions under a given metric, allowing e.g. for the analysis of probabilistic inference algorithms as approximate gradient flows (Liu et al., 2019) and providing well-defined ways of comparing feature distributions or even graphs in computer vision and machine learning (Kolouri et al., 2017; Kandasamy et al., 2018; Togninalli et al., 2019). Gromov-Wasserstein distances (Mémoli, 2011; Vincent-Cuaz et al., 2022) even allow to compare distributions across metric spaces, which has been of use e.g. in computational biology (Demetci et al., 2020) or imitation learning (Fickinger et al., 2022). In some sense, this paper can be seen as introducing curriculum reinforcement learning as another application domain to which the powerful concept of optimal transport can be applied. An important issue of applied optimal transport is its computational complexity. In Appendix A, we discuss computational aspects of optimal transport in more detail.

3. Preliminaries

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

3.1. Contextual Reinforcement Learning

Contextual reinforcement learning (Hallak et al., 2015) can be seen as a conceptual extension to the (single task) reinforcement learning (RL) problem

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

which aims to maximize the above expected discounted reward objective by finding an optimal policy $\pi : \mathcal{S}\times\mathcal{A} \mapsto \mathbb{R}$ 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 $\mathcal{M}(\mathbf{c})$ that the agent is expected to encounter. Objective $J(\pi, \mathbf{c})$ in Eq. (2) corresponds to the objective $J(\pi)$ in Eq. (1) where, however, the initial state distribution p_0, the transition dynamics p as well as the reward function r of

$\mathcal{M}$ are 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 (Wilson et al., 2007), goal-conditioned (Schaul et al., 2015) or curriculum RL (Narvekar et al., 2020).

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 in which initially random agent behavior is unlikely to observe any meaningful learning signals, as e.g. is the case 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 (Portelas et al., 2019), regret (Jiang et al., 2021b) or tasks of intermediate success probability (Florensa et al., 2018). In this paper, we focus on CRL methods which 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 (Klink et al., 2020a;b; 2021) defines p_i as the distribution with minimum KL divergence to μ that fulfills a constraint on the expected agent performance

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

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 $q(\mathbf{c})$. 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 $J(\pi, p)$ caused by a limited amount of samples. Objective (3) can be shown to perform 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 (1781). As of today, generalizations established by Kantorovich (1942) have led to so called **Wasserstein distances** as metrics between probability distributions defined on a metric space $M = (d, \mathcal{C})$ with metric $d: \mathcal{C}{\times}\mathcal{C} \mapsto \mathbb{R}_{\geq 0}$

$$\mathcal{W}_p(p_1, p_2){=}\left(\inf_{\phi \in \Phi(p_1,p_2)} \mathbb{E}_\phi\left[d(\mathbf{c}_1, \mathbf{c}_2)^p\right]\right)^{1/p}, \quad p \geq 1$$
$$\Phi(p_1, p_2){=}\left\{\phi : \mathcal{C}{\times}\mathcal{C}{\mapsto}\mathbb{R}_{\geq 0} | p_i{=}(\text{proj}_i)_\# \phi,\ i{\in}\{1, 2\}\right\},$$

where $(\text{proj}_1)_\#\phi(\mathbf{c}_1){=}\int_\mathcal{C} \phi(\mathbf{c}_1, \mathbf{c}_2)\,\mathrm{d}\mathbf{c}_2$ and $(\text{proj}_2)_\#\phi$ is defined analogously. The distance between p_1 and p_2 results from solving an optimization problem that finds a so-called plan ϕ. This plan encodes how to equalize p_1 and p_2 taking into account the cost of moving density between between parts of the space $\mathcal{C}$. This cost is encoded by the metric d. In the following, we will always assume to work with 2-Wasserstein distances under Euclidean metric, i.e. $p = 2$ and $d(\mathbf{c}_1, \mathbf{c}_2) = \|\mathbf{c}_1 - \mathbf{c}_2\|_2$.

4. Strengths and Pitfalls of Interpolation-Based CRL

This section serves to give a better understanding of both the benefits of the interpolation-based CRL approach as well as its problems. The insights of this section underline why such a take on CRL is important and further motivates the algorithm presented in the next section.

4.1. Convergence to $\mu(\mathbf{c})$

A major motivation for explicitly expressing the context distribution $p_i(\mathbf{c})$ as the result of a constrained (KL) divergence minimization to a target distribution $\mu(\mathbf{c})$ is that it ensures that the training ultimately focuses on the tasks likely under $\mu(\mathbf{c})$. As we discuss now, this is not guaranteed for other existing CRL approaches at least when assuming a *strong learner*.

With a *strong learner*, we refer to one that is neither subject to catastrophic forgetting nor negative interference. Hence after training on a context $\mathbf{c}_{\text{train}}$ and updating the policy π, the new policy π' is guaranteed to improve performance in at least an ϵ-ball $B_\epsilon(\mathbf{c}_{\text{train}})$ around $\mathbf{c}_{\text{train}}$

$$\forall \mathbf{c} \in B_\epsilon(\mathbf{c}_{\text{train}}) : J(\pi, \mathbf{c}) \geq J(\pi', \mathbf{c}) + \Delta,\ \Delta > 0,$$

while keeping at least the same performance on all other contexts in $\mathcal{C}$. Intuitively, this means that when training often enough on a context $\mathbf{c}$, the learner will ultimately converge to the maximally achievable reward $J^*(\mathbf{c}){=}\max_\pi J(\pi, \mathbf{c})$. Nonetheless, the learner may still improve its performance more rapidly in certain parts of

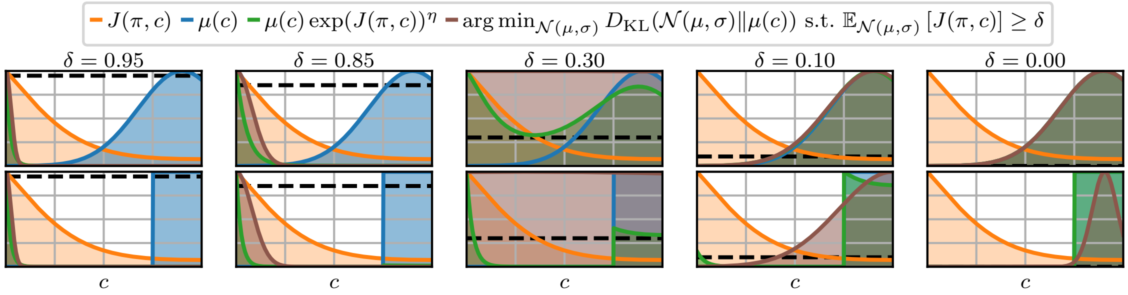


Figure 1. Interpolations (green) generated by optimizing Objective (5). Each row visualizes the interpolation to a different target density $\mu(c)$ (blue) for varying expected performance thresholds δ (horizontal dotted line). The brown density is the result of optimizing Objective (5) while restricting the distribution to a Gaussian. Without a parametric restriction, the green density assigns probability density to distant parts of the context space (see $\delta = 0.3$ in the bottom row).

the context space than others, so the need for a curriculum is not ruled out by these assumptions.

Assuming a strong learner, one can argue that algorithms that define the density $p_i(\mathbf{c})$ as a function that monotonically increases with concepts like regret (Jiang et al., 2021b;a) or learning progress (Portelas et al., 2019; Baranes & Oudeyer, 2010) ultimately converge to a uniform distribution over $\mathcal{C}$. Given that this argument needs certain abstractions from the individual algorithms that would unnecessarily lengthen this exposition, we refer the interested reader to Appendix B. The main point is that without a notion of $\mu(\mathbf{c})$, many existing CRL algorithms aim to make the learner proficient on all of $\mathcal{C}$. While this is reasonable behavior in uninformed settings, this insight hints towards an advantage of interpolation-based CRL methods in scenarios that impose a target distribution $\mu(\mathbf{c})$ different from the uniform distribution on the context space $\mathcal{U}(\mathcal{C})$. The experiments in later sections will empirically evince this supposition.

4.2. Interpolation

The benefits of explicitly encoding a target distribution $\mu(\mathbf{c})$, however, come at a cost: The need to compute $D_{\text{KL}}(p(\mathbf{c}) \parallel \mu(\mathbf{c}))$. Consequently, $\mu(\mathbf{c})$ has either been assumed uniform over $\mathcal{C}$ to ease computation and optimization of a weighted KL divergence objective (Chen et al., 2021a), or been restricted to a Gaussian distribution (Klink et al., 2020a;b; 2021). While empirically successful, these design choices masquerade the pitfalls of the KL divergence to measure distribution similarity in a CRL setting, particularly when dealing with a target distribution that does not assign uniform density over all of $\mathcal{C}$.

Revisiting Eq. (4), we see that the SPRL algorithm by Klink et al. (2021), that we take as an example for this discussion, performs two nested interpolations, which can be obtained as the solution to two individual sub-problem

$$\mu(\mathbf{c})\exp\left(J(\pi, c)\right)^{\eta(\delta)} = \underset{p \in \{q | \mathbb{E}_q[J(\pi,c)] \geq \delta\}}{\arg\min} D_{\text{KL}}\left(p \parallel \mu\right) \tag{5}$$

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

Note that we introduced the notation $\eta(\delta)$ and $\alpha(\epsilon)$ to highlight that the above equalities hold for appropriately chosen values of η and α – which are determined by the values of δ and ϵ. Figures 1 and 2 illustrate these two types of interpolations. We see that the interpolation $\mu(\mathbf{c})\exp(J(\pi, \mathbf{c}))^\eta$ assigns probability density to contexts with high performance in order to assign probability density to contexts that are assigned high probability density by the target distribution $\mu(\mathbf{c})$. However, this trade-off does not necessarily result in focusing on contexts of intermediate agent perfor-

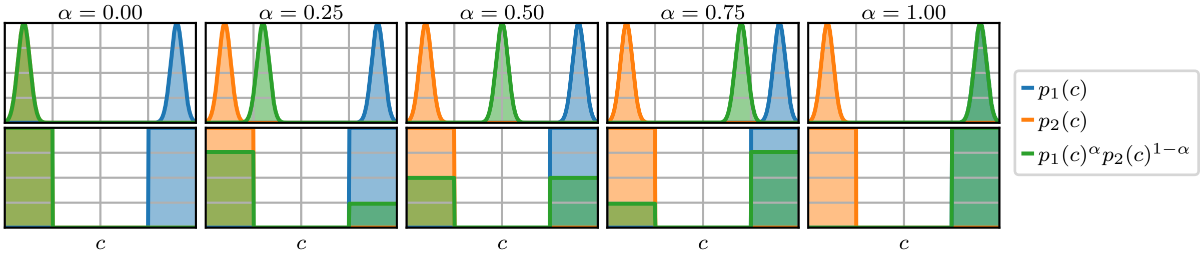


Figure 2. Interpolations (green) generated by optimizing Objective (6) for different values of ϵ (and with that α). In the top row, $p_1(c)$ and $p_2(c)$ are Gaussian while in the bottom row, they assign uniform density over different parts of $\mathcal{C}$.

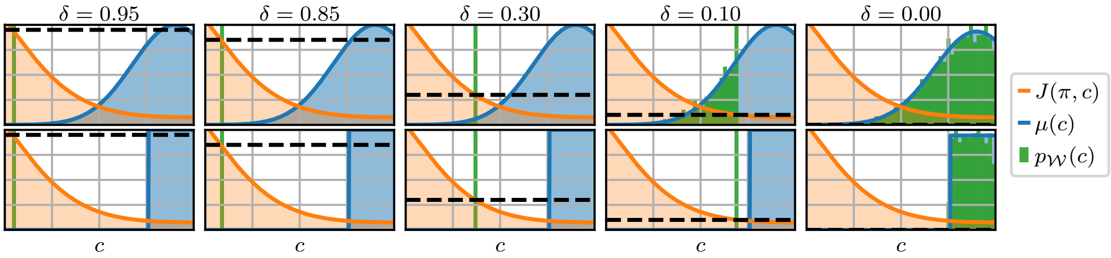


Figure 3. Interpolations (green) based on optimizing Objective (7). Each row visualizes the interpolation to a different target density $\mu(c)$ (blue) for varying performance thresholds δ (horizontal dotted line). The interpolation is approximately computed using the particle-based approach described in Section 5 and visualized as a histogram. The interpolation encodes tasks that satisfy the performance threshold δ and are close to the tasks likely under $\mu(c)$ in a metric sense.

mance and is furthermore highly dependent on the particular shape of $\mu(\mathbf{c})$. For both the Gaussian and uniform target distribution in Figure 1, the agent is putting less probability density on tasks of intermediate agent performance than on tasks of highest or lowest performance in the case of δ=0.3. Particularly for the uniform target distribution in the bottom row of Figure 1, the agent never assigns significant probability density to tasks in which the agent achieves an intermediate level of performance.

The second type of interpolation in Figure 2 points to a similar problem of the KL divergence in a CRL setting. 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. Looking at the interpolation between two uniform distributions in the bottom row of Figure 2, we see that density is displaced from contexts $\mathbf{c}$ to contexts $\mathbf{c}'$ with large Euclidean distance $\|\mathbf{c} - \mathbf{c}'\|_2$. Such a behavior of the context distribution can be problematic in CRL, as the extrapolation of successful behavior in a task $\mathcal{M}(\mathbf{c})$ to a similar task $\mathcal{M}(\mathbf{c}')$ typically assumes that $\|\mathbf{c} - \mathbf{c}'\|_2$ is small. For example when representing the policy π with a deep neural network $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{c})$, the behavior for a fixed state $\mathbf{s}$ typically changes gradually as the context $\mathbf{c}$ changes.

Algorithm 1 Curricula via **O**ptimal **T**ransport (CURROT)

```
Input: Initial context dist. $\hat{p}_{\mathcal{W},0}(\mathbf{c})$, target context dist. $\mu(\mathbf{c})$, performance threshold $\delta$, distance bound $\epsilon$
for $k = 0$ to $K$ do
    Agent Improvement:
    Sample contexts $\mathbf{c}_j \sim \hat{p}_{\mathcal{W},k}(\mathbf{c}),\ j \in [1, M]$
    Train policy $\pi$ under $\mathbf{c}_j$ and observe episodic rewards $R_j = \sum_{t=1}^{\infty} r_{\mathbf{c}_j}(\mathbf{s}_t, \mathbf{a}_t),\ j \in [1, M]$
    Context Distribution Update:
    Update buffers $\mathcal{B}_+$ and $\mathcal{B}_-$ with $\{(\mathbf{c}_j, R_j) | j \in [1, M]\}$
    Estimate $J(\pi, \mathbf{c})$ from $\mathcal{B}_+$ and $\mathcal{B}_-$
    Optimize (7) using $J(\pi, \mathbf{c})$ w.r.t. the $N$ particles $\mathbf{c}_{p_{\mathcal{W}},i}$ to obtain $\hat{p}_{\mathcal{W},k+1}$
end for
```

5. Generating Curricula via Optimal Transport

The previous section highlighted two problems with the interpolation generated by CRL algorithms that minimize KL divergence to a target distribution of tasks: The KL divergence as a notion of similarity between distributions and the expected performance constraint on the interpolating task distribution. These two design choices can result in the generated curriculum neglecting tasks of intermediate agent performance and rather focusing on a mixture of trivial and infeasible tasks whose contextual representations are not close in a metric sense.

Consequently, we investigate the following alternative objective to generate an interpolating task distribution in a CRL setting

$$p_{\mathcal{W}}(\mathbf{c}) = \arg\min_{p} \mathcal{W}_2(p(\mathbf{c}), \mu(\mathbf{c})) \tag{7}$$
$$\text{s.t. } p(\mathbf{c}) > 0 \Rightarrow J(\pi, \mathbf{c}) \geq \delta \quad \forall \mathbf{c} \in \mathcal{C}$$
$$\mathcal{W}_2(p(\mathbf{c}), q(\mathbf{c})) \leq \epsilon.$$

The proposed objective replaces the KL divergence by Wasserstein distances to take the metric structure of $\mathcal{C}$ into account. Further, it replaces the expected performance constraint with a constraint that enforces the desired level of performance in any context that may occur under the curriculum distribution $p_{\mathcal{W}}(\mathbf{c})$. Figure 3 visualizes the behavior of this interpolation for the same setting as investigated in Figure 1. We see that the curriculum now puts all probability density on the border of the desired agent capability (i.e. the performance threshold 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. Indeed, this behavior is similar to those of CRL methods based on task-prioritization. However, it retains the benefit of explicitly encoding a target task distribution $\mu(\mathbf{c})$. If we for example assume a binary reward task and a performance threshold of δ=0.5, the distribution generated by Objective (7) is conceptually similar to the idea of GOALGAN (Florensa et al., 2018), which prioritizes tasks with an intermediate success rate (i.e. around 50%). This small example suggests a link between interpolation-based approaches to CRL and methods that prioritize learning tasks based on surrogate objectives such as success rate (Florensa et al., 2017), regret (Jiang et al., 2021b), or learning progress (Portelas et al., 2019) to target learning tasks at the boundary of agent capability. However, we consider these investigations future work.

For the remainder of this paper, we focus on an empirical evaluation of Objective (7). For this evaluation, we represent the curriculum distribution $p_{\mathcal{W}}(\mathbf{c})$ by a set of N particles

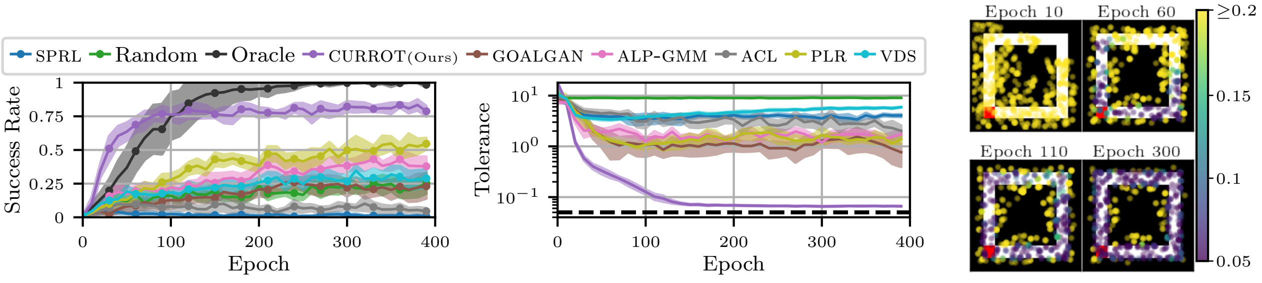
$$p_{\mathcal{W}}(\mathbf{c}) \approx \hat{p}_{\mathcal{W}}(\mathbf{c}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\mathbf{c}_{p_{\mathcal{W}},i}}(\mathbf{c}),$$

where $\delta_{\mathbf{c}_{p_{\mathcal{W}},i}}(\mathbf{c})$ is a Dirac delta at $\mathbf{c}_{p_{\mathcal{W}},i}$. Apart from the N particles, we maintain two context buffers $\mathcal{B}_+$ and $\mathcal{B}_-$ of size N. These two buffers get updated with the results of policy rollouts $(\mathbf{c}, R_{\mathbf{c}})$ during agent training, where $R_{\mathbf{c}} = \sum_{t=0}^{\infty}\gamma^t r_{\mathbf{c}}(\mathbf{s}_t, \mathbf{a}_t)$ is the discounted cumulative return obtained by the agent in context $\mathbf{c}$. While $\mathcal{B}_-$ is simply a circular buffer that keeps the most recent N rollouts with $R_{\mathbf{c}}$ below the performance threshold δ, $\mathcal{B}_+$ contains contexts $\mathbf{c}$ for which $R_{\mathbf{c}} \geq \delta$. However, $\mathcal{B}_+$ is updated differently if full. Once full, we treat $\mathcal{B}_+$ as the particle-based representation of a distribution $p_+(\mathbf{c})$ and replace rollouts in $\mathcal{B}_+$ with new ones such that $\mathcal{W}_2(p_+, \mu)$ is minimized. This can be achieved by sampling N contexts from $\mu(\mathbf{c})$ and solving an assignment problem (more details in Appendix C). After updating the context buffers, we update the particles of $\hat{p}_{\mathcal{W}}$ by optimizing objective (7). Details on this optimization can again be found in Appendix C. We use the data in $\mathcal{B}_+$ and $\mathcal{B}_-$, to approximate $J(\pi, \mathbf{c})$ via a Gaussian Process (Williams & Rasmussen, 2006). A final detail of the algorithm that we evaluate in the next section is its behavior, if the agent performance on $\hat{p}_{\mathcal{W},0}$ is below the performance threshold δ. In this case, we use a simple randomized search method to find areas of $\mathcal{C}$ in which the agent achieves returns above δ. This search procedure is again detailed in Appendix C. While more elaborate methods for finding contexts $\mathbf{c}$ with $J(\pi, \mathbf{c}) \geq \delta$ could be used, we found the employed simple method to be sufficient for our purposes. Algorithm 1 summarizes the outlined CRL algorithm (assuming that the initial agent performance on $\hat{p}_{\mathcal{W},0}$ is sufficient).

6. Experiments

The experiments in this section serve to validate the identified benefits of the proposed interpolation-based CRL method, which we will refer to as CURROT. We proceed by showing that the method can generate curricula for different target distributions $\mu(\mathbf{c})$ while avoiding problems arising from parametric restrictions on the context distribution that e.g. SPRL imposes. We also show that even in scenarios with target distributions uniformly covering $\mathcal{C}$, the proposed method significantly improves over previous evaluations of interpolation-based CRL methods, matching and surpassing the performance of best performing methods so far. As baselines, we will evaluate ACL, GOALGAN, ALP-GMM, VDS, PLR, and SPRL (Graves et al., 2017; Florensa et al., 2018; Portelas et al., 2019; Zhang et al., 2020; Jiang et al., 2021b; Klink et al., 2021) [1].

[1] Additional experimental details are provided in Appendix E. Code is provided under: https://github.com/psclklnk/currot.



(a) SGR Learning Performance and Task Tolerance

(b) SGR Curriculum

Figure 4. a) Left: Success rate on the feasible subspace of $\mathcal{C}$ over learning epochs for different CRL methods in the sparse goal reaching (SGR) task. We also include a uniform sampling baseline (referred to as Random) as well as an oracle baseline which only samples the feasible tasks in the context space $\mathcal{C}$. Right: Median tolerance of tasks generated by different CRL methods as well as the uniform and oracle baseline. For both plots, mean and standard error are computed from 10 runs. b) Context distributions $\hat{p}_{\mathcal{W}}(\mathbf{c})$ for a run of CURROT on the sparse goal reaching task for epochs 10, 60, 110 and 300. The area in which the agent starts each episode is highlighted in red. Walls are shown in black. The position of the samples encodes the goal to be reached while the color encodes the tolerance with which the goal needs to be reached.

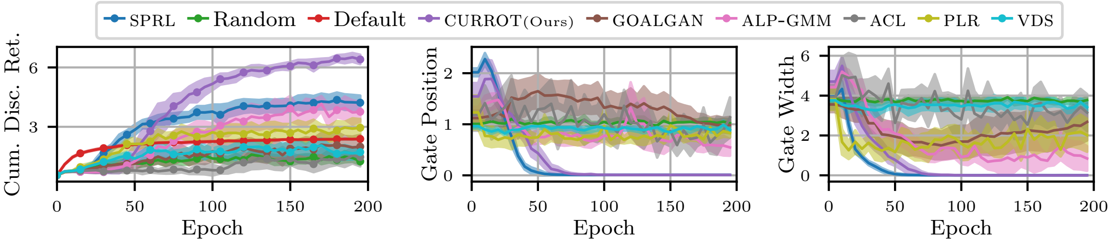


Figure 5. Left: Discounted cumulative return over learning epochs obtained in the point mass environment under different curricula as well as baselines that sample tasks uniformly from $\mathcal{C}$ (Random) or $\mu(\mathbf{c})$ (Default). Middle and Right: Median distance to the target contexts of $\mu(\mathbf{c})$ for the two dimensions of the context space (i.e. gate position and -width). Statistics of the visualizations (mean and standard error) are computed from 10 seeds.

6.1. Sparse Goal Reaching (SGR)

We first turn to a sparse-reward, goal-reaching environment in which an agent needs to reach a desired position with a high precision (Figure 4b). Such environments have e.g. been investigated by Florensa et al. (2018). The contexts $\mathbf{c} \in \mathcal{C} \subseteq \mathbb{R}^3$ of this environment encode the 2D goal position as well as the allowed tolerance for reaching the goal. Given that ultimately, the agent is 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 $\mathcal{C}$ in which the tolerance of each context is minimal. The walls that are present in the environment (Figure 4b) render many tasks encoded by $\mu(\mathbf{c})$ infeasible and hence the curriculum needs to identify the feasible subspace of tasks to achieve a good learning performance. Figure 4a compares the performance of the different CRL algorithms. We see 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 precision. However, even compared to the oracle, we see an increased learning speed of CURROT at the beginning of training. Looking at the right plot of Figure 4a, we see that CURROT continuously decreases the precision with which the goals need to be reached. We suspect that the final gap between tolerance in tasks generated by CURROT and $\mu(\mathbf{c})$ is the cause for the lower final performance of CURROT compared to the oracle. The baseline CRL methods sample tasks with comparatively high tolerance even towards the end of training, which may explain the lower performance on $\mu(\mathbf{c})$. This behavior is to be expected given our discussion in Section 4.1 that algorithms like ALP-GMM, PLR, or GOALGAN ultimately target a uniform distribution over $\mathcal{C}$. Interestingly, SPRL does not progress to high precision tasks but continues to sample tasks of high tolerance in later training epochs. As we show in Appendix D, this behavior is caused by the Gaussian context distribution of SPRL converging to a quasi-uniform distribution over $\mathcal{C}$ as it is otherwise not able to cover the non-Gaussian target distribution of feasible high-precision tasks without encoding a lot of infeasible tasks. Figure 4b shows the evolution of particles for a run of CURROT. We see that the initially high tolerance for reaching the goal is gradually decreased over epochs, starting from goals close to the initial agent position and then spreading to goals that are further away.

6.2. Point Mass

We now consider the point mass environment investigated by (Klink et al., 2020a;b; 2021). As shown in Figure 6, a point mass needs to be steered through a narrow gate to reach a goal position on the other side of a wall. While Klink et al. only considered a narrow gate at one specific position as the target task, we investigate a version in which a narrow gate is located at one of two opposing positions. This is modeled by a bi-modal target context distribution $\mu(\mathbf{c})$ that encodes the contexts $\mathbf{c}_1{=}[-3\;0.5]$ and $\mathbf{c}_2{=}[3\;0.5]$. This distribution challenges the Gaussian restriction of the context distribution in the SPRL algorithm. Figures 5 and 6 show a similar picture as in the sparse goal reaching task. CURROT first identifies the easy tasks in which large gates

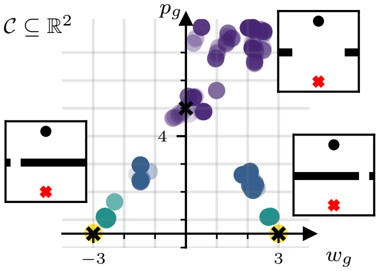
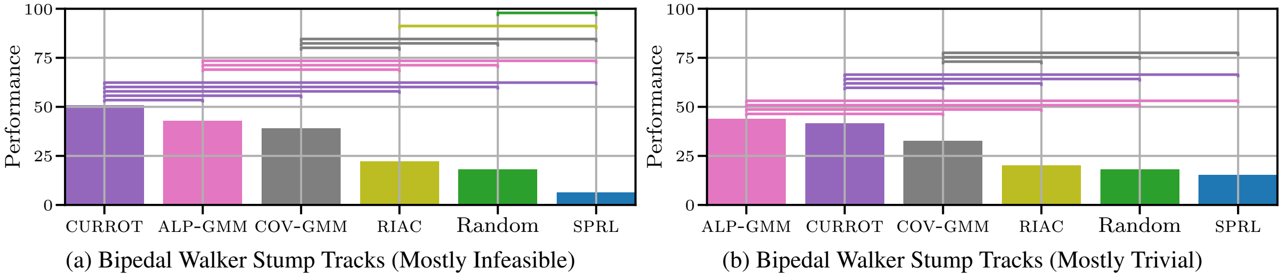


Figure 6. The point mass environment with its two-dimensional context space. The target distribution $\mu(\mathbf{c})$ encodes the two gates with width $w_g{=}0.5$, in which the agent (black dot) is required to navigate through a narrow gate at different positions to reach the goal (red cross). The colored dots visualize a curriculum generated by CURROT for the point mass environment. Each dot represents a particle of the distribution $\hat{p}_{\mathcal{W}}(\mathbf{c})$. The color of the dot indicates the epoch, where brighter colors indicate later epochs.



(a) Bipedal Walker Stump Tracks (Mostly Infeasible)

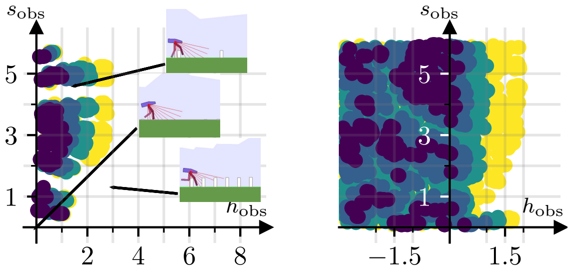
(b) Bipedal Walker Stump Tracks (Mostly Trivial)

Figure 7. Final agent performances (measured in percentage of mastered tasks) on the bipedal walker stump track environment in task spaces with (a) *mostly infeasible* and (b) *mostly trivial* tasks for different CRL methods. Please refer to (Romac et al., 2021) for a detailed explanation of the setup (the evaluation was performed in the *no expert knowledge* setting). Note that all results except for the ones of CURROT are taken from (Romac et al., 2021). Statistics of the CURROT performance have been computed from 32 seeds. Horizontal lines between two methods highlight that their performance was significantly different according to a Welch's t-test with $p < 0.05$. In the *mostly infeasible* setting, CURROT is statistically significantly better than any other method. In the *mostly trivial* setting, CURROT and ALP-GMM are statistically significantly better than all other methods.

rather centered in the middle need to be passed. Starting from those easy tasks, it then creates a curriculum that gradually moves the gate positions to the target ones and decreases the width of the gates. As shown in Appendix D, SPRL proceeds similarly, however only targeting one of the two target gate configurations due to its restriction to a unimodal Gaussian distribution. Looking at Figure 5, we see that the lack of notion of $\mu(\mathbf{c})$ does not allow the other CRL algorithms to sample tasks of increasing similarity to the two target tasks. While particularly ALP-GMM consistently decreases the width of the sampled gates over epochs, it does not necessarily focus on gates that are far away from the center. While this behavior is reasonable and expected given the analysis in Section 4, the visualization of final policies learned with the different curricula in Appendix D shows that this lack of focus on the target tasks leads to less reliable and direct behavior. Summarizing, the first two environments showed the expected benefit of the proposed interpolation-based CRL method in settings, in which $\mu(\mathbf{c})$ is different from the uniform distribution on the whole context space $\mathcal{U}(\mathcal{C})$. In the next environment, we move to a setting in which this difference does not exist and further SPRL has been shown to perform poorly compared to other existing CRL methods.

6.3. Bipedal Walker Stump Tracks

A final environment for evaluation is the modified bipedal walker environment introduced by (Portelas et al., 2019) and extended in (Romac et al., 2021). In this environment a bipedal agent needs to learn to maneuver over a track of evenly spaced obstacles of a specified height (see Figure 8). The context of this environment encodes the spacing of the obstacles as well as their height. The evaluations by (Romac et al., 2021) attested the SPRL algorithm a poor performance, often performing statistically significantly worse than a random curriculum. We revisit two learning scenarios investigated by (Romac et al., 2021) in this environment, in which CRL methods demonstrated a particularly strong 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. Figures 7 and 8 show the sampling distribution of CURROT as well as the resulting performance in comparison to other CRL methods already evaluated by (Romac et al., 2021). We see that CURROT performs either statistically significantly better or statistically insignificantly worse than the best method evaluated by (Romac et al., 2021). These results highlight that empirically successful curricula can be generated by framing CRL as an interpolation between context distributions, even for uniform target context distributions.



(a) Mostly Infeasible

(b) Mostly Trivial

Figure 8. Sampling distribution $\hat{p}_{\mathcal{W}}(\mathbf{c})$ of CURROT on the bipedal walker stump track environment in the *no expert knowledge* setting in task spaces with (a) *mostly infeasible* and (b) *mostly trivial* tasks. Brighter colors indicate samples that correspond to later epochs of agent training. The small image visualize the obstacles that are encoded by the corresponding contexts. For environment details, please see (Romac et al., 2021).

7. Conclusion

In this paper, we proposed a novel approach to curriculum RL that generates a curriculum as an interpolation between task distributions. Opposed to previous methods that aim to minimize the KL divergence to a target distribution of tasks, our method employs Wasserstein distances to measure the difference between the current- and the target task distribution. As we showed, this introduces a notion of metric between training tasks that is missing under the KL divergence. Combined with a performance constraint on the tasks generated by the interpolation, our approach generates curricula that focus on the boundary of agent competence while being able to match non-uniform target distributions. Empirical evaluations highlighted the benefit of the method, matching and surpassing the performance of baseline algorithms. Our findings motivate a variety of future investigations, such as establishing more detailed connections between existing CRL approaches based on surrogate objectives and the newly proposed one. Removing approximations of the current algorithmic realization can improve the fidelity of the interpolating task distribution. Finally, the use of Wasserstein distances opens up a variety of investigations into the use of different metrics to measure the distance between tasks. This promises particularly efficient learning in non-Euclidean context spaces.

Acknowledgments

This project has received funding from the DFG project PA3179/1-1 (ROBOLEAP).

References

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

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

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

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

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

Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

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

Chen, Y., Georgiou, T. T., and Pavon, M. Stochastic control liaisons: Richard sinkhorn meets gaspard monge on a schrödinger bridge. *SIAM Review (SIREV)*, 63(2):249–313, 2021b.

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

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

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

Feydy, J. and Roussillon, P. Geomloss, 2019. URL https://www.kernel-operations.io/geomloss/index.html.

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

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

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

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

Ghavamzadeh, M., Mannor, S., Pineau, J., and Tamar, A. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

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

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2021. URL https://www.gurobi.com.

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

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

Jiang, M., Grefenstette, E., and Rocktäschel, T. Prioritized level replay. In *International Conference on Machine Learning (ICML)*, 2021b.

Jonker, R. and Volgenant, A. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

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

Kantorovich, L. On the transfer of masses (in russian). *Doklady Akademii Nauk*, 37(2):227–229, 1942.

Klink, P., Abdulsamad, H., Belousov, B., and Peters, J. Self-paced contextual reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2020a.

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

Klink, P., Abdulsamad, H., Belousov, B., D'Eramo, C., Peters, J., and Pajarinen, J. A probabilistic interpretation of self-paced learning with applications to reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 22(182):1–52, 2021.

Kolouri, S., Park, S. R., Thorpe, M., Slepcev, D., and Rohde, G. K. Optimal mass transport: Signal processing and machine-learning applications. *IEEE Signal Processing Magazine*, 34(4):43–59, 2017.

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

Kumar, M. P., Packer, B., and Koller, D. Self-paced learning for latent variable models. In *Neural Information Processing Systems (NeurIPS)*, 2010.

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

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

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

Mémoli, F. Gromov–wasserstein distances and the metric approach to object matching. *Foundations of computational mathematics*, 11(4):417–487, 2011.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529, 2015.

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

Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., and Stone, P. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research (JMLR)*, 21(181):1–50, 2020.

Neal, R. M. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

Peyré, G., Cuturi, M., et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

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

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

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

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

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

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

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

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

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

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

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

Weinshall, D. and Amir, D. Theory of curriculum learning, with convex loss functions. *Journal of Machine Learning Research (JMLR)*, 21(222):1–19, 2020.

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

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

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

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

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

A. Computational Complexity of Optimal Transport

The benefits of optimal transport (OT) come at the price of a rather 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 the context spaces investigated in this paper) 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 $\mathcal{O}(n^3)$ (Jonker & Volgenant, 1987). 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 200 samples for 2-D spaces and 500 samples for 3-D spaces, leading to solving times of less than 100ms with the `linear_sum_assignment` function of the SciPy library (Virtanen et al., 2020) on an AMD Ryzen 9 3900X. Since the algorithm presented in Appendix C only solves one OT problem per context distribution update, the computational costs of OT are rather small for the investigated environments.

Furthermore, approximations have emerged to tackle problems that require a large number of particles. For example, the GeomLoss library (Feydy & Roussillon, 2019), that we use in our implementations, 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 (Feydy et al., 2019). So-called sliced Wasserstein distances (Bonneel et al., 2015; Kolouri et al., 2019) 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 $M \ll n$.

B. Context Distributions of CRL Methods

We start the discussion on context distributions of existing CRL methods by restating our assumptions about the learning agent as well as the context space in which the curriculum is generated.

Assumption B.1. We assume a *strong learner* $l : \Pi \times \mathcal{C} \mapsto \Pi$, i.e. a function that maps a given policy $\pi \in \Pi$ and learning task $\mathbf{c}_{\text{learn}} \in \mathcal{C}$ to a new policy $\pi' \in \Pi$ that fulfills

$$(\forall \mathbf{c} \in \mathcal{C} : J(\pi', \mathbf{c}) \geq J(\pi, \mathbf{c})) \wedge (\forall \mathbf{c} \in B_\epsilon(\mathbf{c}_{\text{learn}}) : J(\pi, \mathbf{c}) < J^*(\mathbf{c}) \Rightarrow J(\pi', \mathbf{c}) \geq J(\pi, \mathbf{c}) + \Delta)$$

with $J^*(\mathbf{c})$=$\arg\max_{\pi \in \Pi} J(\pi, \mathbf{c})$, $\epsilon > 0$ and $\Delta > 0$. An epsilon ball $B_\epsilon(\mathbf{c})$ around a context $\mathbf{c}$ is defined as $B_\epsilon(\mathbf{c}) = \{\mathbf{c}' \mid \|\mathbf{c} - \mathbf{c}'\|_2 < \epsilon\}$. Note that we assume a Euclidean space $\mathcal{C}$, although the concept of epsilon balls naturally extend to general metric spaces.

Assumption B.2. The context space $\mathcal{C}$ is a metric compact space in $\mathbb{R}^d$.

Under above assumptions, we first consider the prioritized level replay (PLR) algorithm introduced by (Jiang et al., 2021b). As the authors argue in a follow-up work, PLR can be interpreted to prioritize contexts $\mathbf{c}$ with large regret (Jiang et al., 2021a)

$$\text{Regret}(\pi, \mathbf{c}) = J^*(\mathbf{c}) - J(\pi, \mathbf{c}).$$

Assuming that $p(\mathbf{c})$ of PLR is a monotonically increasing function of $\text{Regret}(\pi, \mathbf{c})$ allows to argue that $p(\mathbf{c})$ ultimately converges to a uniform density over $\mathcal{C}$ for a strong learner. We will only make intuitive arguments here and leave detailed proofs for future work. Given that $\mathcal{C}$ is a compact metric space, we know that every sequence in $\mathcal{C}$ has a converging sub-sequence whose limit is in $\mathcal{C}$. Consequently, we know that there exist contexts $\mathbf{c} \in \mathcal{C}$ for which an increasing number of samples fall into $B_\epsilon(\mathbf{c})$ for any sequence of sampling distributions on $\mathcal{C}$. Hence, there will – after a certain amount of samples – exist contexts for which $\text{Re}(\pi, \mathbf{c})$ is arbitrarily close to zero, causing PLR to prioritize the remaining contexts $\mathbf{c} \in \mathcal{C}$. Continuing this line of thought, the learner will ultimately reach close to maximum performance across all of $\mathcal{C}$, causing $\text{Re}(\pi, \mathbf{c}) \approx 0$ everywhere in $\mathcal{C}$. Consequently, the context distribution of PLR will ultimately converge to an (approximately) uniform distribution over $\mathcal{C}$. Similar arguments can be made for the GOALGAN or ALP-GMM algorithm. As for PLR, there are some abstractions to be made from the algorithmic implementation of these algorithms and their theory. For example, the absolute learning progress metric of ALP-GMM for a rollout $(\mathbf{c}, R_\mathbf{c})$

$$\text{ALP}(\mathbf{c}) = |R_\mathbf{c} - R_{\mathbf{c}_{\text{nn}}}|$$

is computed using the previously completed rollout $(\mathbf{c}_{\text{nn}}, R_{\mathbf{c}_{\text{nn}}})$ with minimal distance $\|\mathbf{c}_{\text{nn}} - \mathbf{c}\|_2$. This computation can be seen as an approximation to the concept of prediction gain $G(\pi, \mathbf{c}) = J(\pi', \mathbf{c}) - J(\pi, \mathbf{c})$, i.e. the performance improvement resulting from learning in $\mathbf{c}$ (see e.g. Graves et al. (2017)). Clearly, $G(\pi, \mathbf{c}) = 0$ for contexts $\mathbf{c}$ with $J(\pi, \mathbf{c}) = J^*(\mathbf{c})$. At this point, the analysis reduces to the one of PLR, as $p(\mathbf{c})$ of ALP-GMM is a monotonically increasing function of $G(\pi, \mathbf{c})$. In general, this analysis may be extended to any CRL algorithm that defines $p(\mathbf{c})$ as a monotonically increasing function of some quantity that decays to zero as $J(\pi, \mathbf{c})$ approaches $J^*(\pi)$.

C. Additional Implementation Details

As mentioned in the main paper, our implementation of Objective (7) relies on three key steps that we want to more closely describe in this appendix: the update of the buffer $\mathcal{B}_+$, computing the next context distribution $\hat{p}_{\mathcal{W}}$ as well as searching for contexts $\mathbf{c}$ with $J(\pi, \mathbf{c}) \geq \delta$.

C.1. Updating $\mathcal{B}_+$

The buffer $\mathcal{B}_+ = \left\{(\mathbf{c}_i, R_{\mathbf{c}_i}) \middle| \mathbf{c}_i \in \mathcal{C}, R_{\mathbf{c}_i} \geq \delta, i \in [1, N_{\mathcal{B}_+}]\right\}$ contains $N_{\mathcal{B}_+}$ episodes for which the agent obtained a return above the desired performance threshold δ. In the algorithm, we limit the size of $\mathcal{B}_+$ to the number of particles N with which we represent the context distribution $p_{\mathcal{W}}$. Initially, $\mathcal{B}_+$ is empty. After completing M episodes during agent training, we add those of the M episodes $(\mathbf{c}, R_{\mathbf{c}})$ with $R_{\mathbf{c}} \geq \delta$ to $\mathcal{B}_+$. At some point, this leads to $N_{\mathcal{B}_+}$ becoming larger than N. If this happens, we sub-sample the episodes in $\mathcal{B}_+$ by solving the following assignment problem

$$\min_{\pi:[1,N_{\mathcal{B}_+}]\times[1,N]\mapsto\{0,1\}} \sum_{i=1}^{N_{\mathcal{B}_+}} \sum_{j=1}^{N} \pi(i,j) d(\mathbf{c}_i, \mathbf{c}_{\mu,j})$$
$$\text{s.t. } \forall j \in [1, N] : \sum_{i=1}^{N_{\mathcal{B}_+}} \pi(i,j) = 1, \qquad \forall i \in [1, N_{\mathcal{B}_+}] : \sum_{j=1}^{N} \pi(i,j) \leq 1,$$

where $\mathbf{c}_{\mu,j} \sim \mu(\mathbf{c})$ are N particles sampled from $\mu(\mathbf{c})$ and $\pi : [1, N_{\mathcal{B}_+}] \times [1, N] \mapsto \{0, 1\}$ is an indicator function that represents the assignment between particles in $\mathcal{B}_+$ and the particles sampled from $\mu(\mathbf{c})$. This transport problem is a variation of the seminal problem of Monge in which we need to select N particles from $N_{\mathcal{B}} > N$ candidates to assign to the target particles. In our implementation, we use the Gurobi optimization software (Gurobi Optimization, LLC, 2021) to solve the above problem. We then select those N particles from $\mathcal{B}_+$, for which there exists a $j \in [1, N]$ with $\pi(i, j) = 1$.

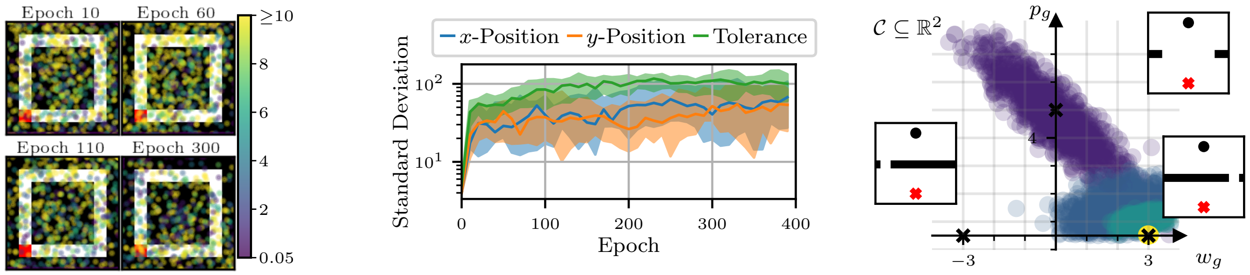
C.2. Updating $\hat{p}_{\mathcal{W}}(\mathbf{c})$

After updating the buffers $\mathcal{B}_+$ and $\mathcal{B}_-$ with the M recently completed episodes and regressing the expected performance $J(\pi, \mathbf{c})$ from the information in $\mathcal{B}_+$ and $\mathcal{B}_-$, we update the position of the particles $\mathbf{c}_{p_{\mathcal{W}},i}$. To do this is in a computationally feasible way, we first sample N particles $\mathbf{c}_{\mu,j}$ from $\mu(\mathbf{c})$ to obtain a particle-based representation of the target distribution $\hat{\mu}(\mathbf{c}) = \frac{1}{N}\sum_{j=1}^{N} \delta_{\mathbf{c}_{\mu,j}}(\mathbf{c})$. We then solve a classical Monge problem to obtain a permutation $\pi : [1, N] \mapsto [1, N]$ that assigns particles of $\hat{p}_{\mathcal{W}}$ to particles of $\hat{\mu}$ and minimizes $\sum_{i=1}^{N} d(\mathbf{c}_{p_{\mathcal{W}},i}, \mathbf{c}_{\mu,\pi(i)})$. We can then express the gradient flow of $p_{\mathcal{W}}(\mathbf{c})$ to $\mu(\mathbf{c})$ under our particle-based representation as

$$T^{\alpha}_{\#}\hat{p}_{\mathcal{W}} = \sum_{i=1}^{M} \delta_{T^{\alpha}(\mathbf{c}_{p_{\mathcal{W}},i})}, \quad T^{\alpha}(\mathbf{c}_{p_{\mathcal{W}},i}) = \mathbf{c}_{p_{\mathcal{W}},i} + \alpha(\mathbf{c}_{\mu,\pi(i)} - \mathbf{c}_{p_{\mathcal{W}},i}), \quad \alpha \in [0, 1]. \tag{8}$$

The above particle flow moves the particles of $\hat{p}_{\mathcal{W}}$ along straight lines to the assigned particles of $\hat{\mu}$. The particular flow arises from our use of the Euclidean metric. For other metrics, the particles would move along the induced geodesics. Having parameterized the flow of the particle-based representation, we can define an approximate version of Objective (7)

$$\min_{\boldsymbol{\alpha}\in[0,1]^N} \left(\frac{1}{N}\sum_{i=1}^{N} d(T^{\alpha_i}(\mathbf{c}_{p_{\mathcal{W}},i}), \mathbf{c}_{\mu,\pi(i)})^2\right)^{\frac{1}{2}}$$
$$\text{s.t. } J(\pi, T^{\alpha_i}(\mathbf{c}_{p_{\mathcal{W}},i})) \geq \delta \qquad \left(\frac{1}{N}\sum_{i=1}^{N} d(\mathbf{c}_{p_{\mathcal{W}},i}, T^{\alpha_i}(\mathbf{c}_{p_{\mathcal{W}},i}))^2\right)^{\frac{1}{2}} \leq \epsilon.$$



(a) SPRL Curriculum (SGR)

(b) SPRL Sampling Distribution Stds. (SGR)

(c) SPRL Curriculum (Point Mass)

Figure 9. a) Visualization of the sampling distribution of SPRL in the sparse goal reaching (SGR) task. The color of the dots encode the tolerance of the corresponding contexts and the position represents the goal to be reached under that tolerance. 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 10 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.

This objective is computationally cheaper compared to Objective (7), since we only parameterize the individual particle movements along the unconstrained gradient flow (8). This approximation to the gradient flow in the constrained setting avoids the recomputation of the permutation π after each gradient step during the optimization and further makes the decision variable $\boldsymbol{\alpha} \in [0, 1]^N$ independent of the dimensionality of the context space $\mathcal{C}$. However, it reduces the fidelity of the solution, as it overly constrains the movement of the particles along the pre-computed geodesics $T^{\alpha}(\mathbf{c}_{p_{\mathcal{W}},i})$.

C.3. Searching for Contexts Satisfying δ

Depending on the learning scenario, the initial agent performance may be below δ for all or most of the initial episodes. In this case, we want to first find tasks $\mathbf{c}$ in which the agent robustly achieves a performance of at least δ. Consequently, our implementation contains an initial search procedure for such tasks that is triggered if not at least half of the first M episodes $(\mathbf{c}, R_{\mathbf{c}})$ fulfill the desired performance threshold. During this procedure, $\mathcal{B}_+$ contains the best samples. When a batch of M new episodes arrives, we add those episodes to the buffer whose return is at least as large as the median return in $\mathcal{B}_+$ – and for each new episode added, remove the worst performing episode. The sampling distribution for the initial search procedure is a (truncated) Gaussian Mixture Model

$$p_{\mathcal{B}_+}(\mathbf{c}) = \sum_{i=1}^{N_{\mathcal{B}}} w_i \mathcal{N}\left(\mathbf{c} \middle| \mathbf{c}_i, \sigma_i^2 \mathbf{I}\right), \quad w_i \propto \max(0, R_{\mathbf{c}_i} - R_{\text{med}}), \quad \sigma_i = \max\left(10^{-3}, 2\frac{\delta - R_{\mathbf{c}_i}}{\delta - R_{\text{min}}}\right),$$

where R_{min} is the minimum return observed over all episodes and R_{med} is the median performance of the episodes currently contained in $\mathcal{B}_+$. 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. Once this search procedure has led to at least half of the samples in $\mathcal{B}_+$ being above the performance threshold δ, we switch to the main algorithm outlined in the paper.

D. Additional Experimental Results

Figure 9 visualizes the behavior of SPRL in the sparse goal reaching (SGR) and point mass environments. 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 9a and 9b,). Given that the policy learned with SPRL performs worse than a policy learned under a curriculum that uniformly samples tasks from the context space, it seems that this Gaussian approximation to a uniform distribution is – at least for this environment – inferior. For the point mass environment, Figure 9c shows that the context distribution of SPRL converges to one of the target tasks encoded by $\mu(\mathbf{c})$.

Figure 10 shows trajectories generated by agents that have been 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

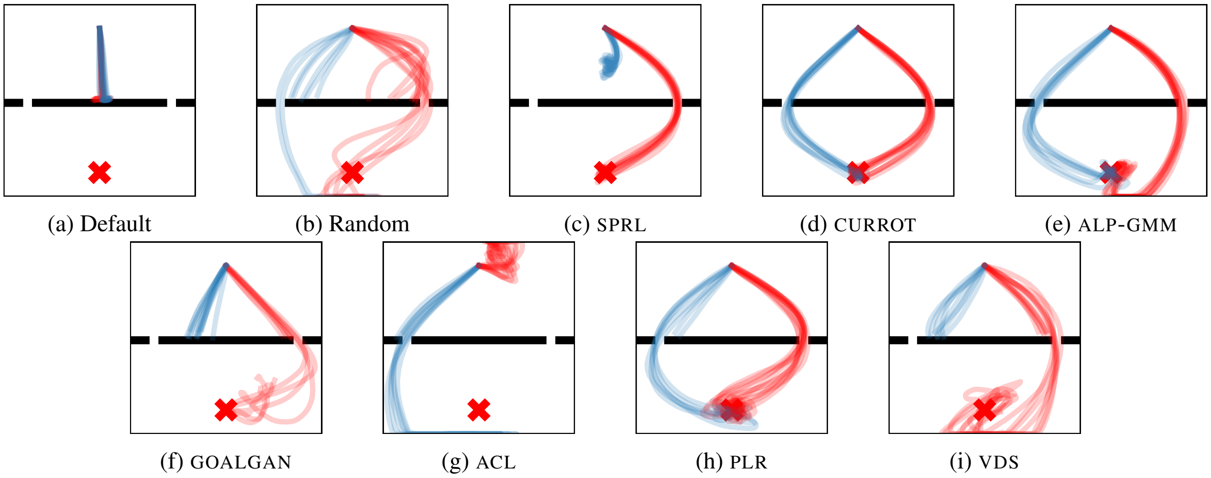

(a) Default (b) Random (c) SPRL (d) CURROT (e) ALP-GMM
(f) GOALGAN (g) ACL (h) PLR (i) VDS

Figure 10. 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 gates positioned at the right.

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 paper, SPRL only learns to pass through one of the gates, as its uni-modal Gaussian distribution can only encode one of the modes of $\mu(\mathbf{c})$. CURROT learns a policy that can pass through both gates reliably, showing that the gradual interpolation towards both target tasks allowed to learn both of them. ALP-GMM and PLR also learn good policies. The generated trajectories are, however, not as precise as the ones learned with CURROT. ACL, GOALGAN, and VDS partly create meaningful behavior. However, this behavior is unreliable and hence leads to low returns due to the agent frequently crashing into the wall.

E. 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.

E.1. Algorithm Hyperparameters

The two main parameters of the SPRL algorithms are the performance threshold δ as well as the allowed distance between subsequent distributions ϵ. We did not perform an extensive hyperparameter search for these parameters but chose them as follows: The performance threshold δ is chosen such that it is around 50% of the maximally achievable reward. We then evaluated a larger and a lower value of the parameters and chose the best. For CURROT, the parameter ϵ is chosen such that it is around 5% of the maximum distance between any two points in the context space. Since the context spaces in the considered environments are d-dimensional intervals $[l_1, h_1] \times [l_2, h_2] ... \times [l_d, h_d]$, this means $\epsilon = 0.05\|\mathbf{h} - \mathbf{l}\|_2$. For SPRL, we initialized ϵ with value of 0.05 used in the initial experiments by Klink et al. However, we realized that larger values slightly improved performance. When targeting narrow target distributions, Klink et al. introduce a lower bound

	SPRL				CURROT	
ENV.	δ	ϵ	$\boldsymbol{\sigma}_{\text{LB}}$	$D_{\text{KL}_{\text{LB}}}$	δ	ϵ
SPARSE GOAL REACHING	0.6	.25	-	-	0.6	1.5
POINT MASS	4	.25	[.2 .1875]	8000	4	0.5
BIPEDAL WALKER	-	-	-	-	180	0.5|0.4

Table 1. Hyperparameters of the SPRL and CURROT in the different learning environments. The ϵ parameter of CURROT is computed according to the procedure described in appendix E. Note that for the bipedal walker environment, we do not provide the parameters for SPRL as we rely on the results reported by (Romac et al., 2021).

	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

Table 2. Hyperparameters of the investigated baseline algorithms in the different learning environments, as described in appendix E.

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 to early, causing the KL divergence constraint on subsequent distributions to only allow for very small 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 1 shows the parameters of CURROT 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} as well as the maximum buffer size of past trajectories to keep s_{buffer}. Similar to Klink et al. (2021), 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 the updates of 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 conduct 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. Implementing the ACL algorithm by (Graves et al., 2017), we had numerical issues 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 as well as 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 n_{ep} and the number of minibatches 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 tables 2 and 3.

E.2. Task Descriptions

We now detail on the individual experiments, such as context-, state- and action spaces as well as the employed RL algorithms. As RL agents, we use SAC and PPO implemented in the `Stable Baselines 3` library (Raffin et al., 2021) for the sparse goal reaching and point mass environment. For the bipedal walker stump track environment, we use the SAC implementation provided by (Romac et al., 2021).

E.2.1. Sparse Goal Reaching

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 maze. The target context distribution is a uniform distribution over tasks with a tolerance of 0.05

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

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

Table 3. Hyperparameters of the investigated baseline algorithms in the different learning environments, as described in appendix E.

The state s of the environment is given by the x- and y-position of the agent. The reward is sparse, only rewarding the agent if the goal is reached. A goal is considered reached if the Euclidean distance between 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.
We use the SAC algorithm for learning in this task. 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.

E.2.2. POINT MASS

The environment setup is the same as the one investigated by Klink et al. (2020b; 2021) 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 λ=0.99. All other parameters are left to the implementation defaults of the `Stable Baselines 3` implementation.

E.2.3. BIPEDAL WALKER STUMP TRACKS

As mentioned in the main paper, we used the environment and SAC learning agent implementation provided by Romac et al. (2021). We only interfaced CURROT to the setup provided by them, allowing to reuse the baseline evaluations provided by Romac et al. (2021). 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, however encodes stumps with a height in $[-3, 3]$. Since a stump with negative height is considered as not present at all, half of the context space of the *mostly trivial* setting does not encode any obstacles for the bipedal walker to master. The target context distribution $\mu(\mathbf{c})$ is uniform over the respective context space $\mathcal{C}$ for both settings.