Exploration-Driven Representation Learning in Reinforcement Learning

Akram Erraqabi [1 2] **Mingde Zhao** [1 3] **Marlos C. Machado** [4 5 6] **Yoshua Bengio** [1 2 6] **Sainbayar Sukhbaatar** [7]
Ludovic Denoyer [7] **Alessandro Lazaric** [7]

Abstract

Learning reward-agnostic representations is an emerging paradigm in reinforcement learning. These representations can be leveraged for several purposes ranging from reward shaping to option discovery. Nevertheless, in order to learn such representations, existing methods often rely on assuming uniform access to the state space. Without such a privilege, the agent's coverage of the environment can be limited which hurts the quality of the learned representations. In this work, we introduce a method that explicitly couples representation learning with exploration when the agent is not provided with a uniform prior over the state space. Our method learns representations that constantly drive exploration while the data generated by the agent's exploratory behavior drives the learning of better representations. We empirically validate our approach in goal-achieving tasks, demonstrating that the learned representation captures the dynamics of the environment, leads to more accurate value estimation, and to faster credit assignment, both when used for control and for reward shaping. Finally, the exploratory policy that emerges from our approach proves to be successful at continuous navigation tasks with sparse rewards.

1. Introduction

Representation learning has been at the core of many recent machine learning advances (c.f. Bengio et al., 2013). With the advent of deep reinforcement learning (RL) (Mnih et al., 2015), representation learning has also become one of the main topics of interest in RL. For example, in the goal-conditioned hierarchical setting (Vezhnevets et al., 2017; Nachum et al., 2019a), one learns a *representation* which maps state observations to an abstract space, the representation space, in which the higher-level policy defines the desired behavior of the lower-level policy. Distance in the representation space can then be used to reward and guide the lower-level policy towards specific goal states. Representation learning was also shown to be crucial in environments with rich observations and complex dynamics (e.g., Bellemare et al., 2020). This has motivated recent works about learning representations that capture controllable or contingent features (Bengio et al., 2017; Choi et al., 2019), on top of which one can potentially learn latent models in the perspective of planning (Hafner et al., 2019b; Nasiriany et al., 2019; Schrittwieser et al., 2020) and control (Watter et al., 2015; Banijamali et al., 2018; Hafner et al., 2019a).

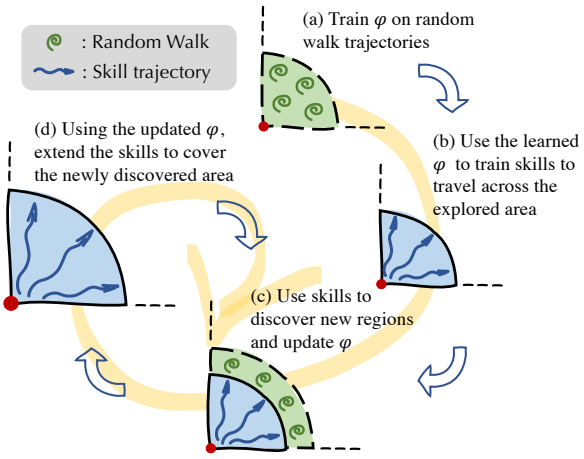


Figure 1. The representation is trained to encode the area that the agent has learned to cover. Skills are continuously trained on the representation to discover new areas where novel data is collected to refine the representation, progressively extending its coverage. Inpsired by Machado (2019).

In this work, we are interested in the reward-agnostic setting in which an RL agent first interacts with the environment to build a representation ϕ of the state space $\mathcal{S}$ without relying on any task-specific reward signal. This representation can later be used to solve tasks posed in the environment in the form of reward functions. In this setting, the agent can learn about the environment dynamics and its structure.

[1]Mila [2]Université de Montréal [3]McGill University [4]Amii [5]University of Alberta [6]CIFAR Fellow [7]Facebook AI Research. Correspondence to: Akram Erraqabi <akram.erraqabi@mila.quebec>.

Accepted at the Unsupervised Reinforcement Learning workshop, ICML 2021.

Moreover, we are specifically interested in a family of representation learning approaches based on contrastive losses that require the ability to *uniformly* sample pairs of similar and dissimilar data-points from the available data. This sampling is trivial in vision and text settings as one has a direct access to the whole image dataset or text corpus. This is not as trivial in the RL setting given that the agent has to learn to explore the state space to be able to access arbitrary states.

An illustrative example of this class of approaches is the recently proposed stochastic approximation of the Laplacian representation (Wu et al., 2019). By framing the graph drawing objective as a contrastive *pair sampling-based* loss, it generalizes spectral representations beyond the tabular case while helping to overcome potentially prohibitive eigendecompositions. However, it assumes access to a uniform sampling prior over $\mathcal{S}$. Practically, this translates in the ability to reset the agent to a uniformly random starting state in the environment, which artificially alleviates the exploration problem. As we will show later, the uniformity of that distribution is crucial for the quality of the learned representation. Thus, in the absence of the uniform prior privilege, one must handle the exploration along with the representation learning in order to preserve the representational quality. In this work, we propose a representation learning framework that conciliate similar contrastive approaches with exploration in the reward-agnostic setting.

In practice, the representation is trained on data collected with a uniformly random policy μ (random walk trajectories). Without a uniform access to the state space, the collected data would only cover a limited area around the accessible starting states (coverage area scales as $\sqrt{n}$ with n steps long trajectories). To achieve a better data collection, we propose to tie the representation learning problem to that of learning a covering strategy. Our approach as illustrated in Figure 1 is inspired by the cyclic option discovery framework proposed by Machado (2019) which has also motivated several recent option discovery approaches (Machado et al., 2017; 2018; Jinnai et al., 2020). Our work is an attempt to put such framework at the service of representation learning.

Briefly, our method consists in learning a skill-based covering strategy along with the representation learning. The representation is used to train directional skills to cover the explored area, while the skills are used to discover yet unseen parts of the state space providing novel data to refine and extend the representation. We also propose to augment the representation learning objective with a term reflecting the dynamics allowed by the skills; a trained skill brings its initiation and termination areas closer in terms of dynamics. In effect, beyond encouraging exploration, this augmentation enforces the representation's *dynamics-awareness*, by improving how the representation induced metric captures distances along the environment dynamics.

We empirically show our agent's ability to progressively explore the state space and steadily extend the representation domain. We show that even without a non-uniform prior over the state space our representation leads to better value predictions than the Laplacian representation and recovers the representational quality that a uniform prior would provide. We also evaluate our representation in shaping rewards for goal-achieving tasks and show that it outperforms existing techniques, confirming its higher ability in capturing dynamics. Finally, the skills learned in our framework also prove to be competitive as they were, among the evaluated methods, the only ones successful at a hard continuous navigation task with **sparse** rewards.

2. Preliminaries

2.1. Task-agnostic Reinforcement Learning

We describe a task-agnostic RL environment as a task-agnostic Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \gamma, d_0)$ where $\mathcal{S}$ is state space, $\mathcal{A}$ the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is transition dynamics defining the next state distribution given current state and taken action, $\gamma \in [0, 1)$ is the discount factor, and d_0 the initial state distribution. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ maps states $s \in \mathcal{S}$ to distributions over actions.

Any knowledge acquired from task-agnostic interactions with the environment (e.g. in the form of a representation or a policy) can then be leveraged for specific tasks. For a given task, the associated reward function $R : \mathcal{S} \to \mathbb{R}$ combined with the task-agnostic MDP define the task objective as finding the optimal policy maximizing the expected return $\mathbb{E}_{\pi, d_0}\left[\sum_t \gamma^t R(s_t, a_t)\right]$ starting from the state $s_0 \sim d_0$ and acting according to $a_t \sim \pi(\cdot|s_t)$.

2.2. Unsupervised Representation Learning

Contrastive losses are at the core of the recent notable advances in representation learning (Bachman et al., 2019; He et al., 2020; Chen et al., 2020; Grill et al., 2020). They are well-suited to the generic unsupervised learning paradigm. These losses are usually comprised of an attractive term and a repulsive one, where the former guarantees similar samples to have close representations while the later spreads significantly different samples' representations far apart. In temporal settings, they are used to learn *slow* features that preserve temporal coherence (Wiskott & Sejnowski, 2002) which makes them relevant to RL as well (Wu et al., 2019; Li et al., 2021).

Wu et al. (2019) showed the competitive representational capacity of the Laplacian representation (Lap-rep) when provided with a uniform prior over $\mathcal{S}$. Note that their proposed objective is also a contrastive loss whose repulsive term was derived from the orthonormality constraint of the Laplacian eigenfunctions. In our framework, the representation would be continuously used to learn a progressive exploration strategy. In such scenario, the orthonormality constraint could make the online representation learning highly non-stationary, and thus hurt the continual exploration.[1] For this reason, we adopt a more generic repulsive term in favor of more stability. More precisely, we will consider the following objective where the repulsive term is a smoother version of the one proposed by Li et al. (2021):

$$\mathcal{L}_{cont}(\phi) = \mathbb{E}_{(u\sim d_\mu, v\sim P^\mu(\cdot|u)}\left[\|\phi(u)-\phi(v)\|_2^2\right] + \beta\, \mathbb{E}_{u\sim d_\mu, v\sim d_\mu}\left[\exp(-\|\phi(u)-\phi(v)\|_2)\right], \quad (1)$$

with $\phi : \mathcal{S} \to \mathbb{R}^d$, μ the uniformly random policy, d_μ the corresponding stationary distribution over the state space $\mathcal{S}$ (uniform prior) and β a hyperparameter.

3. Exploration-driven Representation Learning

In RL, exploration is deeply coupled to the problem of representation learning because it defines the data distribution used during learning. In the task-agnostic setting, this distribution is usually required to be uniform as in the objective (1). Unless the agent can teleport to any state – which would actually alleviate the exploration problem –, it is not trivial to collect uniformly distributed samples. Indeed, a random walk from a given initial state would generally take relatively long times (the mixing time) to visit a finite volume of the state space. Therefore, the induced visitation distribution can be quite concentrated around the start state distribution when solely relying on random walks.

The non-uniform prior setting. To study the problem described above, we define a setting where the environment has a fixed predefined state s_0 to which it resets with a probability p_{reset} every K steps. With a uniformly random behavior policy, this setting is equivalent to a initial state distribution $d_{n\mu}$ that is concentrated around s_0 and whose density decays exponentially away from it. We will refer to this setting as the *non-uniform prior* setting, as opposed to the *uniform prior* setting where the agent has access to the uniform state distribution d_μ.

In this section, we present our representation learning framework in which we propose a method that leverages the representation to learn a skill-based covering strategy which in turn provides better data collection to train the representation. We first describe the exploratory component, focusing on the choice of its design and its training. Then, we introduce an augmentation of the representation learning objective that improves exploration and enforces the representation's dynamics-awareness. We conclude this section by synthesizing the proposed algorithm in the non-uniform prior setting.

3.1. Representation-based Covering Policy

To achieve better exploration of the environment, we adopt a hierarchical RL agent to leverage the exploratory efficiency of temporally-extended actions or skills (Sutton et al., 1999; Nachum et al., 2019b). Concretely, the agent acts according to a bi-level policy $(\pi_{\text{hi}}, \pi_{\text{low}})$. The high-level policy $\pi_{\text{hi}} : \mathcal{S} \to \Delta(\Omega)$ defines, at each state s, a distribution over a set Ω of unit vectors in the representation space ($\Omega = \{\boldsymbol{\delta} \mid \boldsymbol{\delta} \in \mathbb{R}^d, \|\boldsymbol{\delta}\|_2{=}1\}$). The low-level policy $\pi_{\text{low}} : \mathcal{S} \times \Omega \to \Delta(\mathcal{A})$ encodes fixed length skills that are expected to travel *in the representation space* along the directions instructed by π_{hi}. In short, given a sampled direction $\pi_{\text{hi}}(\cdot|s) \sim \boldsymbol{\delta} \in \Omega$, the low-level policy executes the directional skill $\pi_{\text{low}}(\cdot|s, \boldsymbol{\delta})$ for a fixed number of steps c before a new direction is sampled.

Now, we describe the intrinsic rewards used to train these policies. π_{hi} and π_{low}.

Low-level Policy. π_{low} is simply trained to follow directions defined by π_{hi} in the representation space. For a given $\boldsymbol{\delta} \in \Omega \subset \mathbb{R}^d$, the corresponding skill $\pi_{\text{low}}(\cdot|s, \boldsymbol{\delta})$ is trained to maximize the intrinsic reward function:

$$r^{\boldsymbol{\delta}}(s, s') = \frac{\boldsymbol{\delta}^\top(\phi(s') - \phi(s))}{\|\phi(s') - \phi(s)\|} = \cos(\boldsymbol{\delta}, \phi(s') - \phi(s)) \quad (2)$$

where (s, s') is an observed state transition and ϕ the representation being learned. We use the cosine similarity to ensure we only reward the agent for steps in the instructed direction $\boldsymbol{\delta}$, regardless of their magnitude.

High-level Policy. The high-level policy is expected to guide the covering strategy. It should do so by sampling the skills of the most promising directions in terms of exploration, fostering new discoveries while avoiding to spend more time than needed in previously explored areas. For this purpose, we design a reward function defined over a sequence of $L{=}\lceil K/c \rceil$ consecutive skills. Let $\{s_k^{\text{hi}}\}_{k=1}^L$ be the sequence of states where their directions were sampled $\boldsymbol{\delta}_k \sim \pi_{\text{hi}}(\cdot|s_k^{\text{hi}})$. Since the representation is trained to capture the dynamics, the travelled distance in the representation space is a good proxy of how far the choices made by π_{hi} eventually brought the agent in the environment. Therefore, for a given high-level trajectory $\tau^{\text{hi}} = (s_1^{\text{hi}}, s_2^{\text{hi}}, ..., s_L^{\text{hi}}, s_f^{\text{hi}})$,

[1] In general, even along a smooth update of the Laplacian, maintaining the orthonormality of the eigenvectors could considerably change all of them.

with s_f^{hi} the final state reached by the last skill, the high-level policy is trained to maximize the following quantity:

$$\forall k \in \{1, ..., L\}, R^{\text{hi}}(s_k^{\text{hi}}, \boldsymbol{\delta}_k) = \|\phi(s_1^{\text{hi}}) - \phi(s_f^{\text{hi}})\|, \quad (3)$$

where $\boldsymbol{\delta}_k \sim \pi_{\text{hi}}(\cdot|s_k^{\text{hi}})$ is the direction sampled at s_k^{hi}. This term looks at reaching s_f^{hi} as the result of a sequential collaboration of L skills and rewards them equally. It only values how far this sequence of skills brought the agent.

These policy training choices are closely related to how the representation is trained as well. Indeed, the desired exploratory behavior emerges from a closed-loop interaction between the policy and the representation while training. In the following section, we describe how the representation benefits in its turn from the skill-based exploration policy.

3.2. Augmented Representation Learning Objective

In practice, representation is trained on batches of random walk trajectories collected over the area that the agent learned to cover with $(\pi_{\text{hi}}, \pi_{\text{low}})$. Let $\mathcal{D}_\mu$ be a batch of such trajectories. As the agent tends to uniformly extend its covering, the objective (1) can be approximated by the following batch-specific loss:

$$\tilde{\mathcal{L}}_{cont}(\phi; \mathcal{D}_\mu) = \mathbb{E}_{(u,v)\sim\mathcal{D}_\mu}\left[\|\phi(u) - \phi(v)\|_2^2\right] + \beta\, \mathbb{E}_{u\sim\mathcal{D}_\mu, v\sim\mathcal{D}_\mu}\left[\exp(-\|\phi(u) - \phi(v)\|_2)\right]. \quad (4)$$

While the directional skills are trained on the area covered by ϕ, this representation is continuously refined to integrate the environment dynamics knowledge captured in these skills. By construction, a skill extends the area reachable in a single (macro-)decision. In terms of dynamics, it brings its start state s_i^{low} and terminal state s_f^{low} – and their respective neighbourhoods – closer. This property can be enforced into the representation space by simply minimizing $\|\phi(s_i^{\text{low}}) - \phi(s_f^{\text{low}})\|$. However, to preserve the local representation structure and avoid representation collapse, we instead minimize a contracting term along the skills trajectories:

$$\mathcal{B}(\phi; \mathcal{D}_s) = \mathbb{E}_{\substack{\tau_{\boldsymbol{\delta}}\sim\mathcal{D}_s \\ \tau_{\boldsymbol{\delta}}=(s_0,...,s_c)}}\left[\sum_{k=0}^{c-1} \|\phi(s_k) - \phi(s_{k+1})\|\right], \quad (5)$$

where $\mathcal{D}_s$ is a set of collected skills trajectories. Recall that for any skill trajectory $\tau_{\boldsymbol{\delta}} = (s_0, ..., s_c) \in \mathcal{D}_s$, $\boldsymbol{\delta}$ denotes its direction, such as we have $s_{k+1} \sim \pi_{\text{low}}(\cdot|s_k, \boldsymbol{\delta})$ for every $k \in \{0, ..., c-1\}$.

Closed-loop training fuels exploration. Combined with the choice of the high-level policy reward (3), this representation contracting term (5) induces a progressive exploration mechanism. In effect, π_{hi} would be more often sampling skills that travel further, i.e. with larger R^{hi} (3). The more a skill is sampled, the less rewarding it becomes due the minimization of $\mathcal{B}(\phi)$ (5). This will increase the probability of sampling the remaining potentially under-sampled skills, fostering more opportunities to explore less visited parts of the state space. In short, the interplay between the policy and the representation dynamically fights what can be considered as accumulated *boredom* along over-sampled skills trajectories which increases the agent curiosity and urge it to explore.

Finally, the proposed objective to train the representation ϕ consists in the objective (4) augmented with the boredom term (5), and can be written as

$$\mathcal{L}_{\text{rep}}(\phi; \mathcal{D}_s, \mathcal{D}_\mu) = \tilde{\mathcal{L}}_{cont}(\phi; \mathcal{D}_\mu) + \beta' \mathcal{B}(\phi; \mathcal{D}_s) \quad (6)$$

with β' a hyperparameter controlling the strength of boredom term.

3.3. Representation Learning in the non-uniform Prior setting

The proposed approach consists in a simultaneously training of the representation ϕ and the hierarchical agent $(\pi_{\text{low}}, \pi_{\text{hi}})$. The idea is to progressively extend the explored area while maintaining the previously collected knowledge. To do so, in the non-uniform prior setting, the agent switches with some probability p_{rw} between following a uniformly random policy μ and executing the hierarchical policy (skills). The latter helps reach further areas, more efficiently, where data collected by random walks would be used to train the representation ϕ. Along their training, the skills would progressively extend to reach newly discovered areas, advancing the exploration frontier. Algorithm 1 describes the proposed approach in the non-uniform prior setting.

4. Experiments

In this section, we investigate the behavior of the proposed algorithm in two types of environments: gridworld environments with discrete state and action spaces, and a continuous navigation environment (MuJoCo (Todorov et al., 2012)) for continuous state and action spaces. Implementation details of all the experiments in this section can be found in the Appendix.

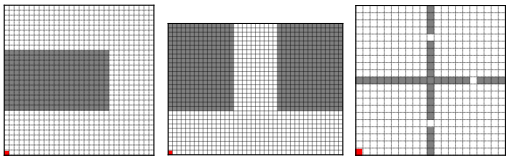
4.1. GridWorld

For gridworld environments, we evaluate on three different domains: U-MAZE, T-MAZE and 4-ROOMS. These environments, visualized in Figure 2, raise different explorations challenges. U-MAZE is perhaps the simplest but the most relevant environment to test the dynamics-awareness of the learned representations; T-MAZE raises the challenge of splitting the exploration focus at an intersection while

Algorithm 1 Exploration-driven representation learning in the non-uniform prior setting

Input: L, c, p_{rw}, N
for $iteration = 1, 2, \ldots$ **do**
  $D_\mu = \emptyset$, $D_s = \emptyset$
  for $batch = 1, 2, \ldots, N$ **do**
    Reset to s_0 with probability p_{reset}.
    $p \sim$ *Unif*([0,1])
    if $p < p_{rw}$ **then**
      Run the uniformly random policy μ to collect L random walk trajectories $\{\tau'_i\}_{i=1}^{L}$ of c steps each.
      $D_\mu \leftarrow D_\mu \cup \{\tau'_k\}_{k=1}^{L}$
    else
      Run $(\pi_{\text{hi}}, \pi_{\text{low}})$ to collect L consecutive skills' trajectories $\{(\tau_k, \boldsymbol{\delta}_k)\}_{k=1}^{L}$ and their corresponding directions
      $D_s \leftarrow D_s \cup \{(\tau_k, \boldsymbol{\delta}_k)\}_{k=1}^{L}$
    end if
  end for
  Optimize the policies $(\pi_{\text{hi}}, \pi_{\text{low}})$ using their intrinsic objectives 3 and 2 (vanilla actor-critic update)
  Optimize ϕ so as to minimize $\mathcal{L}_{\text{rep}}(\phi; \mathcal{D}_s, \mathcal{D}_\mu)$
end for

maintaining exploration and coverage in both corridors; 4-ROOMS is similar to U-MAZE, but may require learning more controlled skills to efficiently move from one room to another.



(a) U-Maze 30×30 (b) T-Maze 40×30 (c) 4-rooms 21×21

Figure 2. The gridworld domains with the fixed initial state s_0 highlighted in red.

The states are one-hot encoded such that no positional information is provided to the agent. For our method, we learn a 2D representation ($d = 2$), and define Ω as a set of 8 unit vectors equally spaced on the unit sphere.

4.1.1. PROGRESSIVE REPRESENTATION LEARNING

Figure 3 shows the progression of the representations throughout the training. The agent progressively explores the environment starting from s_0, builds the representation by continuously integrating newly discovered parts.

U-MAZE. The agent starts from the bottom left corner of the maze. Figure 3 shows how the representation progressively expands till reaching the first corner (R1 (a-e)). During this phase the agent learns skills to travel further away from s_0 along the corridor. At this stage, the rest of the environment is still not explored; note how its representation falls back in the compressed cluster of unseen states. The remaining exploration phase (R1 (f-j)) shows not only the complete discovery of the corridor but also the flattening of the full domain representation. This indicates the representation's success in capturing the dynamics, by placing the last corner further from the starting corner than the intermediate ones.

T-MAZE. The agent starts from the bottom left corner of the maze. As in the U-Maze, it starts learning to travel along that corridor (R2 (a-b)) until reaching the intersection. There, the exploration focus is shared between both possible paths whose representations are progressively disentangled (R2 (c-f)). Eventually, the agent fully explores both corridors and finalizes their representations. Note that, the discovery of one of the corridors did not hinder finishing the discovery of the other.

4-ROOMS. The agents starts in the first room. It progressively discovers and learns about the rooms. Once the domain is fully explored, and similarly to U-MAZE, the representation straightens, reflecting the environment dynamics.

We have also conducted an ablation study to validate the importance of the boredom term to the agent's exploratory behavior, and the dynamics-awareness of the representation (see Appendix A).

4.1.2. EVALUATING THE LEARNED REPRESENTATION

We choose to compare our representation against the Laplacian representation (Lap-rep) (Wu et al., 2019) since it is a representative approach relying on the uniform prior assumption which we are addressing in this work. First, to appreciate the sensitivity of Lap-rep to the said prior, Lap-rep was learned in the two different settings defined in Section 3: (i) the uniform prior setting where the agent can be set to any arbitrary state as in Wu et al. (2019), (ii) and then, the non-uniform prior setting in which our representation is learned (resetting to the fixed state s_0 with probability p_{reset}), which induces a concentrated episode's starting positions distribution around s_0. We find Lap-Rep to be extremely sensitive to this change, which is reflected in the following experiments.

Linear Function Approximation. To evaluate the learned representations, we first consider how well they can be used as input features to linearly approximate a given task's optimal value function. To do so, we train an actor-critic agent with a linear critic on top of the learned representations. In Figure 4, we note a significant loss in the representational power of the Laplacian method when the access to the state

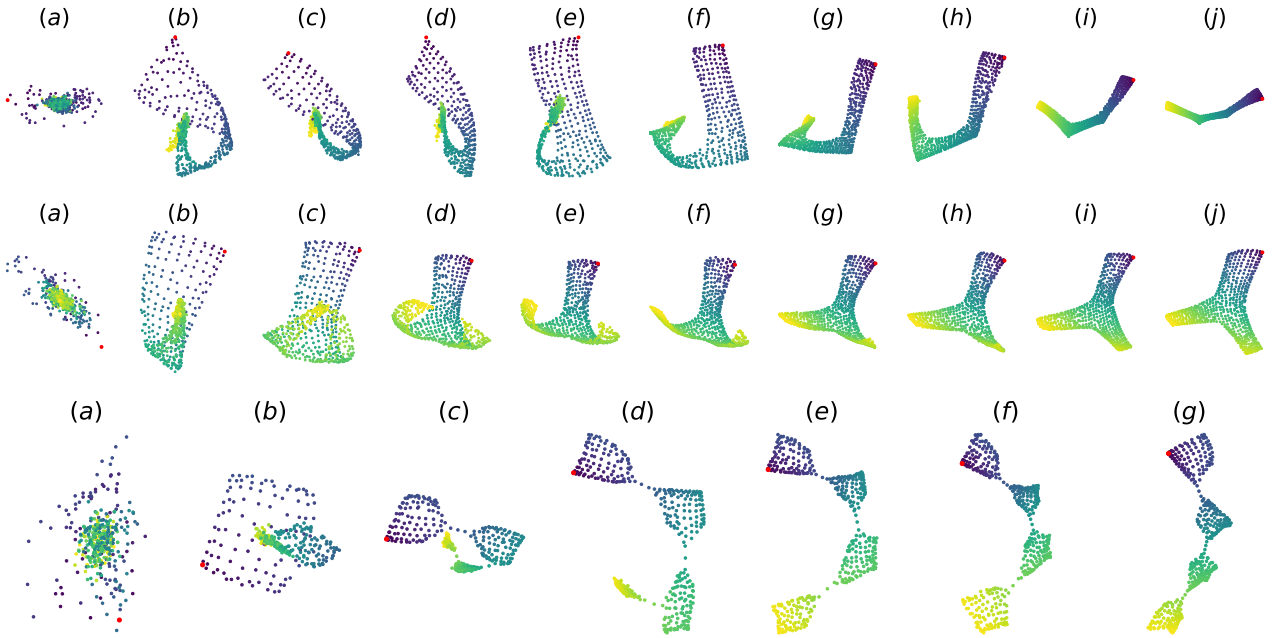


Figure 3. The domains representations learned throughout the training. Top row (**R1**): U-MAZE. Middle row (**R2**): T-MAZE. Bottom row (**R3**): 4-ROOMS. The colors reflect the distances in terms of the dynamics. They can be seen as quantities proportional to the length of the shortest path from s_0 (marked in red) to the represented state.

space is no more uniformly distributed. This figure shows that our representation outperforms the Lap-rep learn with the same initial state distribution, and succeeds in recovering Lap-rep's representational power when this one is learned from the *unrealistic* uniform prior.

Control. We also compare the representations from the perspective of control, by training a deep (i.e., non-linear) actor-critic agent on top of each representation to solve a goal-reaching task in the same environments as above. The agent is only rewarded ($r = 1$) upon reaching the goal state. Figure 5 shows that our representation consistently outperforms the Laplacian representation, which confirms the competitive quality of our representation.

4.2. Continuous Control

The second set of experiments, which focuses on continuous state and action spaces, is conducted on the AntMaze continuous control environment. AntMaze is essentially a MuJoCo counterpart of U-MAZE where a four-legged agent has to learn to control its joints to maneuver along a U-shaped corridor.

To visualize the learned representation in this environment, in Figure 6 we show a set of positional states (defined by a grid over the state space) in both the environment domain and their mapped representations. Similarly to U-MAZE, the learned representation translates the dynamics of the environments by pushing the end of the corridor (top left) away from the initial state (bottom left), represented in red.

4.2.1. REWARD SHAPING WITH LEARNED REPRESENTATION

Following Wu et al. (2019), to demonstrate the ability of our learned representations to improve an RL agent's performance, we evaluate them in a goal-achieving task using reward shaping. We define a goal-achieving task by defining a goal state g in the rewardless environment. This goal is set at the upper end of the corridor. The objective is to learn to control the agent and navigate to a state s close enough to the goal ($\|s - g\|_2 \leq \epsilon$). A **sparse** reward $r_t = \mathbb{1}\left[\|s_{t+1} - g\|_2 \leq \epsilon\right]$, would not provide enough signal to properly guide the agent, and a dense L2 distance-based reward would be deceptive since it does not take the dynamics into account (the presence of a wall). For this, we define reward functions based on distance in the representations spaces (ours and Lap-rep's), learned in the non-uniform prior setting, to train a soft actor-critic (SAC) agent (Haarnoja et al., 2018) to reach the goal (details provided in the Appendix). More specifically, we define the **dense** reward as $r_t^{dense} = -\|\phi(s_{t+1}) - \phi(g)\|$. Following previous work (Wu et al., 2019), we also compare against the half-half **mix** of the dense reward and the sparse reward $r_t^{mix} = 0.5 \cdot r_t^{dense} + 0.5 \cdot \mathbb{1}\left[\|s_{t+1} - g\|_2 \leq \epsilon\right]$.

Here, the Laplacian representation (Lap-rep) was learned

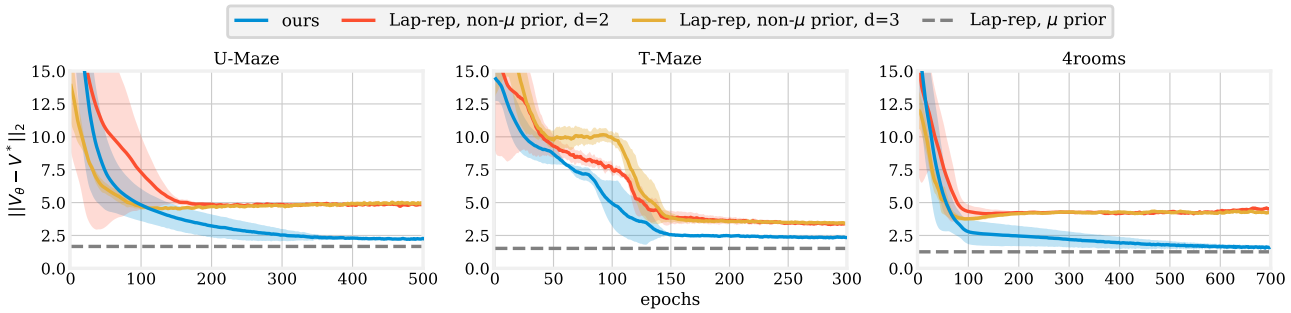


Figure 4. Linear function approximation on top of learned representation. The Laplacian representation (Lap-rep) was learned in the same non-uniform prior setting (non-μ) with $d = 2$ and $d = 3$ (no improvement was observed for higher values). The dashed line gives the performance of Lap-rep in the uniform prior setting (μ). Our representation outperforms Lap-rep in non-μ setting, and succeeds in recovering Lap-rep's representational power when learned from the *unrealistic* uniform prior. The curves and the corresponding confidence intervals are obtained from 5 different runs.

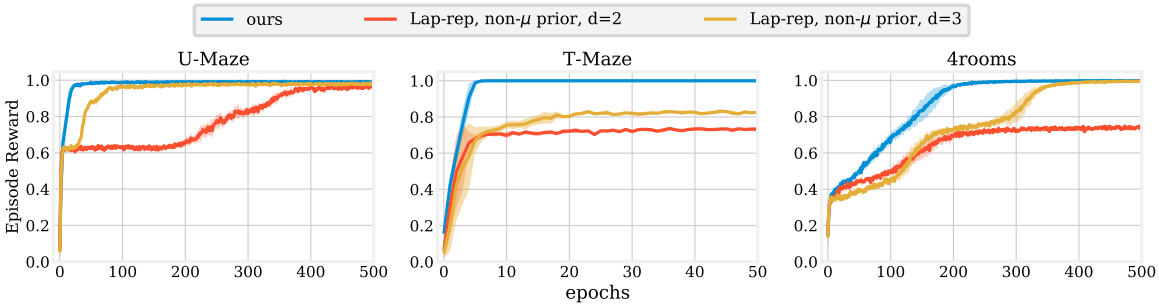


Figure 5. Control performance (episode reward) in the fixed initial state setting (non-uniform prior). The curves and the corresponding confidence intervals are obtained from 5 different runs.

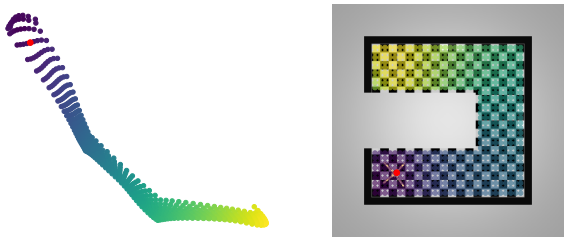


Figure 6. Visualization of the learned representation on a grid of positional states. Colors reflect the distance in the representation space from the initial state highlighted in red. We can visually appreciate how the U-shaped state domain is mapped to a flatter manifold reflecting the presence of the wall: our method succeeds capturing the dynamics of the continuous state space.

with uniform prior over the state space as in Wu et al. (2019). In this challenging environment with continuous state and actions spaces, Lap-rep fails at capturing the continuous and smooth dynamics uniformly over $\mathcal{S}$, which explains its inability to consistently guide the agent to reach the goal on the other end of the maze; note that if the shaping representation suffers from singularities even in a small but decisive area on the way to the end of the maze, the agent won't be properly guided to it. This shows the brittleness of Lap-rep in large continuous state spaces even with the uniform prior privilege. As shown in Figure 7, our representation is effective in reward shaping, with both **mix** and **dense** variants, which further confirms its benefits even in the challenging non-uniform prior setting.

4.2.2. The Learned Skills

We validate if our approach generates skills that are useful to exploring the state space. Here, we compare these skills against DCO, an hierarchical skill discovery method (Jinnai et al., 2020). This skills' training requires a pre-trained Laplacian representation (which approximates the Laplacian's second eigenvector). Here, we train the required representation, as well as the options, with data collected from a uniform prior over the state space. For fairness, we also train a DCO agent to learn 8 options. Low-level policies of both agents, ours and DCO's, are then fixed and used to train a discrete high-level policy to solve the goal-reaching task presented above with only the sparse reward $r_t = \mathbb{1}\left[\|s_{t+1} - g\|_2 \leq \epsilon\right]$.

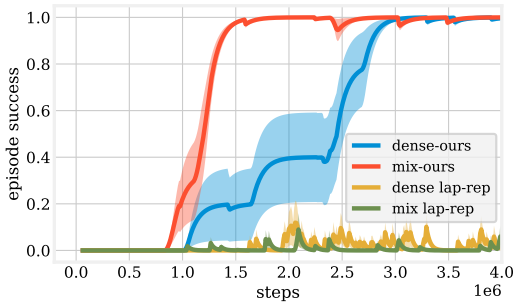


Figure 7. Results of reward shaping using learned representations: each curve and the corresponding confidence intervals are gathered from 5 different runs and then exponentially smoothed (0.9) for better visualization.

The sparsity of the reward poses challenge on the quality of the learned policies as no additional signal could guide the agent towards the goal. The results, illustrated in Figure 8, indicate that even in the non-uniform prior setting, skills trained by our method could quickly assist to complete the task while the options learned with DCO could not. This could suggest that DCO options, in order to succeed, require a richer signal like the distance based dense reward used by Jinnai et al. (2020).

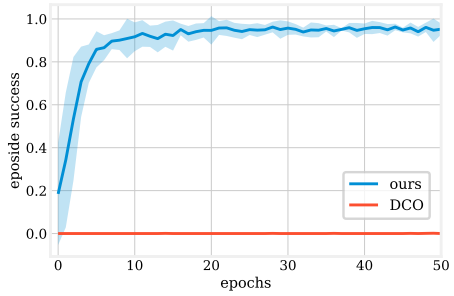


Figure 8. Skill Quality Experiment: each curve and the corresponding confidence intervals are gathered from 5 independent runs and then exponentially smoothed (0.9) for better visualization.

5. Related Work

The main contribution of our work is a representation learning framework for the **reward-agnostic** RL setting. The potential of spectral representations in capturing functional information about the environments, has motivated several works (e.g., Mahadevan, 2005; Machado et al., 2017) to use them in this context. These are powerful tools that proved to scale beyond the tabular case (Machado et al., 2017) to the continuous one (Wu et al., 2019; Jinnai et al., 2020). While these previous works proved to learn useful representations, they often overlook the challenging exploration problem that emerges when collecting the representation training data. In contrast, our framework explicitly couples this challenge with the representation learning objective.

Our work draws inspiration from the self-supervised representation learning literature, more specifically from the idea of contrastive losses (Bromley et al., 1994; Chopra et al., 2005). As mentioned in Section 2, these losses have, conveniently enough, a nice interpretation in the RL setting. Indeed, they correspond to the objective of learning slow features (Bengio et al., 2013; Wiskott & Sejnowski, 2002) that were recently proved to be relevant for exploration (Li et al., 2021). In our work, we show how such a representation objective can be augmented with skill-based knowledge to penalize the agent's boredom (Schmidhuber, 1991; Oudeyer et al., 2007; Oudeyer & Kaplan, 2009) and encourage exploration. The idea of using skills to foster curiosity has also been investigated by Bougie & Ichise (2020). The same proposed augmentation proves to be useful in enforcing the representation's dynamics awareness

For our method's exploratory component, we adopted a hierarchical policy. This has actually been the default setting to model temporally extended strategies, termed options or skills (Sutton et al., 1999). This work shares the same motivation as in Vezhnevets et al. (2017) for training skills to follow latent directions. Among the large body of work on skill discovery, the eigenoptions framework proposed by Machado et al. (2017) and the extensions that followed such as (Machado et al., 2018; Jinnai et al., 2020) are probably closest to our skill training scheme. These eigenoptions also fit in the directional skills definition as they are trained to travel along the directions defined by the Laplacian eigenvectors in some given representation space (of the state space dimensionality). To contrast, we propose to train directional skills defined by arbitrarily diverse set of directions in our *learned* representation space (of small dimensionality).

6. Conclusion

In this work, we conciliate reward-agnostic representation learning with exploration. Focusing on temporal contrast-based methods, we tackle their need for a state space ($\mathcal{S}$) covering strategy and address it away from unrealistic assumptions (uniform prior over $\mathcal{S}$). Our approach leverages the practical skills' training that such representations allow, and uses the learned skills to better cover the state space and learn better representations. We validate our method in tabular as well as continuous environments, and show that even with a concentrated initial state distribution, the induced progressive discovery of the environment provides a suitable covering for the representation objective. The learned representation proved to enjoy a comparable representational power to the one acquired from a uniform prior. Thus, with these results, we hope to bring such representations' applicability one step closer to realistic RL contexts.

References

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. *NeurIPS'2019, arXiv:1906.00910*, 2019.

Banijamali, E., Shu, R., Bui, H., Ghodsi, A., et al. Robust locally-linear controllable embedding. In *International Conference on Artificial Intelligence and Statistics*, pp. 1751–1759, 2018.

Bellemare, M. G., Candido, S., Castro, P. S., Gong, J., Machado, M. C., Moitra, S., Ponda, S. S., and Wang, Z. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.

Bengio, E., Thomas, V., Pineau, J., Precup, D., and Bengio, Y. Independently controllable features. *arXiv preprint arXiv:1703.07718*, 2017.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, August 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.50. URL https://doi.org/10.1109/TPAMI.2013.50.

Bougie, N. and Ichise, R. Skill-based curiosity for intrinsically motivated reinforcement learning. *Machine Learning*, 109(3):493–512, 2020.

Bromley, J., Guyon, I., and LeCun, Y. Signature verification using a" siamese" time delay neural network. 1994.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Choi, J., Guo, Y., Moczulski, M., Oh, J., Wu, N., Norouzi, M., and Lee, H. Contingency-aware exploration in reinforcement learning. *7th International Conference on Learning Representations, ICLR 2019*, pp. 1–19, 2019.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 539–546. IEEE, 2005.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870. PMLR, 2018.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., and Davidson, J. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pp. 2555–2565, 2019b.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Hinton, G., Srivastava, N., and Swersky, K. Neural networks for machine learning lecture 6d - a separate, adaptive learning rate for each connection.

Jinnai, Y., Park, J. W., Machado, M. C., and Konidaris, G. Exploration in reinforcement learning with deep covering options. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeIyaVtwB.

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

Li, S., Zheng, L., Wang, J., and Zhang, C. Learning subgoal representations with slow dynamics. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=wxRwhSdORKG.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Machado, M. C. *Efficient Exploration in Reinforcement Learning through Time-Based Representations*. PhD thesis, University of Alberta, 2019.

Machado, M. C., Bellemare, M. G., and Bowling, M. A laplacian framework for option discovery in reinforcement learning. *34th International Conference on Machine Learning, ICML 2017*, 5:3567–3582, 2017.

Machado, M. C., Rosenbaum, C., Guo, X., Liu, M., Tesauro, G., and Campbell, M. Eigenoption discovery through the deep successor representation. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*, pp. 1–22, 2018.

Mahadevan, S. Proto-value functions: Developmental reinforcement learning. *ICML 2005 - Proceedings of the 22nd International Conference on Machine Learning*, pp. 553–560, 2005. doi: 10.1145/1102351.1102421.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Nachum, O., Gu, S., Lee, H., and Levine, S. Near-optimal representation learning for hierarchical reinforcement learning. *7th International Conference on Learning Representations, ICLR 2019*, pp. 1–15, 2019a.

Nachum, O., Tang, H., Lu, X., Gu, S., Lee, H., and Levine, S. Why does hierarchy (sometimes) work so well in reinforcement learning? *arXiv preprint arXiv:1909.10618*, 2019b.

Nasiriany, S., Pong, V., Lin, S., and Levine, S. Planning with goal-conditioned policies. In *Advances in Neural Information Processing Systems*, pp. 14814–14825, 2019.

Oudeyer, P.-Y. and Kaplan, F. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6, 2009.

Oudeyer, P.-Y., Kaplan, F., and Hafner, V. V. Intrinsic motivation systems for autonomous mental development. *IEEE transactions on evolutionary computation*, 11(2): 265–286, 2007.

Schmidhuber, J. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats*, pp. 222–227, Cambridge, MA, USA, 1991. MIT Press. ISBN 0262631385.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839): 604–609, 2020.

Sutton, R. S., Precup, D., and Singh, S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112 (1):181–211, 1999. ISSN 00043702. doi: 10.1016/S0004-3702(99)00052-1.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. Feudal networks for hierarchical reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3540–3549. JMLR. org, 2017.

Watter, M., Springenberg, J., Boedecker, J., and Riedmiller, M. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pp. 2746–2754, 2015.

Wiskott, L. and Sejnowski, T. J. Slow feature analysis: Unsupervised learning of invariances. *Neural computation*, 14(4):715–770, 2002.

Wu, Y., Tucker, G., and Nachum, O. The Laplacian in RL: Learning representations with efficient approximations, 2019.

A. Representation Objective Augmentation: Ablation Study

A.1. Boredom augmentation helps exploration

In order to illustrate the importance of the proposed augmentation – with the boredom term $\mathcal{B}$ – in the objective 6, we conducted the same representation learning experiments for the three gridworld domains in the non-uniform prior setting, but this time with the non-augmented representation learning objective ($\beta' = 0$).

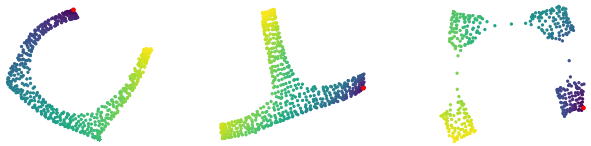
Figure 9 shows how the agent failed at exploring the whole domain. In T-MAZE, it focused only on one corridor without getting curious about the other one. Regarding U-MAZE and 4-ROOMS, the agent stops exploring after discovering the end of the first corridor and the second room respectively. This is due to the lack of incentive to visit the yet unseen states, as they are less rewarding for π_{hi} (i.e. closer in the representation space, hence smaller R^{hi}) than the furthest explored state. The effect of the proposed augmentation would compress the representation of the explored area, say the first corridor in U-MAZE, which makes the rest of the environment more appealing to explore for π_{hi} (i.e. relatively further in the representation space, hence larger R^{hi}). This emphasizes the importance of the boredom term in inducing the agent's exploratory behavior.

(a) U-MAZE (b) T-MAZE (c) 4-ROOMS

Figure 9. Representations learned in the gridworld domains with the *non-augmented* representation objective. Without the boredom term augmentation, the agent fails to cover the state space (cf. Figure 3), and settles for incomplete representations. Colors reflect the distance in terms of the dynamics from the fixed initial state s_0 shown in red.

A.2. Boredom augmentation enforces dynamics-awareness

To verify the benefit of the boredom term beyond helping exploration, we train the representation with the non-augmented objective ($\beta' = 0$) but this time in the uniform prior setting, so that to marginalize the exploration problem. Figure 10 illustrates the learned representations in the three gridworld domains. These representations have failed to capture the dynamics. For example, in the case of 4-ROOMS, the distances from the first room to the fourth and third rooms are comparable in the representation space, which indicates that the representation does not take into account the relative order in which the rooms should be visited, when moving from the first room to the last. Similarly, in U-MAZE, the end of the maze is closer to the initial area than the second corner is. However, in order to reach the former on must pass by the latter. This proves that the boredom term is not only important for the desired exploratory behavior (cf. Figure 9), but also enhances the dynamics-awareness of our representation.



(a) U-Maze (b) T-Maze (c) 4-rooms

Figure 10. Representations learned when uniformly sampling over the state space. Without the boredom term, the representation objective does not provide a dynamics-awareness, Colors reflect the distance in terms of the dynamics from the fixed initial state s_0 shown in red.

B. Implementation details

B.1. GridWorld

For all the experiments, we defined the representation network as an MLP of two hidden layers of size 128 and tanh activations and a linear output layer of the size of representation's dimensionality d. The high-level and the low-level policies are both MLPs of two hidden layers of size 128 with tanh activations and a logsoftmax output layer of the size of their respective action spaces: the environment's 4 actions for the low-level policy and 8 actions for the high-level policy corresponding to the 8 directions $\Omega = \{(\cos(2k\pi/n), \sin(2k\pi/n)) \mid k \in \{0, ..., 7\}\}$ that define diverse skills.

The policies were trained with vanilla A2C with MC returns from the collected trajectories (Monte-Carlo estimates), i.e. no bootstrapped values where used. The skills being of a fixed size they could be trained without any reward discount ($\gamma = 1$). The high-level and low-level policies were entropy-regularized with the coefficients 0.3 and 0.1 respectively.

All of these networks were trained with RMSprop (Hinton et al.) and a learning rate of 0.001. Environments specific hyperparameters are provided below.

B.1.1. REPRESENTATION LEARNING

U-MAZE. Our representation is learned in the non-uniform prior setting with p_{reset}=0.3, p_{rw}=0.4 and K=90

(around the number of steps between s_0 and the furthest state in the maze). We learn a 2-dimensional representation ($d = 2$) using the representation learning objective 6 with $\beta = 0.2$ and $\beta' = 2$. We fix the skills length to $c = 30$ steps (so $L = K/c = 3$), and jointly train the representation ϕ and the policies $(\pi_{\text{hi}}, \pi_{\text{low}})$ by collecting, for each update, a batch of $N = 32$ trajectories of length c to fill D_s and D_{rw} as described in Algorithm 1. We train them for 700 epochs where each epoch corresponds to 10 updates (convergence to the complete representation required around 500 epochs).

T-MAZE. Our representation is learned in the non-uniform prior setting with p_{reset}=0.2, p_{rw}=0.4 and K=40 (around the number of steps between s_0 and the furthest state in the maze). We learn a 2-dimensional representation ($d = 2$) using the representation learning objective 6 with $\beta = 0.2$ and $\beta' = 2$. We fix the skills' length to $c = 20$ steps (so $L = K/c = 2$). and jointly train the representation ϕ and the policies $(\pi_{\text{hi}}, \pi_{\text{low}})$ by collecting, for each update, a batch of $N = 48$ trajectories of length c to fill D_s and D_{rw} as described in Algorithm 1. We train them for 700 epochs where each epoch corresponds to 10 updates (convergence to the complete representation required around 350 epochs).

4-ROOMS. Our representation is learned in the non-uniform prior setting with p_{reset}=0.25, p_{rw}=0.5 and K=60 (around the number of steps between s_0 and the furthest state in the maze). We learn a 2-dimensional representation ($d = 2$) using the representation learning objective 6 with $\beta = 0.2$ and $\beta' = 2$. We fix the skills' length to $c = 20$ steps (so $L = K/c = 3$). and jointly train the representation ϕ and the policies $(\pi_{\text{hi}}, \pi_{\text{low}})$ by collecting, for each update, a batch of $N = 32$ trajectories of length c to fill D_s and D_{rw} as described in Algorithm 1. We train them for 700 epochs where each epoch corresponds to 10 updates (convergence to the complete representation required around 350 epochs).

Note that to keep a clear distinction between the uniform prior and the non-uniform prior settings, we do not allow p_{reset} to be arbitrary small.

The Laplacian representation (Lap-rep) was trained in the same environments' settings described above, for both the uniform and non-uniform prior settings (of course no policy is trained here so $p_{rw} = 1$, and (s_0, p_{reset}) are not relevant for the uniform prior setting). We used the representation learning objective and the associated hyperparameters proposed by Wu et al. (2019). For the uniform prior setting, our online data collection does not cause any discrepancy compared to the offline scheme used in Wu et al. (2019). Indeed, for a minibatch size large enough, the stochastic minibatch based training of Lap-rep when using a **uniform** prior is agnostic to the data collection sheme (offline vs online) since in both cases the minibatches are sampled from the exact same uniform distribution over the state space.

B.1.2. LINEAR FUNCTION APPROXIMATION AND CONTROL

In the Linear Function Approximation (LFA) and control experiments, we evaluate each pretrained representation by training an actor-critic agent to solve a goal-achieving task with a sparse reward ($r = 1$ upon reaching the goal). The episode size was set to 100 steps for all the gridworld domains.

For the LFA, the critic head is a linear function in the given representation, while the actor is a MLP with two hidden layers of size 64 and tanh activations, a logsoftmax output layer of size 4 (discrete gridworld actions) and the actor's input is the state one-hot code. For the control experiments, the actor-critic agent is defined on top of the representation as a MLP of two hidden layers of size 64 with tanh activations that feed two output heads: a linear critic head and a logsoftmax action head for the 4 actions. The agent is trained with A2C with MC returns and a discount of $gamma = 0.98$, a batchsize of 80 episodes, an entropy regularization with a 0.01 coefficient and Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.001.

B.2. MuJoCo: AntMaze

In this navigation task, the environment is composed of $4 \times 4 \times 4$ blocks defining a U-shaped corridor. The environment's action space is 8 dimensional. For the sake of simplifying the RL training algorithm[2], we mapped each dimension values interval to a discrete set of 5 values equally spaced over the interval. We used the same architectures for the representation and the policies as for the gridworld, with the only difference that for the low-level policy, the action head was adapted to the discretization of the action space by having 8 logsoftmax output heads of size 5, one for each action dimension and the corresponding 5 discrete values. This choice makes the training algorithm simpler as it allows using A2C here as well.

Our representation is learned in the non-uniform prior setting with $p_{reset} = 0.2$, $p_{rw} = 0.3$ and $K = 500$. We learn a 2-dimensional representation ($d = 2$) using the representation learning objective 6 with $\beta = 0.2$ and $\beta' = 5$. We fixed their length to $c = 100$ steps (so $L = K/c = 5$). and jointly train the representation ϕ and the policies $(\pi_{\text{hi}}, \pi_{\text{low}})$ by collecting, for each update, a batch of $N = 32$ trajectories of length c to fill D_s and D_{rw} as described in Algorithm 1. We train them for 1000 epochs where each epoch corresponds to 10 updates (convergence to the complete representation required around 650 epochs).

The policies were trained with the same A2C used in

[2] orthogonal to our contributions.

gridworld domains and the same RMSprop hyperparameters. The high-level and low-level policies were entropy-regularized with the coefficients 0.15 and 0.1 respectively.

B.2.1. Reward Shaping

Regarding the Laplacian representation baseline, Lap-rep was learned in the same non-uniform prior setting described above, with the representation objective and its associated hyperparameters proposed by Wu et al. (2019). In this setting, the data collection and the representation training are performed simultaneously in an online fashion. We have also tested the offline representation training, replicating the training scheme in Wu et al. (2019). Still in the non-uniform prior setting, we collected 500000 training samples (10 times more than in (Wu et al., 2019)) according to a uniformly random policy, then we trained the representation on the large dataset built this way. For all other hyperparameters, we used the same as provided in (Wu et al., 2019). Both trainings ended up giving the same performance for the reward shaping task.

Now, for the reward shaping, we train a Soft Actor-Critic (SAC) (Haarnoja et al., 2018) agent to reach a goal area (neighbourhood around the goal position) with episodes of size 1000 steps. We use the following hyperparameters:

- Discount $\gamma = 0.99$
- Entropy coefficient (temperature) $\alpha = 0.1$
- Soft critic updates with smoothing constant $\tau = 0.005$
- Replay buffer of size $5 \cdot 10^6$ (equal to the number of training steps).
- Adam optimize with learning rate of 0.0001

As SAC is sensitive to the reward scale (Haarnoja et al., 2018), we grid-searched this hyperparameter in $\{10^{-5}, 10^{-4}, \cdots, 1.0, 2.0\}$, and the best performing one for our representation was 1.0, while for Lap-rep the SAC agent didn't succeed with any of these values to solve the task.

B.2.2. Skills Evaluation

To train the *Deep Covering Options* (DCO), we first collect a dataset to estimate the second eigenvector and then use the same dataset to train a policy – the option – using DDPG (Lillicrap et al., 2015). Each DCO option is tied to its own eigenvector estimate and its own dataset. Each dataset is of size 500000 (10 times the size used in Jinnai et al. (2020)). As suggested by the authors of DCO (Jinnai et al., 2020), the remaining the hyperparameters to estimate the eigenvectors and train their corresponding options were taken from Wu et al. (2019). For fair comparison, we train 8 DCO options as well.

For the skills / options evaluation stage, we freeze the learned low-level policies and train a high-level policy to use the 8 skills as the only available actions to reach the goal g on the other end of the AntMaze environment using a **sparse** reward $r_t = \mathbb{1}\left[\|s_{t+1} - g\|_2 \leq \epsilon\right]$ within a finite horizon of 1000 steps . Note that this tasks is quite challenging given the type of reward and the length of episode especially in a continuous state space. As our skills offer some flexibility in their execution (can be started everywhere and run for arbitrary number of steps), this episode length was decomposed to 5 skills of 200 steps each. The high-level policy was trained with A2C with MC returns (no discount given the finite horizon) a batch size of 8 episodes, and RMSprop optimizer with a learning rate of 0.001.

C. The switching utility of the augmentation term

Note that $\mathcal{D}_s$ may contain trajectories from skills that are not yet duly trained (for example early in the training or in a freshly discovered area). Since at that stage, these skills' trajectories are close to random walks, their contribution in the boredom term $\beta' \mathcal{B}(\phi; \mathcal{D}_s)$ is similar to the first attractive term which is based on random walks. This means that a new skill trajectory initially contributes to the similarity term (attractive term) in training the representation, thus making the most out of the sampled skills' trajectories while these are still early in their training. This computationally improves the representation learning for it uses all the relevant trajectories to train the representation and not only those collected from the uniformly random policy μ. The more a skill is trained, the more structured its trajectories become and the more they contribute to the intended "boredom" effect, that is encouraging exploration and dynamics awareness (see Appendix A).