Towards Better Laplacian Representation in Reinforcement Learning with Generalized Graph Drawing

Kaixin Wang [*1] **Kuangqi Zhou** [*1] **Qixin Zhang** [2] **Jie Shao** [3] **Bryan Hooi** [1] **Jiashi Feng** [1]

Abstract

The Laplacian representation recently gains increasing attention for reinforcement learning as it provides succinct and informative representation for states, by taking the eigenvectors of the Laplacian matrix of the state-transition graph as state embeddings. Such representation captures the geometry of the underlying state space and is beneficial to RL tasks such as option discovery and reward shaping. To approximate the Laplacian representation in large (or even continuous) state spaces, recent works propose to minimize a spectral graph drawing objective, which however has *infinitely many* global minimizers other than the eigenvectors. As a result, their learned Laplacian representation may differ from the ground truth. To solve this problem, we reformulate the graph drawing objective into a generalized form and derive a new learning objective, which is proved to have eigenvectors as its *unique* global minimizer. It enables learning high-quality Laplacian representations that faithfully approximate the ground truth. We validate this via comprehensive experiments on a set of gridworld and continuous control environments. Moreover, we show that our learned Laplacian representations lead to more exploratory options and better reward shaping.

1. Introduction

Reinforcement learning (RL) aims to train an agent that can take proper sequential actions based on the perceived states from the environments (Sutton & Barto, 2018). Thus the quality of state representations is important to the agent performance by benefiting its generalization ability (Zhang et al., 2018; Stooke et al., 2020; Agarwal et al., 2021), ex-

---
[*]Equal contribution [1]National University of Singapore [2]City University of Hong Kong [3]ByteDance AI lab. Correspondence to: Kaixin Wang <kaixin.wang@u.nus.edu>, Kuangqi Zhou <kzhou@u.nus.edu>.
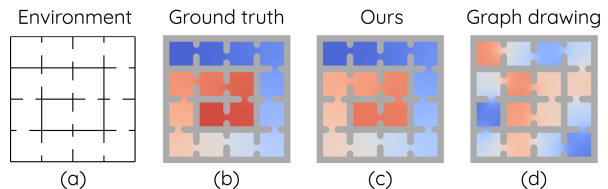
Figure 1. Visualization of environment and Laplacian state representations (2nd dimension). (a) Top view of a continuous control navigation environment. (b) Ground-truth Laplacian representation. It encodes geometry of the environment: nearby states have similar values while distant states have dissimilar values. (c) Our learned representation. It can be seen that our result is very similar to ground truth. (d) Representation learned by spectral graph drawing (Wu et al., 2019). It significantly diverges from ground truth and fails to capture geometric information about the state space. Best viewed in color.

ploration ability (Pathak et al., 2017; Machado et al., 2017; 2020), and learning efficiency (Dubey et al., 2018; Wu et al., 2019), *etc*. Recently, the Laplacian representation receives increasing attention (Mahadevan, 2005; Machado et al., 2017; Wu et al., 2019; Jinnai et al., 2019). It views the states and transitions in an RL environment as nodes and edges in a graph, and forms a d-dimension state representation with d smallest eigenvectors of the graph Laplacian. Such representations can capture the geometry of the underlying state space, as illustrated in Fig. 1(b), which greatly benefits RL in option discovery (Machado et al., 2017; Jinnai et al., 2019) and reward shaping (Wu et al., 2019).

However, to compute the exact Laplacian representation is very challenging, as directly computing eigendecomposition of the graph Laplacian requires access to the environment transition dynamics and involves expensive matrix operation. Hence it is largely limited to small finite state spaces. To deal with large (or even continuous) state spaces, previous works resort to approximation methods (Machado et al., 2017; 2018; Wu et al., 2019). A most efficient one is (Wu et al., 2019), which minimizes a spectral graph drawing objective (Koren, 2005). However, this objective has infinitely many other global minimizers besides the ground truth Laplacian representation (*i.e.*, d smallest eigenvectors), as it is invariant to an arbitrary orthogonal transformation over the optimization variables. The resulted representa-

tions can then correspond to other minimizers and diverge from the ground truth, thus unable to encode the geometry of the state space as expected (see Fig. 1(d)).

To break such invariance and approximate Laplacian representation closer to the ground-truth, we reformulate the graph drawing objective into a generalized form by introducing coefficients for each term in it. By assigning decreasing values for these coefficients, we can derive a training objective that breaks the undesired invariance. We provide theoretical guarantees that the proposed objective has the d smallest eigenvectors as its *unique* global minimizer under mild assumptions. As shown in Fig. 1(c), minimizing the new objective is able to ensure faithful approximation to the ground truth Laplacian representation.

To verify the effectiveness of our method for learning high-quality Laplacian representations, we conduct experiments in gridworld and continuous control environments. We show that the learned representations by our method more accurately approximate the ground truth, compared with the ones from the graph drawing objective. Furthermore, we apply the learned representations to two downstream RL tasks. It is demonstrated that, for option discovery task (Machado et al., 2017) our method leads to discovered options that are more exploratory compared to using the representation learned by graph drawing; in reward shaping task (Wu et al., 2019), our learned representation is better at accelerating agents' learning than previous work (Wu et al., 2019)..

The rest of the paper is organized as follows. In Sec. 2, we introduce background about RL and Laplacian representation in RL. In Sec. 3, we propose our new objective for learning Laplacian representation. Then, we conduct experiments to demonstrate that our proposed objective is able to learn high-quality representations in Sec. 4. In Sec. 5 we review related works and Sec. 6 concludes the paper.

2. Background

2.1. Reinforcement Learning

In the RL framework (Sutton & Barto, 2018), an agent interacts with an environment by observing states and taking actions, with an aim of maximizing cumulative reward. We consider Markov Decision Process (MDP) formalism in this paper. An MDP can be described by a 5-tuple $\langle \mathcal{S}, \mathcal{A}, r, p, \gamma \rangle$. Specifically, at time t the agent observes state $s_t \in \mathcal{S}$ and takes an action $a_t \in \mathcal{A}$. The environment then yields a reward signal R_t sampled from the reward function $r(s_t, a_t)$. The state observation in the next timestep $s_{t+1} \in \mathcal{S}$ is sampled according to an environment-specific transition distribution function $p(s_{t+1}|s_t, a_t)$. A policy is defined as a mapping $\pi : \mathcal{S} \to \mathcal{A}$ that returns an action a given a state s. The goal of the agent is to learn an optimal policy π^* that

maximizes the expected cumulative reward:

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{p,\pi} \sum_{t=0}^{\infty} \gamma^t R_t, \tag{1}$$

where Π denotes the policy space and $\gamma \in [0, 1)$ is a discount factor.

2.2. Laplacian Representation in RL

By considering states and transitions in an MDP as nodes and edges in a graph, the Laplacian state representation is formed by the d smallest eigenvectors of the graph Laplacian. Specifically, each eigenvector (of length $|\mathcal{S}|$) corresponds to a dimension of the Laplacian representation for all states. Formally, we denote the graph as $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ where $\mathcal{E}$ is the edge set consisting of transitions between states. The graph Laplacian of $\mathcal{G}$ is defined as $L = D - A$, where $A \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ is the adjacency matrix of $\mathcal{G}$, and $D = \text{diag}(A\mathbf{1})$ is the degree matrix (Chung & Graham, 1997). We denote the i-th smallest eigenvalue of L as λ_i, and the corresponding unit eigenvector as $\mathbf{e}_i \in \mathbb{R}^{|\mathcal{S}|}$. The d-dimensional Laplacian representation of a state s is $\varphi(s) = (\mathbf{e}_1[s], \cdots, \mathbf{e}_d[s])$, where $\mathbf{e}_i[s]$ denotes the entry in vector $\mathbf{e}_i$ that corresponds to state s. In particular, $\mathbf{e}_1$ is a normalized all-ones vector and has the same value for all s.

The Laplacian representation is known to be able to capture the geometry of the underlying state space (Mahadevan, 2005; Machado et al., 2017), and thus has been applied in option discovery (Machado et al., 2017; Jinnai et al., 2019) and reward shaping (Wu et al., 2019).

In the Laplacian framework for option discovery (Machado et al., 2017), each dimension of the Laplacian representation defines an intrinsic reward function $r_i(s, s') = \mathbf{e}_i[s] - \mathbf{e}_i[s']$. The options (Sutton et al., 1999) are discovered by maximizing the cumulative discounted intrinsic reward. These options act at different time scales; that is, when an agent follows an option to take actions, the length of its trajectory until termination varies across different options (see Fig. 3 in (Machado et al., 2017)). Such a property makes these options helpful for exploration: longer options enable agents to quickly reach distant areas and shorter options ensure sufficient exploration in local areas.

When using the Laplacian representation for reward shaping in goal-achieving tasks (Wu et al., 2019), the reward is shaped based on Euclidean distance, as in (Pong* et al., 2018; Nachum et al., 2018). Specifically, the pseudo-reward is defined as the negative ℓ_2 distance between the agent's state and the goal state in representation space: $r_t = -\|\varphi(s_{t+1}) - \varphi(s_{\text{goal}})\|_2$. Since the Laplacian representation can reflect the geometry of the environment dynamics, such pseudo-reward can be helpful in accelerating the learning process.

2.3. Approximating Laplacian Representation

Obtaining Laplacian representation by directly computing eigendecomposition of graph Laplacian requires access to transition dynamics of the environment and involves expensive matrix operations, which is infeasible for environments with a large or even continuous state space. One efficient approach for approximating Laplacian representation is proposed by Wu et al. (2019), which minimizes the following spectral graph drawing objective (Koren, 2005):

$$\min_{\mathbf{u}_1,\cdots,\mathbf{u}_d} \quad \sum_{i=1}^{d} \mathbf{u}_i^\top L \mathbf{u}_i \tag{2}$$
$$\text{s.t.} \quad \mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}, \forall i, j = 1, \cdots, d,$$

where $(\mathbf{u}_1, \cdots, \mathbf{u}_d) \in \mathbb{R}^{|\mathcal{S}| \times d}$ are to approximate the eigenvectors $(\mathbf{e}_1, \cdots, \mathbf{e}_d)$, and δ_{ij} is the Kronecker delta. However, minimizing such an objective can only ensure that $\mathbf{u}_1, \cdots, \mathbf{u}_d$ span the same subspace as $\mathbf{e}_1, \cdots, \mathbf{e}_d$, as mentioned in (Wu et al., 2019). It does not guarantee $\mathbf{u}_i = \mathbf{e}_i$ for $i = 1, \cdots, d$, because the global minimizer $(\mathbf{e}_1, \cdots, \mathbf{e}_d)$ is not unique.

Transforming $(\mathbf{e}_1, \cdots, \mathbf{e}_d)$ with an arbitrary orthogonal transformation also achieves global minimum (Koren, 2005). Therefore, the problem in Eqn. (2) does not ensure that the solution is the eigenvectors, and may converge to any other global minima. Accordingly, the learned Laplacian representations may diverge from the ground truth. We will show in Sec. 4 that such representation is less helpful in discovering exploratory options and reward shaping.

3. Method

As discussed in Sec. 2.3, the graph drawing objective is invariant under orthogonal transformation: applying an arbitrary orthogonal transformation to the d smallest eigenvectors also yields a global minimizer, which hinders learning Laplacian representations close to the ground truth.

We then consider breaking such invariance for achieving more accurate approximation. To this end, we reformulate the graph drawing objective into a weighted-sum form, yielding the following generalized graph drawing objective:

$$\min_{\mathbf{u}_1,\cdots,\mathbf{u}_d} \quad \sum_{i=1}^{d} c_i \mathbf{u}_i^\top L \mathbf{u}_i \tag{3}$$
$$\text{s.t.} \quad \mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}, \forall i, j = 1, \cdots, d,$$

where c_i is the coefficient for the i-th term $\mathbf{u}_i^\top L \mathbf{u}_i$. When $c_i = 1$ for every i, it degenerates to the original graph drawing objective in Eqn. (2).

We find that, under mild assumptions, if $c_1, \cdots, c_d$ are strictly decreasing, then the d smallest eigenvectors (ground-truth Laplacian representation) make the *unique* global minimizer of the above generalized graph drawing objective, as stated in the following theorem.

Theorem 1. *Assume* $\mathbf{u}_i \in \text{span}(\{\mathbf{e}_1, \cdots, \mathbf{e}_d\})$, $\forall i = 1, \cdots, d$, *and* $\lambda_1 < \cdots < \lambda_d$. *Then,* $c_1 > \cdots > c_d > 0$ *is a sufficient condition for the generalized graph drawing objective to have a unique global minimizer* $(\mathbf{u}_1^*, \cdots, \mathbf{u}_d^*) = (\mathbf{e}_1, \cdots, \mathbf{e}_d)$, *and the corresponding minimum is* $\sum_{i=1}^{d} c_i \lambda_i$.

Proof. Here we give the proof sketch of Theorem 1 and the full proof is deferred to the Appendix. Denote the objective of problem (3) as $h(\mathbf{u}_1, \cdots, \mathbf{u}_d)$. Let $g = h(\mathbf{u}_1, \cdots, \mathbf{u}_d) - \sum_{i=1}^{d} c_i \lambda_i$. Since $\mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}$ and $\mathbf{u}_i \in \text{span}(\{\mathbf{e}_1, \cdots, \mathbf{e}_d\})$, without loss of generality, $(\mathbf{u}_1, \cdots, \mathbf{u}_d)$ can be written as $(\mathbf{e}_1, \cdots, \mathbf{e}_d)Q$, where $Q = (q_{ij})$ is an orthogonal matrix.

We first prove optimality. By applying Fubini's Theorem (Fubini, 1907) to g, we can rewrite g as $g = \sum_{k=1}^{d-1} s_k (\lambda_{k+1} - \lambda_k)$, where s_k depends on Q and d. We can prove $\forall k \in \{1, \cdots, d-1\}, s_k \geqslant 0$ (this is given by $c_1 > \cdots > c_d > 0$). Hence $g \geqslant 0$. Notice that the inequality is tight when $(\mathbf{u}_1, \cdots, \mathbf{u}_d) = (\mathbf{e}_1, \cdots, \mathbf{e}_d)$, which proves the optimality.

Then, we prove uniqueness by contradiction. Denote the global minimum $\sum_{i=1}^{d} c_i \lambda_i$ as p^*. Assume that there exists another global minimizer, denoted as $(\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_d)$, *i.e.*, $\exists i, \tilde{\mathbf{u}}_i \neq \pm \mathbf{e}_i$. Here we require $\tilde{\mathbf{u}}_i \neq \pm \mathbf{e}_i$ because the sign of $\mathbf{e}_i$ is arbitrary. Again, we rewrite $(\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_d) = (\mathbf{e}_1, \cdots, \mathbf{e}_d)\tilde{Q}$, where $\tilde{Q} = (\tilde{q}_{ij})$ is an orthogonal matrix. Therefore, the assumption is equivalent to $\exists i \in \{1, \cdots, d\}, \tilde{q}_{ii} \notin \{1, -1\}$. Due to the optimality of $(\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_d)$, we know $h(\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_d) = p^*$. However, we can prove that this equality implies $\forall i \in \{1, \cdots, d\}, \tilde{q}_{ii} \in \{1, -1\}$. This contradicts with $\exists i, \tilde{\mathbf{u}}_i \neq \pm \mathbf{e}_i$, and hence proves uniqueness. $\square$

We will empirically show that the two assumptions hold in our experiments (see Sec. 4.4.1). Based on the above theorem, we can choose $c_1 > \cdots > c_d > 0$ to obtain a learning objective that can faithfully approximate the Laplacian representation. A natural choice is $c_d = 1, c_{d-1} = 2, \cdots, c_1 = d$, which gives the following objective:

$$\min_{\mathbf{u}_1,\cdots,\mathbf{u}_d} \quad \sum_{i=1}^{d} (d - i + 1) \mathbf{u}_i^\top L \mathbf{u}_i \tag{4}$$
$$\text{s.t.} \quad \mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}, \forall i, j = 1, \cdots, d.$$

We use the above objective throughout the rest of our paper, and conduct ablative experiments with other choices of the coefficients (see Sec.4.4.2).

Note that Theorem 1 implies a property of the generalized graph drawing objective in Eqn. (3): there is one-to-one correspondence between its solutions and the smallest d eigenvectors, $i.e.$, $\mathbf{u}_i^* = \mathbf{e}_i, \forall i \in \{1, \cdots, d\}$. With this, a specific dimension ($e.g.$, the 2nd dimension) of the Laplacian representation can be easily derived from the corresponding solution ($e.g.$, $\mathbf{u}_2$). This exact correspondence is useful for studying how each dimension of the representation influences an RL task, $e.g.$, reward shaping (see Sec. 4.3). Note that the spectral graph drawing objective does not have such a property.

The above theoretical results can be easily generalized to the function space ($i.e.$, Hilbert space), which corresponds to a continuous state space in RL (see Appendix).

Training objective In RL applications, it is hard to directly optimize the problem (4) because L is not accessible and enumerating the state space may be infeasible. To make the optimization amenable, we follow the practice in (Wu et al., 2019) to express the objective as an expectation. The objective in Eqn. (4) can be rewritten as

$$\sum_{k=1}^{d}\sum_{i=1}^{k} \mathbf{u}_i^\top L \mathbf{u}_i = \sum_{k=1}^{d}\sum_{i=1}^{k}\sum_{(s,s')\in\mathcal{E}} (\mathbf{u}_i[s] - \mathbf{u}_i[s'])^2 \quad (5)$$

where the inner summation of the right hand side is over all edges ($i.e.$ transitions) in the graph, and $\mathbf{u}_i[s]$ denotes the entry of vector $\mathbf{u}_i$ corresponding to state s. In practice, we train a neural network $\phi(s)$ with d-dimension output $[f_1(s), \cdots, f_d(s)]$ to approximate the Laplacian representation for state s. Since we only have sampled transitions, we can express Eqn. (5) as an expectation and minimize the following objective

$$G(f_1, \cdots, f_d) \triangleq \mathbb{E}_{(s,s')\sim\mathcal{T}} \sum_{k=1}^{d}\sum_{i=1}^{k} (f_i(s) - f_i(s'))^2 \tag{6}$$

where (s, s') is a state-transition sampled from a dataset of transitions $\mathcal{T}$.

The orthonormal constraints in Eqn. (4) can be implemented as a penalty term:

$$\mathbb{E}_{s\sim\rho, s'\sim\rho} \sum_{l=1}^{d}\sum_{j=1}^{l}\sum_{k=1}^{l} h_{jk}(s, s') \tag{7}$$

where

$$h_{jk}(s, s') = (f_j(s)f_k(s) - \delta_{jk})(f_j(s')f_k(s') - \delta_{jk}). \tag{8}$$

Here ρ denotes the distribution of states in $\mathcal{T}$. Please refer to (Wu et al., 2019) and the Appendix for detailed derivation.
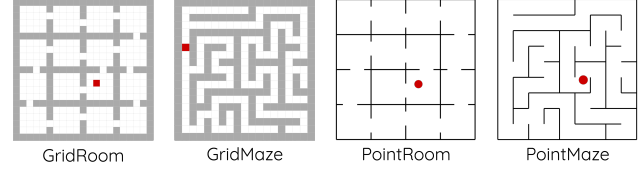

Figure 2. Environments for experiments (agents depicted in red).

4. Experiments

In this section, we conduct extensive experiments to validate the effectiveness of our method in improving learned Laplacian representation. Specifically, in Sec. 4.1, we evaluate the learned representations on how well they approximate the ground truth. In Sec. 4.2 and Sec. 4.3, we evaluate the learned representations on their effectiveness in two downstream tasks, $i.e.$ for discovering exploratory options and improving reward shaping. Finally, in Sec. 4.4, we empirically verify the assumptions used in Theorem 1 and evaluate other coefficient choices.

We use two discrete gridworld environments and two continuous control environments for our experiments (see Fig. 2), following previous work (Wu et al., 2019). The gridworld environments are built with MiniGrid (Chevalier-Boisvert et al., 2018) and the continuous control environments are created with PyBullet (Coumans & Bai, 2016–2019). Note that for gridworld environments, our setting is not tabular, since we approximate the Laplacian representation via training neural networks on raw observations (such as (x, y) positions or top-view images) rather than learning a mapping table for all states. For all experiments, we use $d = 10$ for the dimension of the Laplacian representation. More details about training setup can be found in the Appendix. For clarity, throughout the experiments we use *baseline* to refer to the method of (Wu et al., 2019).

4.1. Learning Laplacian Representations

We take (Wu et al., 2019) as our baseline and following its practice, we also train a neural network to approximate the Laplacian representation, using trajectories collected by a uniformly random policy with random starts. For the environments with discrete state-space (`GridRoom` and `GridMaze`), we conduct experiments with both (x, y) position observation and image observation. The ground-truth Laplacian representations ($i.e.$, eigenvectors) are computed by eigendecomposing the graph Laplacian matrix. For environments with continuous state spaces (`PointRoom` and `PointMaze`), we use (x, y) positions as observations, and the ground-truth representations ($i.e.$, eigenfunctions) are approximated by the finite difference method with 5-point stencil (Peter & Lutz, 2003). Please see the Appendix for more training details.
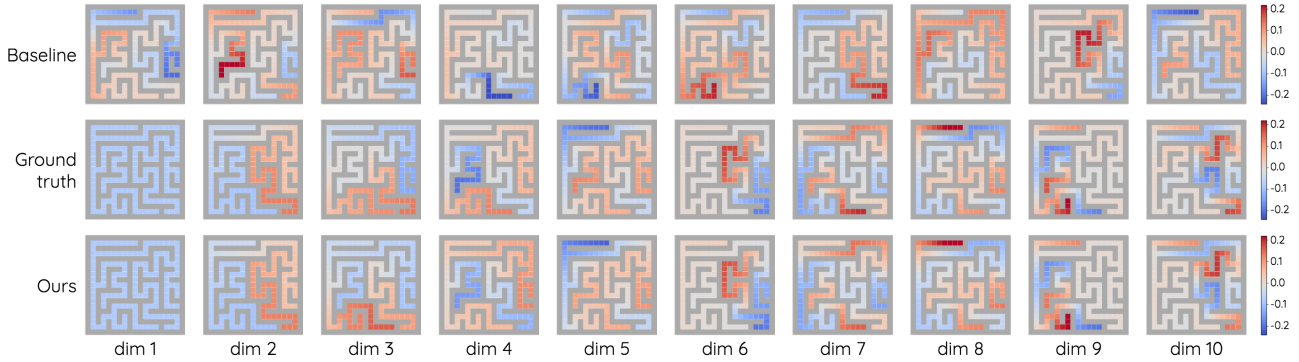
Figure 3. Visualization of the learned 10-dimension Laplacian representation and the ground truth on `GridMaze`. Each heatmap shows a dimension of the representation for all states in the environment, where each state is a single cell. Best viewed in color.
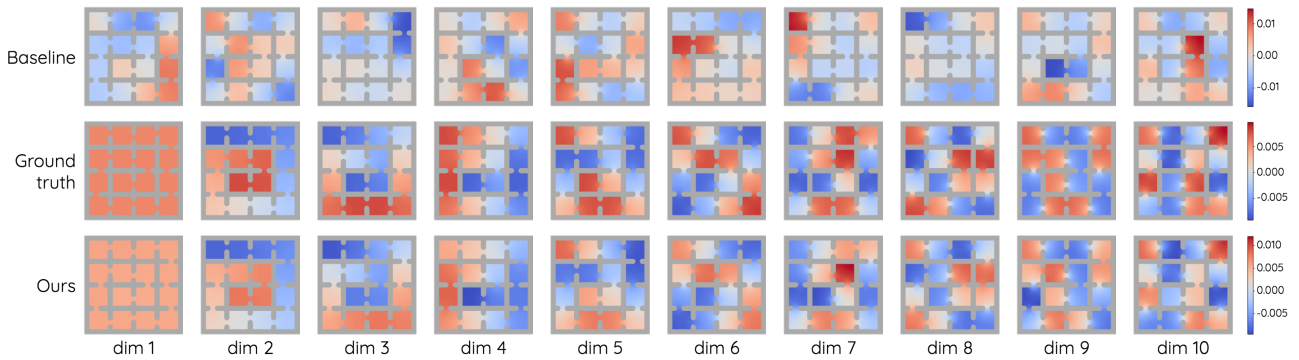

Figure 4. Visualization of the learned 10-dimension Laplacian representations and the ground truth on `PointRoom`. Each heatmap shows a dimension of the representation for all states (via interpolation) in the environment. Best viewed in color.

Table 1. SimGT between learned representation and ground truth, averaged across 3 runs. (image) denotes using image observations.

Environment	GridRoom	GridMaze	GridRoom (image)	GridMaze (image)	PointRoom	PointMaze
Baseline	0.239	0.220	0.310	0.229	0.239	0.255
Ours	0.991	0.962	0.985	0.984	0.963	0.779

To get an intuitive comparison between our method and the baseline in approximating the Laplacian representation, we visualize the learned state representations as well as the ground truth ones of `GridMaze` and `PointRoom` in Fig. 3 and Fig. 4. As the figures show, our learned Laplacian representations approximate the ground truth much more accurately, while the baseline representations significantly diverge from the ground truth. Similar results in other environments are included in the Appendix.

To quantify the approximation quality of the learned representations, we calculate the absolute dimension-wise cosine similarities between the learned representations and ground-truth ones, and take the average over all dimensions, which yields the following SimGT metric:

$$\text{SimGT} = \frac{1}{d} \sum_{i=1}^{d} \left| \sum_{s} f_i(s) \mathbf{e}_i[s] \right|, \qquad (9)$$

where s is a state, $f_i(s)$ is the i-th dimension of the learned representation of s (defined in Sec. 3), and $\mathbf{e}_i[s]$ is the i-th dimension of the ground truth for s (defined in Sec. 2.2) respectively. Note that SimGT is in the range $[0, 1]$, and larger SimGT means that the representation is closer to the ground truth. As shown in Tab. 1, our method achieves much higher SimGT than the baseline, indicating better approximation.

Moreover, we provide empirical evidence for our discussion in Sec. 3, that our method converges to the unique global minimizer, while the baseline method can converge to different minima. To illustrate this, we visualize the learned representations in 3 different runs, and use the following SimRUN metric to measure the variance between the representations learned in l-th run and those learned in the m-th
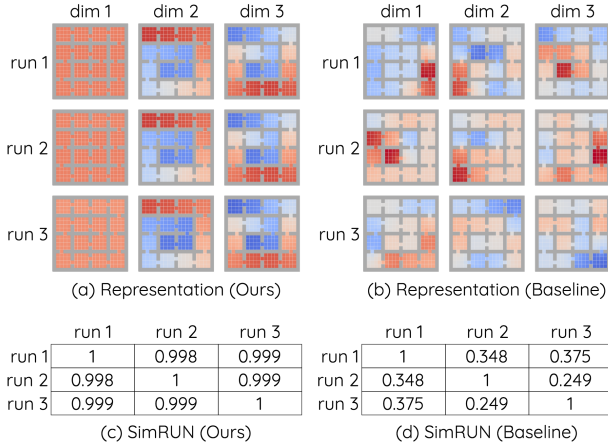
(a) Representation (Ours)

(b) Representation (Baseline)

	run 1	run 2	run 3
run 1	1	0.998	0.999
run 2	0.998	1	0.999
run 3	0.999	0.999	1

(c) SimRUN (Ours)

	run 1	run 2	run 3
run 1	1	0.348	0.375
run 2	0.348	1	0.249
run 3	0.375	0.249	1

(d) SimRUN (Baseline)

Figure 5. (a) and (b): Visualization of first 3 dimensions of representations learned by our method (a) and baseline method (b). (c) and (d): SimRUN(l, m) for $l, m = 1, 2, 3$ computed with our learned representations (c) and those learned by baseline (d).

run ($1 \leqslant l, m \leqslant 3$) via the following:

$$\text{SimRUN}(l, m) = \frac{1}{d} \sum_{i=1}^{d} \left| \sum_s f_i^l(s) f_i^m(s) \right|, \qquad (10)$$

where $f_i^l(s)$ and $f_i^m(s)$ denote the i-th dimension of the learned representation of state s in the l-th and m-th run, respectively. SimRUN(l, m) is in the range $[0, 1]$, and larger value implies larger inconsistency in the learned representations between 2 runs. As Fig. 5 shows, the learned representations of the baseline method vary a lot across runs, indicating convergence to different minima. In contrast, our method yields consistent approximations.

The above results demonstrate the superiority of learning Laplacian representation with our proposed objective and empirically support our theoretical analysis in Sec. 3.

4.2. Option Discovery

As discussed in Sec. 2.2, the Laplacian representation can be applied in discovering exploratory options. We here evaluate effectiveness of our learned representations in discovering exploratory options, to further show the superiority of our method over the baseline.

Following (Machado et al., 2017), we learn 2 options for each dimension of the learned representation: one with an intrinsic reward function $r_i(s, s') = f_i(s') - f_i(s)$ and the other with $-r_i(s, s')$, where $f_i(s)$ denotes i-th dimension of the representation for state s (see Sec. 3). The options are learned with Deep Q-learning (Mnih et al., 2013). Since the first dimension of the Laplacian representation has the same value for every state (see Sec. 2.2), it cannot provide informative intrinsic reward. Therefore, we do not learn options for the first dimension of our learned representation.
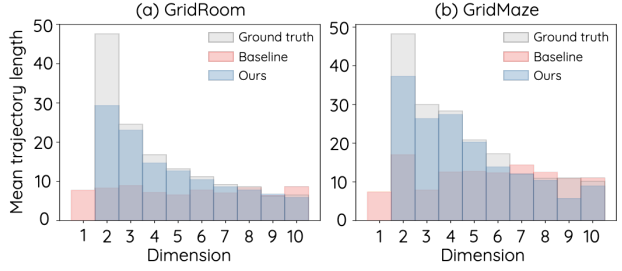

Figure 6. Average length of trajectories for options discovered from each dimension of different representations. Longer length implies the option acts at a longer time scale. For each dimension, the result is averaged from corresponding 2 options.
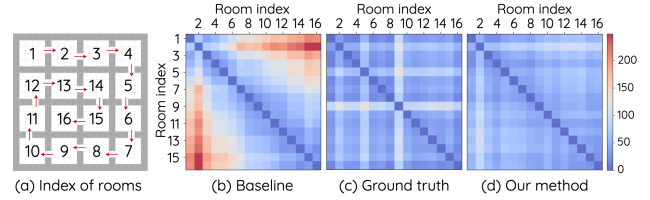

Figure 7. (a) Index of rooms. Neighboring rooms are assigned with consecutive indexes. Larger difference between two indexes means that the corresponding rooms are farther in the environment. (b) Average steps needed to navigate between two rooms (baseline). The agent needs many steps to navigate between distant rooms, *e.g.*, room 1 and room 16. (c) Average steps needed to navigate between two rooms (ground truth). (d) Average steps needed to navigate between two rooms (our method).

For each learned option, we compute the average trajectory length for an agent that starts from each state and follows this option until arriving at termination states. This reflects the time scale at which an option acts: options acting at longer time scales enable agents to quickly reach distant areas, and shorter options ensure sufficient exploration in local areas. As Fig. 6 shows, trajectory lengths for our method vary across different dimensions, implying the options operate at different time scales. Such options enable exploration in both nearby and distant areas. In contrast, options discovered from the baseline representation operate at similarly short time scales, which may hinder exploration to the distant areas.

To further validate this, we measure the expected number of steps required for an agent (equipped with learned options) to navigate between different rooms in the `GridRoom` environment. Specifically, we denote $N_{i \to j}$ as the average number of steps required for an agent starting from room i to reach room j. We then calculate $N_{i,j} = (N_{i \to j} + N_{j \to i})/2$ as the expected number of steps required to navigate between two rooms. As shown in Fig. 4.2, agents equipped with options discovered from the baseline representation typically take more steps to reach distant rooms. In comparison, with our method, agents can reach faraway rooms within a similar number of steps as for nearby rooms.
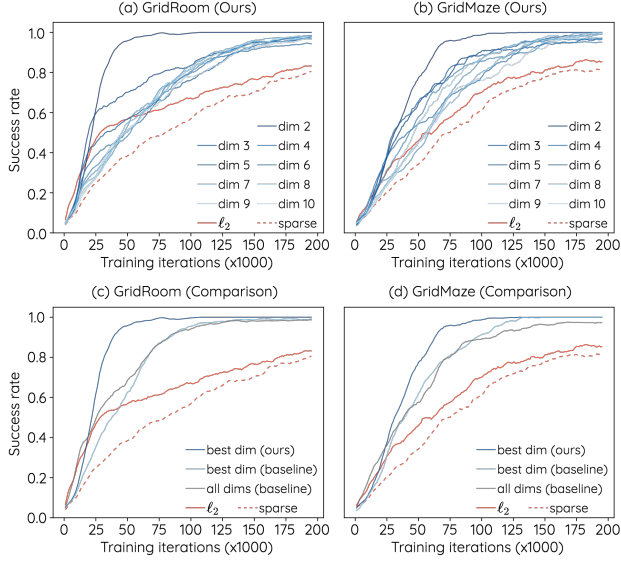
Figure 8. Results of reward shaping with each dimension of learned Laplacian representations. ℓ_2 denotes reward shaping with L2 distance in raw observation space (*i.e.*, (x, y) position), and *sparse* denotes no reward shaping.

Figure 9. Sum of absolute cosine similarities between $\mathbf{u}_i$ and eigenvectors $\mathbf{e}_1, \cdots, \mathbf{e}_d$, during training.

Figure 10. Eigenvalues of the graph Laplacian matrix for two environments with discrete state-space.

4.3. Reward Shaping

The Laplacian representation can be used for reward shaping in goal-achieving tasks, as mentioned in Sec. 2.2. Previous work (Wu et al., 2019) uses all d dimensions of the learned representation to define the pseudo-reward, *i.e.*, $r_t = -\|\phi(s_{t+1}) - \phi(s_{\text{goal}})\|_2$, where $\phi(s)$ is the d-dimension representation of state s output by a neural network. Such pseudo-reward is influenced by each dimension of the representation. As each dimension of Laplacian representation captures different geometric information about the state space (*e.g.*, see Fig. 4), a natural question is: which dimension of the Laplacian representation matters more to reward shaping? Furthermore, can we achieve better reward shaping than using all d dimensions?

In this subsection, we study these questions by comparing individual dimensions of the learned representation for reward shaping. Specifically, we define the pseudo-reward as $r_t = -\left(f_i(s_{t+1}) - f_i(s_{\text{goal}})\right)^2$ where $f_i(s)$ denotes i-th dimension of the representation for state s. Following (Wu et al., 2019), we train the agents using Deep Q-learning (Mnih et al., 2013) with (x, y) positions as observations, and measure the agent's success rate of reaching the goal state. To eliminate the bias brought by the goal position, we select multiple goals for each environment such that they spread over the state space, and average the results of different goals. We do not experiment with the first dimension of our learned representation since every state has same value and the pseudo-reward is always 0.

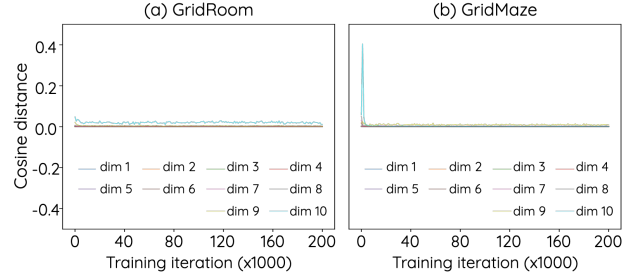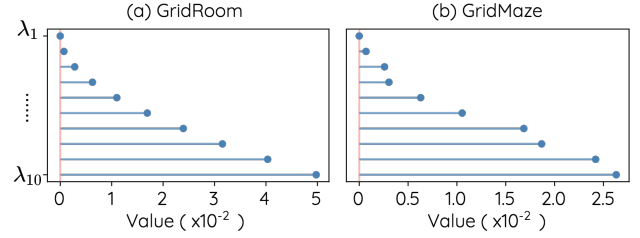As shown in Fig. 8(a) and Fig. 8(b), using lower dimensions

of our learned Laplacian representation for reward shaping better accelerates the agent's learning process. Furthermore, we compare using best dimension and all dimensions, of both our learned representation and the baseline representation, in Fig. 8(c) and Fig. 8(d). Results show that the lower dimension of our representation significantly outperforms others, further improving reward shaping. The above results suggest that lower dimensions of the Laplacian representation are more important to reward shaping. By learning a high-quality Laplacian representation, our method enables one to more easily choose eigenvectors to use for reward shaping, which leads to improved performance.

4.4. Analysis

Here we first empirically verify the two assumptions in Theorem 1, and then conduct ablative experiments with different choices of coefficients $c_1, \cdots, c_d$ for our generalized graph drawing objective (3). We use `GridRoom` and `GridMaze` environments for experiments in this subsection.

4.4.1. VERIFICATION OF ASSUMPTIONS

The first assumption requires that all optimization variables $\mathbf{u}_i$ lie in the span space of the d smallest eigenvectors during the optimization process, *i.e.*, $\forall i \in \text{span}(\{\mathbf{e}_1, \cdots, \mathbf{e}_d\})$.

We empirically verify a necessary and sufficient condition for this assumption: for each $\mathbf{u}_i$, the angle between it and its projection onto $\text{span}(\{\mathbf{e}_1, \cdots, \mathbf{e}_d\})$ (denoted as $\hat{\mathbf{u}}_i \triangleq \sum_{j=1}^{d}(\mathbf{e}_j^\top \mathbf{u}_i)\mathbf{e}_j$) is 0. Specifically, we compute the cosine distance between $\mathbf{u}_i$ and $\hat{\mathbf{u}}_i$ during training. As Fig. 9 shows,
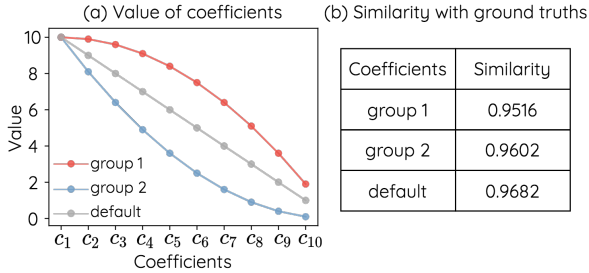
Figure 11. (a) Coefficient values of different groups. (b) Absolute cosine similarity (averaged across dimensions) between our learned representation and ground truth.

the cosine distance is close to 0 during the whole training process, which implies that the assumption holds in our experiments.

To verify whether the second assumption holds, *i.e.* whether the d smallest eigenvalues of the graph Laplacian are distinct: $\lambda_1 > \cdots > \lambda_d$, we calculate the smallest d eigenvalues of the graph Laplacian of our environments, and plot them in Fig. 10. It is clear that the eigenvalues are distinct, demonstrating the validity of this assumption.

4.4.2. EVALUATION ON OTHER COEFFICIENT CHOICES

In this above experiments, we choose the coefficients of our generalized graph drawing objective to be $c_i = (d - i + 1)$. In this subsubsection, we evaluate the effectiveness of other choices of $c_1, \cdots, c_d$.

Specifically, we select two groups of c_i that are different from the default group (*i.e.*, $c_i = (d - i + 1)$ as used in Eqn. 4), of which the first group has increasing first-order difference (*i.e.*, $c_{i+1} - c_i > c_i - c_{i-1}$), while the second group has decreasing first-order difference (*i.e.*, $c_{i+1} - c_i < c_i - c_{i-1}$). We plot the two groups in Fig. 11(a). For comparison, we also include the default coefficient group, which has a constant first-order difference of 1. We then train neural networks with the generalized objective in Eqn. (3) using group 1 and group 2 (other experimental settings are the same with Sec. 4.1). We use the similarity between learned representations and ground truth to evaluate the quality of the representations, as done in Sec. 4.1. As can be seen from Fig. 11(b), representations learned with objectives using group 1 or group 2 are as good as those learned by the default setting.

5. Related Works

By viewing the state transition process in RL as a graph where nodes are states and edges are transitions, previous works build a Laplacian-based state representation and successfully apply it in value function approximation (Mahadevan, 2005), option discovery (Machado et al., 2017) and reward shaping (Wu et al., 2019).

Mahadevan (2005) proposes proto-value functions, viewing the Laplacian representations as basis state representations, and use them to approximate value functions. Recently, Machado et al. (2017) introduce a framework for option discovery, which builds options from the Laplacian representation. They show that such options are useful for multiple tasks and helpful in exploration. Later, Machado et al. (2018) extend their Laplacian option discovery framework to settings where handcrafted features are not available, based on an connection between proto-value functions (Mahadevan, 2005) and successor representations (Dayan, 1993; Stachenfeld et al., 2014; Barreto et al., 2017). Jinnai et al. (2019) leverage the approximated Laplacian representation to learn deep covering options for exploration.

Our work focuses on better approximating the Laplacian representation in environments with large state space. Most related to our method is the approach proposed in (Wu et al., 2019). The authors optimize a spectral graph drawing objective (Koren, 2005) to approximate the eigenvectors. Though being efficient, their method has difficulties in learning a Laplacian representation close to the ground truth, due to the fact that their minimization objective has infinitely many other global minimizers besides the eigenvectors. Our work improves their method by proposing a new objective, which admits eigenvectors as its unique global minimizer and hence greatly increases the approximation quality in empirical evaluations. Other approaches for approximating the Laplacian representation include performing singular value decomposition on the incidence matrix (Machado et al., 2017; 2018), training neural networks with constrained stochastic optimization (Shaham et al., 2018) or bi-level stochastic optimization (Pfau et al., 2019). However, as discussed in (Wu et al., 2019), these approaches either require expensive matrix operations or suffer poor scalability.

6. Conclusion

Laplacian representation provides a succinct and informative state representation for RL, which captures the geometry of the underlying state space. Such representation is beneficial in discovering exploratory options and reward shaping. In this paper, we propose a new objective that greatly improves the approximation quality of the learned Laplacian, for environments with large or even continuous state space. We demonstrate the superiority of our method over previous work via theoretical analysis and empirical evaluation. Our method is efficient and simple to implement. With our method, one can learn high-quality Laplacian representation and apply it to various RL tasks such as option discovery and reward shaping.

References

Agarwal, R., Machado, M. C., Castro, P. S., and Bellemare, M. G. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qda7-sVg84.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H., and Silver, D. Successor features for transfer in reinforcement learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 4058–4068, 2017.

Chevalier-Boisvert, M., Willems, L., and Pal, S. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

Chung, F. R. and Graham, F. C. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

Coumans, E. and Bai, Y. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2019.

Dayan, P. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993.

Dubey, R., Agrawal, P., Pathak, D., Griffiths, T., and Efros, A. Investigating human priors for playing video games. In *International Conference on Machine Learning*, pp. 1349–1357. PMLR, 2018.

Fubini, G. *Sugli integrali multipli: nota*. Tipografia della R. Accademia dei Lincei, 1907. URL https://books.google.com.sg/books?id=NUEqnQAACAAJ.

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*, 2019.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. URL http://arxiv.org/abs/1412.6980.

Koren, Y. Drawing graphs by eigenvectors: theory and practice. *Computers & Mathematics with Applications*, 49(11-12):1867–1888, 2005.

Machado, M. C., Bellemare, M. G., and Bowling, M. A laplacian framework for option discovery in reinforcement learning. In *International Conference on Machine Learning*, pp. 2295–2304. PMLR, 2017.

Machado, M. C., Rosenbaum, C., Guo, X., Liu, M., Tesauro, G., and Campbell, M. Eigenoption discovery through the deep successor representation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Bk8ZcAxR-.

Machado, M. C., Bellemare, M. G., and Bowling, M. Count-based exploration with the successor representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5125–5133, 2020.

Mahadevan, S. Proto-value functions: Developmental reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, pp. 553–560, 2005.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Nachum, O., Gu, S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 3307–3317, 2018.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning*, pp. 2778–2787. PMLR, 2017.

Peter, K. and Lutz, A. *For the Beginning: The Finite Difference Method for the Poisson Equation*, pp. 19–45. Springer New York, New York, NY, 2003. ISBN 978-0-387-21762-8. doi: 10.1007/0-387-21762-2_2. URL https://doi.org/10.1007/0-387-21762-2_2.

Pfau, D., Petersen, S., Agarwal, A., Barrett, D. G. T., and Stachenfeld, K. L. Spectral inference networks: Unifying deep and spectral learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJzqpj09YQ.

Pong*, V., Gu*, S., Dalal, M., and Levine, S. Temporal difference models: Model-free deep RL for model-based control. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Skw0n-W0Z.

Shaham, U., Stanton, K., Li, H., Basri, R., Nadler, B., and Kluger, Y. Spectralnet: Spectral clustering using deep neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJ_aoCyRZ.

Stachenfeld, K. L., Botvinick, M., and Gershman, S. J. Design principles of the hippocampal cognitive map. *Advances in neural information processing systems*, 27: 2528–2536, 2014.

Stooke, A., Lee, K., Abbeel, P., and Laskin, M. Decoupling representation learning from reinforcement learning. *arXiv preprint arXiv:2009.08319*, 2020.

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

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

Wu, Y., Tucker, G., and Nachum, O. The laplacian in RL: Learning representations with efficient approximations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJlNpoA5YQ.

Zhang, A., Satija, H., and Pineau, J. Decoupling dynamics and reward for transfer learning. *arXiv preprint arXiv:1804.10689*, 2018.