A. Proof of Theorem 1

To prove Theorem 1, we first introduce the following Lemma 1.

Lemma 1. *Let $Q = (q_{ij}) \in \mathbb{R}^{d\times d}$ be an orthogonal matrix, and $A = (a_{ij}) = Q \odot Q = (q_{ij}^2)$. For $k \in \{1, \cdots, d-1\}$, we have*

$$\sum_{i=1}^{k}\sum_{j=k+1}^{d} a_{ij} = \sum_{i=1}^{k}\sum_{j=k+1}^{d} a_{ji} \tag{11}$$

Proof. Since Q is an orthogonal matrix, we know that the sum of A's first k rows is equal to that of A's first k columns, *i.e.*:

$$\sum_{i=1}^{k}\sum_{j=1}^{d} a_{ij} = \sum_{i=1}^{k}\sum_{j=1}^{d} a_{ji} = k \tag{12}$$

Therefore, we have

$$\begin{aligned}\sum_{i=1}^{k}\sum_{j=k+1}^{d} a_{ij} &= \sum_{i=1}^{k}\sum_{j=1}^{d} a_{ij} - \sum_{i=1}^{k}\sum_{j=1}^{k} a_{ij} \\ &= \sum_{i=1}^{k}\sum_{j=1}^{d} a_{ji} - \sum_{i=1}^{k}\sum_{j=1}^{k} a_{ij} \\ &= \sum_{i=1}^{k}\sum_{j=1}^{d} a_{ji} - \sum_{i=1}^{k}\sum_{j=1}^{k} a_{ji} \\ &= \sum_{i=1}^{k}\sum_{j=k+1}^{d} a_{ji}\end{aligned} \tag{13}$$

□

If we view A as a block matrix, *i.e.*

$$A = \begin{pmatrix} A_{1:k,1:k} & A_{1:k,k+1:d} \\ A_{k+1:d,1:k} & A_{k+1:d,k+1:d} \end{pmatrix}, \tag{14}$$

Lemma 1 says the sum of elements in $A_{1:k,k+1:d}$ is equal to the sum of elements in $A_{k+1:d,1:k}$.

With Lemma 1, now we can prove Theorem 1 as following.

Proof. Let E denote the matrix of the first d eigenvectors of d, *i.e.*, $E = (\mathbf{e}_1, \cdots, \mathbf{e}_d)$. Since $\mathbf{u}_i \in \mathrm{span}(\{\mathbf{e}_1, \cdots, \mathbf{e}_d\})$, and $\mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}$, we may rewrite $(\mathbf{u}_1, \cdots, \mathbf{u}_d) = EQ$, where $Q = (q_{ij}) \in \mathbb{R}^{d\times d}$ is an orthogonal matrix. Let $A = (a_{ij}) = Q \odot Q = (q_{ij}^2)$. Then, the objective of problem (3) becomes:

$$\begin{aligned} h(\mathbf{u}_1, \cdots, \mathbf{u}_d) &\triangleq \sum_{i=1}^{d} c_i \mathbf{u}_i^\top L \mathbf{u}_i \\ &= \sum_{i=1}^{d} c_i (\sum_{j=1}^{d} q_{ji}\mathbf{e}_j)^\top L (\sum_{j=1}^{d} q_{ji}\mathbf{e}_j) \\ &= \sum_{i=1}^{d} c_i \sum_{j=1}^{d} q_{ji}^2 \mathbf{e}_j^\top L \mathbf{e}_j \\ &= \sum_{i=1}^{d}\sum_{j=1}^{d} c_i a_{ji}\lambda_j \end{aligned} \tag{15}$$

We first prove optimality. Let g denote the gap between the objective and $\sum_{i=1}^{d} c_i\lambda_i$. We have

$$\begin{aligned} g &\triangleq \sum_{i=1}^{d} c_i \mathbf{u}_i^\top L \mathbf{u}_i - \sum_{i=1}^{d} c_i\lambda_i \\ &= \sum_{i=1}^{d} c_i \sum_{j=1}^{d} a_{ji}\lambda_j - \sum_{i=1}^{d} c_i\lambda_i \end{aligned} \tag{16}$$

Note that $\sum_{j=1}^{d} a_{ji} = 1$, then we have:

$$\begin{aligned} g &= \sum_{i=1}^{d} c_i \sum_{j=1}^{d} a_{ji}\lambda_j - \sum_{i=1}^{d} c_i \sum_{j=1}^{d} a_{ji}\lambda_i \\ &= \sum_{i=1}^{d} c_i \sum_{j=1}^{d} a_{ji}(\lambda_j - \lambda_i) \\ &= \sum_{i=1}^{d}\sum_{j=1}^{d} c_i a_{ji}(\lambda_j - \lambda_i) \end{aligned} \tag{17}$$

Let $\Delta_{ji} = \lambda_j - \lambda_i$, and $r_{ji} = c_i a_{ji}$, then we can rewrite g as:

$$g = \sum_{i=1}^{d}\sum_{j=1}^{d} r_{ji}\Delta_{ji} \tag{18}$$

Note that $\Delta_{ii} = 0$ and that, for $j \geqslant i$, $\Delta_{ji} = \Delta_{i+1,i} + \Delta_{i+2,i+1} + \cdots + \Delta_{j-1,j-2} + \Delta_{j,j-1} = \sum_{k=i}^{j-1} \Delta_{k+1,k}$. We then apply Fubini's Theorem (Fubini, 1907) to g:

$$\begin{aligned} g =& \sum_{j\geqslant i} r_{ji}\Delta_{ji} + \sum_{j\leqslant i} r_{ji}\Delta_{ji} \\ =& \sum_{j\geqslant i}(r_{ji}-r_{ij})\Delta_{ji} \\ =& \sum_{j>i}(r_{ji}-r_{ij})\sum_{k=i}^{j-1}\Delta_{k+1,k} \\ =& \sum_{j>k\geqslant i}(r_{ji}-r_{ij})\Delta_{k+1,k} \\ =& \sum_{k=1}^{d-1}(\sum_{i=1}^{k}\sum_{j=k+1}^{d}(r_{ji}-r_{ij}))\Delta_{k+1,k} \\ \triangleq& \sum_{k=1}^{d-1} s_k\Delta_{k+1,k} \end{aligned} \tag{19}$$

Note that for s_k, we have

$$\begin{aligned} s_k =& \sum_{i=1}^{k}\sum_{j=k+1}^{d}(r_{ji}-r_{ij}) \\ =& \sum_{i=1}^{k}\sum_{j=k+1}^{d}(c_i a_{ji} - c_j a_{ij}) \\ =& \sum_{i=1}^{k} c_i \sum_{j=k+1}^{d} a_{ji} - \sum_{j=k+1}^{d} c_j \sum_{i=1}^{k} a_{ij} \\ \geqslant& c_k \sum_{i=1}^{k}\sum_{j=k+1}^{d} a_{ji} - c_{k+1}\sum_{j=k+1}^{d}\sum_{i=1}^{k} a_{ij} \end{aligned} \tag{20}$$

According to Lemma 1, we know

$$\sum_{i=1}^{k}\sum_{j=k+1}^{d} a_{ji} = \sum_{j=k+1}^{d}\sum_{i=1}^{k} a_{ij} \tag{21}$$

Therefore, we have

$$s_k \geqslant (c_k - c_{k+1})\sum_{i=1}^{k}\sum_{j=k+1}^{d} a_{ji} \geqslant 0. \tag{22}$$

Since $\Delta_{k+1,k} > 0$, with Eqn. (22), we can obtain

$$\begin{aligned} g =& \sum_{i=1}^{d} c_i \mathbf{u}_i^\top L \mathbf{u}_i - \sum_{i=1}^{d} c_i\lambda_i \\ =& \sum_{k=1}^{d-1} s_k \Delta_{k+1,k} \\ \geqslant& 0. \end{aligned} \tag{23}$$

I.e., the following inequality holds:

$$\sum_{i=1}^{d} c_i \mathbf{u}_i^\top L \mathbf{u}_i \geqslant \sum_{i=1}^{d} c_i\lambda_i \tag{24}$$

Since $\mathbf{e}_i^\top L \mathbf{e}_i = \lambda_i$, the inequality is tight when

$$(\mathbf{u}_1, \cdots, \mathbf{u}_d) = (\mathbf{e}_1, \cdots, \mathbf{e}_d). \tag{25}$$

Therefore, we conclude that $\sum_{i=1}^{d} c_i\lambda_i$ is the global minimum, and $(\mathbf{e}_1, \cdots, \mathbf{e}_d)$ is one minimizer.

Next, we prove uniqueness. Assume that there is another minimizer for this problem, denoted as $(\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_d)$. We have

$$\begin{aligned} & (\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_d) \neq (\mathbf{e}_1, \cdots, \mathbf{e}_d) \\ \Leftrightarrow \quad & \exists i \in \{1, \cdots, d\}, \tilde{\mathbf{u}}_i \neq \pm\mathbf{e}_i \end{aligned} \tag{26}$$

Here we require $\tilde{\mathbf{u}}_i \neq \pm\mathbf{e}_i$ because the sign of $\mathbf{e}_i$ is arbitrary and hence we do not distinguish them. Again, $(\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_d)$ can be written as $(\mathbf{e}_1, \cdots, \mathbf{e}_d)\tilde{Q}$, where $\tilde{Q} = (\tilde{q}_{ij}) \in \mathbb{R}^{d\times d}$ is an orthogonal matrix. Therefore, proposition in Eqn. (26) is equivalent to

$$\exists i \in \{1, \cdots, d\}, \tilde{q}_{ii} \notin \{1, -1\}. \tag{27}$$

Denote $\tilde{A} = (\tilde{a}_{ij}) = \tilde{Q} \odot \tilde{Q} = (\tilde{q}_{ij}^2)$. By the optimality of $(\tilde{\mathbf{u}}_1, \cdots, \tilde{\mathbf{u}}_d)$, we have

$$\sum_{i=1}^{d} c_i \tilde{\mathbf{u}}_i^\top L \tilde{\mathbf{u}}_i - \sum_{i=1}^{d} c_i\lambda_i = 0 \tag{28}$$

From Eqn. (16) to Eqn. (22), we know

$$\sum_{i=1}^{d} c_i \tilde{\mathbf{u}}_i^\top L \tilde{\mathbf{u}}_i - \sum_{i=1}^{d} c_i\lambda_i \geqslant 0 \tag{29}$$

The equality holds if and only if $\tilde{a}_{ji} = 0, \forall (i,j) \in \{(i,j) |\, j > i\}$. Additionally, according to Lemma 1, we have

$$\sum_{i=1}^{k}\sum_{j=k+1}^{d} \tilde{a}_{ji} = \sum_{i=1}^{k}\sum_{j=k+1}^{d} \tilde{a}_{ij} \tag{30}$$

Therefore, we also have $\tilde{a}_{ji} = 0, \forall (i,j) \in \{(i,j) |\, j < i\}$. Accordingly, all off-diagonal elements of $\tilde{A}$ are 0, *i.e.*, $\tilde{a}_{ij} = 0, \forall i \neq j$. Moreover, since $\tilde{Q}$ is orthogonal, the following equality holds

$$\sum_{j=1}^{d} \tilde{a}_{ij} = \sum_{j=1}^{d} \tilde{q}_{ij}^2 = 1, \forall i \in \{1, \cdots, d\}. \tag{31}$$

So we have

$$\begin{aligned} & \forall i \in \{1, \cdots, d\}, \tilde{a}_{ii} = 1, \\ \Leftrightarrow \quad & \forall i \in \{1, \cdots, d\}, \tilde{q}_{ii} \in \{1, -1\} \end{aligned} \tag{32}$$

which contradicts with proposition in Eqn. (27). Based on the above, we conclude that $(\mathbf{e}_1, \cdots, \mathbf{e}_d)$ is the unique global miminizer. □

B. Extension to Continuous setting

In Sec. 2 and Sec. 3, we discuss the Laplacian representation and our proposed objective in discrete case. In this section we extend previous discussions to continuous settings. Consider a graph with infinitely many nodes (*i.e.*, states), where weighted edges represent pairwise non-negative affinities (denoted by $D(u, v) \geq 0$ for nodes u and v).

Following (Wu et al., 2019), we give the following definitions. A Hilbert space $\mathcal{H}$ is defined to be the set of square-integrable real-valued functions on graph nodes, *i.e.* $\mathcal{H} = \{f : \mathcal{S} \to \mathbb{R} \mid \int_{\mathcal{S}} |f(u)|^2 \, d\rho(u) < \infty\}$, associated with the inner-product

$$\langle f, g \rangle_{\mathcal{H}} = \int_{\mathcal{S}} f(u)g(u) \, d\rho(u), \tag{33}$$

where ρ is a probability measure, *i.e.* $\int_{\mathcal{S}} d\rho(u) = 1$. The norm of a function f is defined as $\langle f, f \rangle_{\mathcal{H}}$. Functions f, g are orthogonal if $\langle f, g \rangle_{\mathcal{H}} = 0$; functions $f_1, \cdots, f_d$ are orthonormal if $\langle f_i, f_j \rangle_{\mathcal{H}} = \delta_{ij}, \forall i, j \in \{1, \cdots, d\}$. The graph Laplacian is defined as a linear operator $\mathscr{L}$ on $\mathcal{H}$, given by

$$\mathscr{L} f(u) = f(u) - \int_{\mathcal{S}} f(v)D(u, v) \, d\rho(v). \tag{34}$$

Our goal is to learn $f_1, \cdots, f_d$ for approximating the d eigenfunctions $e_1, \cdots, e_d$ associated with the smallest d eigenvalues $\lambda_1, \cdots, \lambda_d$ of $\mathscr{L}$. The graph drawing objective used in (Wu et al., 2019) is

$$\begin{aligned} \min_{f_1, \cdots, f_d} \quad & \sum_{i=1}^{d} \langle f_i, \mathscr{L} f_i \rangle_{\mathcal{H}} \\ \text{s.t.} \quad & \langle f_i, f_j \rangle_{\mathcal{H}} = \delta_{ij}, \forall i, j = 1, \cdots, d. \end{aligned} \tag{35}$$

Extending this objective to the generalized form gives us

$$\begin{aligned} \min_{f_1, \cdots, f_d} \quad & \sum_{i=1}^{d} c_i \langle f_i, \mathscr{L} f_i \rangle_{\mathcal{H}} \\ \text{s.t.} \quad & \langle f_i, f_j \rangle_{\mathcal{H}} = \delta_{ij}, \forall i, j = 1, \cdots, d. \end{aligned} \tag{36}$$

Similarly, for continuous setting, Theorem 1 can be extended to the following theorem:

Theorem 2. *Assume $\forall i, f_i \in \operatorname{span}(\{e_1, \cdots, e_d\})$, and $\lambda_1 < \cdots < \lambda_d$). Then, $c_1 > \cdots > c_d$ is a sufficient condition for the generalized graph drawing objective to have a unique global minimizer $(f_1^*, \cdots, f_d^*) = (e_1, \cdots, e_d)$, and the corresponding minimum is $\sum_{i=1}^{d} c_i \lambda_i$.*

To prove the Theorem 2, we need the following Lemma 2 and Lemma 3.

Lemma 2. *Let $f_1, \cdots, f_d$ be d orthonormal functions in $\operatorname{span}(\{e_1, \cdots, e_d\})$, and q_{ji} be the inner product of f_i and e_j,* i.e., *$q_{ji} = \langle f_i, e_j \rangle_{\mathcal{H}}, \forall i, j \in \{1, \cdots, d\}$. Then we have (i) $\forall i \in \{1, \cdots, d\}, \sum_{j=1}^{d} q_{ji}^2 = 1$, and (ii) $\forall j \in \{1, \cdots, d\}, \sum_{i=1}^{d} q_{ji}^2 = 1$.*

Proof. First, since $e_1, \cdots, e_d$ form an orthonormal basis, consider projection of f_i onto $e_1, \cdots, e_d$. We have

$$f_i = \sum_{j=1}^{d} \langle f_i, e_j \rangle_{\mathcal{H}} e_j = \sum_{j=1}^{d} q_{ji} e_j. \tag{37}$$

Since f_i has a norm of 1, we have

$$\begin{aligned} \langle f_i, f_i \rangle_{\mathcal{H}} &= \langle \sum_{j=1}^{d} q_{ji} e_j, \sum_{j=1}^{d} q_{ji} e_j \rangle_{\mathcal{H}} \\ &= \sum_{j=1}^{d} q_{ji}^2 = 1. \end{aligned} \tag{38}$$

The above equation proves (i). Then, consider projection of e_j onto $f_1, \cdots, f_d$ (note that $f_1, \cdots, f_d$ also form an orthogonal basis for the subspace spanned by $e_1, \cdots, e_d$). We have,

$$e_j = \sum_{i=1}^{d} \langle e_j, f_i \rangle_{\mathcal{H}} f_i = \sum_{i=1}^{d} q_{ji} f_i. \tag{39}$$

Since e_j also has a norm of 1, we have

$$\begin{aligned} \langle e_j, e_j \rangle_{\mathcal{H}} &= \langle \sum_{i=1}^{d} q_{ji} f_i, \sum_{i=1}^{d} q_{ji} f_i \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{d} q_{ji}^2 = 1. \end{aligned} \tag{40}$$

This equation shows that (ii) holds. □

Lemma 3. *Let $f_1, \cdots, f_d$ be d orthonormal functions in $\operatorname{span}(\{e_1, \cdots, e_d\})$, q_{ji} be the inner product of f_i and e_j,* i.e., *$q_{ji} = \langle f_i, e_j \rangle_{\mathcal{H}}, \forall i, j \in \{1, \cdots, d\}$, and $a_{ji} = q_{ji}^2$. Then, for $k \in \{1, \cdots, d-1\}$, we have*

$$\sum_{i=1}^{k} \sum_{j=k+1}^{d} a_{ij} = \sum_{i=1}^{k} \sum_{j=k+1}^{d} a_{ji} \tag{41}$$

Proof. By Lemma 2, we have

$$\sum_{i=1}^{k} \sum_{j=1}^{d} a_{ij} = \sum_{i=1}^{k} \sum_{j=1}^{d} a_{ji} = k \tag{42}$$

Therefore, we have

$$\begin{aligned}\sum_{i=1}^{k}\sum_{j=k+1}^{d} a_{ij} &=\sum_{i=1}^{k}\sum_{j=1}^{d} a_{ij}-\sum_{i=1}^{k}\sum_{j=1}^{k} a_{ij}\\ &=\sum_{i=1}^{k}\sum_{j=1}^{d} a_{ji}-\sum_{i=1}^{k}\sum_{j=1}^{k} a_{ij}\\ &=\sum_{i=1}^{k}\sum_{j=1}^{d} a_{ji}-\sum_{i=1}^{k}\sum_{j=1}^{k} a_{ji}\\ &=\sum_{i=1}^{k}\sum_{j=k+1}^{d} a_{ji}\end{aligned} \tag{43}$$

□

With Lemma 3, we can prove Theorem 2.

Proof. Since $f_i \in \mathrm{span}(\{e_1, \cdots, e_d\})$, without loss of generality, we may rewrite f_i as

$$f_i = \sum_{j=1}^{d} q_{ji} e_j, \tag{44}$$

where $q_{ji} = \langle f_i, e_j \rangle_{\mathcal{H}}$. Then, the objective of problem (36) is

$$\begin{aligned}h(f_1,\cdots,f_d) &\triangleq \sum_{i=1}^{d} c_i \langle f_i, \mathscr{L} f_i\rangle_{\mathcal{H}}\\ &=\sum_{i=1}^{d} c_i \langle \sum_{j=1}^{d} q_{ji}e_j, \mathscr{L}\sum_{j=1}^{d} q_{ji}e_j\rangle_{\mathcal{H}}\\ &=\sum_{i=1}^{d} c_i \langle \sum_{j=1}^{d} q_{ji}e_j, \sum_{j=1}^{d} q_{ji}\mathscr{L}e_j\rangle_{\mathcal{H}}\\ &=\sum_{i=1}^{d} c_i \langle \sum_{j=1}^{d} q_{ji}e_j, \sum_{j=1}^{d} q_{ji}\lambda_j e_j\rangle_{\mathcal{H}}\\ &=\sum_{i=1}^{d} c_i \sum_{j=1}^{d} q_{ji}^2\lambda_j\langle e_j, e_j\rangle_{\mathcal{H}}\\ &=\sum_{i=1}^{d} c_i \sum_{j=1}^{d} a_{ji}\lambda_j,\end{aligned} \tag{45}$$

where $a_{ji} = q_{ji}^2$.

Let g denote the gap between the objective and $\sum_{i=1}^{d} c_i\lambda_i$. Then we have

$$\begin{aligned}g &\triangleq \sum_{i=1}^{d} c_i\langle f_i, \mathscr{L} f_i\rangle_{\mathcal{H}} - \sum_{i=1}^{d} c_i\lambda_i\\ &=\sum_{i=1}^{d} c_i \sum_{j=1}^{d} a_{ji}\lambda_j - \sum_{i=1}^{d} c_i\lambda_i\\ &=\sum_{i=1}^{d} c_i \sum_{j=1}^{d} a_{ji}(\lambda_j-\lambda_i)\\ &=\sum_{i=1}^{d}\sum_{j=1}^{d} c_i a_{ji}(\lambda_j-\lambda_i)\end{aligned} \tag{46}$$

We can see that Eqn. (46) has the same form as Eqn. (17). Thus we can follow the same steps as in the proof of Theorem 1 (*i.e.*, from Eqn. (18) to Eqn. (32), replacing Lemma 1 with Lemma 3) to finish proving Theorem 2. □

C. Obtaining Training objective

In (Wu et al., 2019), the authors express the graph drawing objective as an expectation

$$\mathbb{E}_{(s,s')\sim\mathcal{T}} \sum_{i=1}^{k} \left(f_i(s) - f_i(s')\right)^2 \tag{47}$$

and transform the orthonormal constraints into the following penalty term

$$\mathbb{E}_{s\sim\rho, s'\sim\rho} \sum_{i,j}^{k} \left(f_i(s)f_j(s) - \delta_{ij}\right)\left(f_i(s')f_j(s') - \delta_{ij}\right). \tag{48}$$

Here k denotes the dimension of the representation and $\sum_{i,j}^{k}$ is short for $\sum_{i=1}^{k}\sum_{j=1}^{k}$. From Eqn. (5), we can see that our objective can be viewed as the sum of d graph drawing objectives. Thus we can obtain Eqn. (6) by summing d objectives in Eqn. (47) with k varying from 1 to d. Similarly, we can obtain Eqn. (7) by summing d penalty terms in Eqn. (48).

D. Environment Descriptions

Two discrete gridworld environments used in our experiments: `GridRoom` and `GridMaze`, are built with MiniGrid (Chevalier-Boisvert et al., 2018). The `GridRoom` environment is a 20×20 grid with 271 states, and the `GridMaze` environment is a 18×18 grid with 161 states. In both environments, the agent has 4 four actions: moving *left*, *right*, *up* and *down*. When the agent hits the wall, it remains in previous position. Two raw state representations are considered: (x, y) coordinates (scaled within $[-1, 1]$) and top-view image of the grid (scaled within $[0, 1]$).

Two continuous control navigation environments used in our experiments: `PointRoom` and `PointMaze`, are built with PyBullet (Coumans & Bai, 2016–2019). The `PointRoom` environment is of size 20×20 and each room is of size 5×5. The `GridMaze` environment is of size 18×18 and the width of each corridor is 2. For both environments, a ball with diameter 1 is controlled to navigate in the environment. It takes a continuous action (within range $[0, 2\pi]$) to decide the direction and then move a small step forward along this direction. We consider the (x, y) positions as the raw state representations.

E. Experiment Configurations

E.1 Learning Laplacian Representations

For learning Laplacian representations on `GridRoom` and `GridMaze` environments, we collect a dataset of 100,000 transitions using a uniformly random policy with random starts. Each episode has a length of 50. Following (Wu et al., 2019), we use a fully connected neural network for (x, y) position observations and a convolutional neural network for image observations. The network structures are described in Tab. 2 and Tab. 3. An additional linear layer is used to map the output into representations. We train the networks for 200,000 iterations by Adam optimizer (Kingma & Ba, 2015) with batch size 1024 and learning rate 0.001. The weight for the penalty term in Eqn. (7) is set to 1.0. Following (Wu et al., 2019), we use the discounted multi-step transitions with discount parameter 0.9.

For learning Laplacian representations on `PointRoom` and `PointMaze` environments, we collect a dataset of 1,000,000 transitions using a uniformly random policy with random starts. Each episode has a length of 500. We use the same fully connected network as mentioned above and keep other configurations unchanged except using a larger batch size of 8192.

For computing SimGT and SimRUN for continuous states, we calculate the inner summation in Eqn. (9) and Eqn. (10) over sampled states rather than the entire state space.

Table 2. Network architecture of the fully connected network.

Layer	Number of units	Activation
Linear	256	ReLU
Linear	256	ReLU
Linear	256	ReLU

Table 3. Network architecture of the convolutional network. (C, K, S, P) correspond to number of output channels, kernel size, stride and padding.

Layer	Configurations (C, K, S, P)	Activation
Conv2D	(16, 4, 2, 2)	ReLU
Conv2D	(16, 4, 2, 2)	ReLU
Conv2D	(16, 4, 1, 0)	ReLU

Table 4. Hyperparameters of DQN for learning options.

Timesteps	100,000
Episode length	50
Optimizer	Adam
Learning rate	1e-3
Learning starts	5000
Training frequency	1
Target update frequency	50
Target update rate	0.05
Replay size	100,000
Batch size	128
Discount factor γ	0

E.2 Option Discovery

We run option discovery experiments on `GridRoom` and `GridMaze` environments with (x, y) position observations. Following (Machado et al., 2017), we approximate the options greedily ($\gamma = 0$). For each dimension of the learned representation, one option is trained by Deep Q-learning (Mnih et al., 2013) with an intrinsic reward function $r_i(s, s') = f_i(s) - f_i(s')$ and the other with $-r_i(s, s')$. The termination set of an option is defined as the set of states where $f_i(s)$ is a local maximum (or minimum for the other direction). For the deep Q-network (DQN), we use the same fully connected network as one used for learning representations. The hyperparameters for training DQN are summarized in Tab. 4.

To compute $N_{i \to j}$, we first augment the agent's action space with the learned options. For each starting state in room i, we record how many steps an agent takes to arrive in room j when it follows a uniformly random policy. We run 50 trajectories for each starting state to stabilize the result.

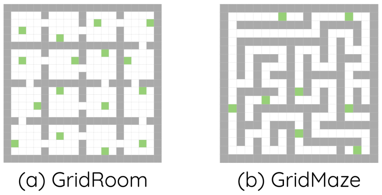

Figure 12. Goal positions in `GridRoom` and `GridMaze` for reward shaping experiments. Each green cell represents a goal.

Table 5. Hyperparameters of DQN for reward shaping.

Timesteps	200,000
Episode length	150
Optimizer	Adam
Learning rate	1e-3
Learning starts	5000
Training frequency	1
Target update frequency	50
Target update rate	0.05
Replay size	100,000
Batch size	128
Discount factor γ	0.99

E.3 Reward Shaping

We run reward shaping experiments on `GridRoom` and `GridMaze` environments. Following (Wu et al., 2019), we train the agent in goal-achieving tasks using Deep Q-learning (Mnih et al., 2013) with (x, y) positions as observations. At each step, the agent receives a reward of 0 if it reaches the goal state and -1 otherwise. The success rate of reaching the goal state is used to measure the performance. As mentioned in the main paper, we use multiple goals to eliminate the bias brought by the goal position. Their locations are depicted in Fig. 12. For the Q-network, we use the same fully connected network as one used for learning representations. The hyperparameters for training DQN are summarized in Tab. 5.

F. Additional Results

F.1 Learning Laplacian Representations

In Sec. 4.1, Fig. 3 and Fig. 4 visualize the learned representations on `GridMaze` and `PointRoom`. Here we include additional visualizations for `GridRoom` and `PointMaze` in Fig. 15 and Fig. 16.

In Sec. 4.1, Fig. 5 visualize first 3 dimensions of learned representations in different runs on `GridRoom`. Here we show all 10 dimensions in Fig. 17 and Fig. 18.

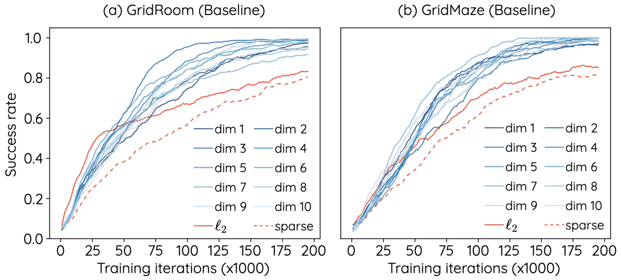

Figure 13. Results of reward shaping with each dimension of Laplacian representations learned by baseline method. ℓ_2 denotes reward shaping with L2 distance in raw observation space (*i.e.*, (x, y) position), and *sparse* denotes no reward shaping.

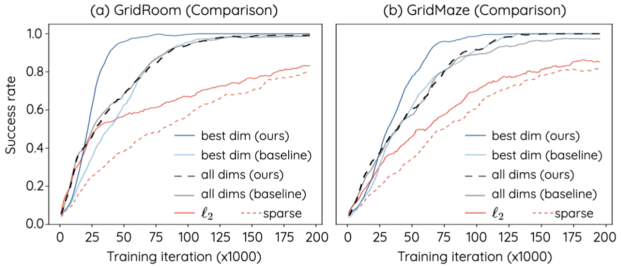

Figure 14. Results of reward shaping with learned Laplacian representations. ℓ_2 denotes reward shaping with L2 distance in raw observation space (*i.e.*, (x, y) position), and *sparse* denotes no reward shaping.

Table 6. Absolute cosine similarity (averaged across dimensions) between our learned representation and ground truth, on `GridRoom` environment.

Coefficients	Similarity
group 1	0.9905
group 2	0.9653
default	0.9913

F.2 Reward Shaping

For completeness, we show the results with each dimension of learned representation for baseline method in Fig. 13, and include the results for “all dims - ours” in Fig. 14.

F.3 Evaluation On Other Coefficient Choices

In Sec. 4.4.2, Fig. 11 shows the similarities between our learned representation (with different coefficient groups) and the ground truth on `GridMaze`. Here we show the results on `GridRoom` in Tab. 6.

F.4 Visualization of the discovered options

In Fig. 19 and 20, we visualize the discovered options by different representations.

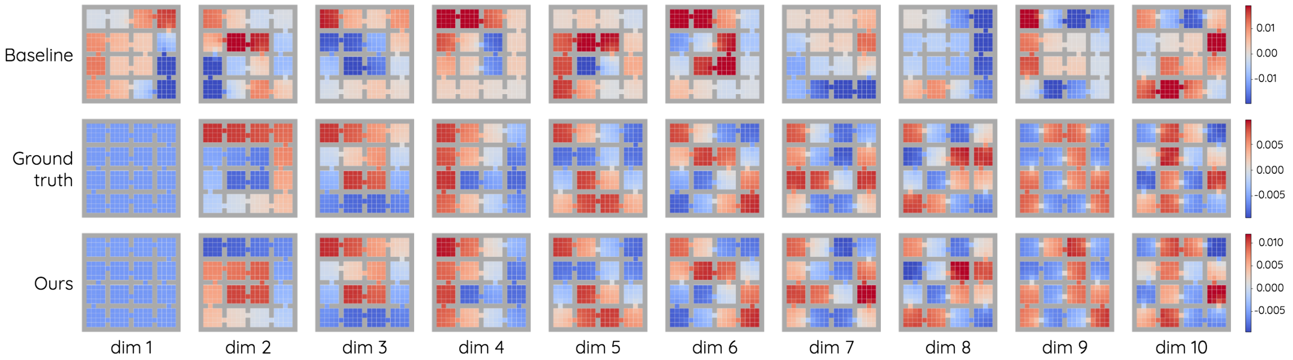

Figure 15. Visualization of the learned 10-dimension Laplacian representation and the ground truth on `GridRoom`. Each heatmap shows a dimension of the representation for all states in the environment. Best viewed in color.

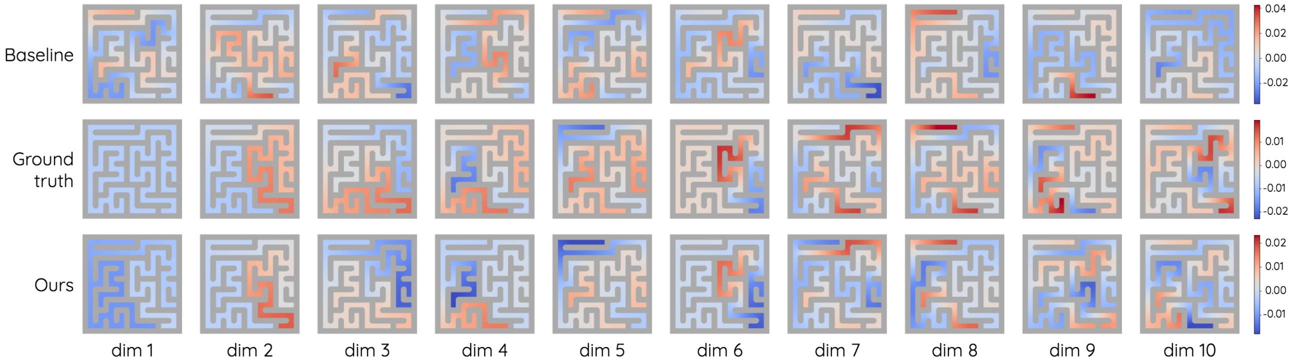

Figure 16. Visualization of the learned 10-dimension Laplacian representations and the ground truth on `PointMaze`. Each heatmap shows a dimension of the representation for all the states in the environment. Best viewed in color.

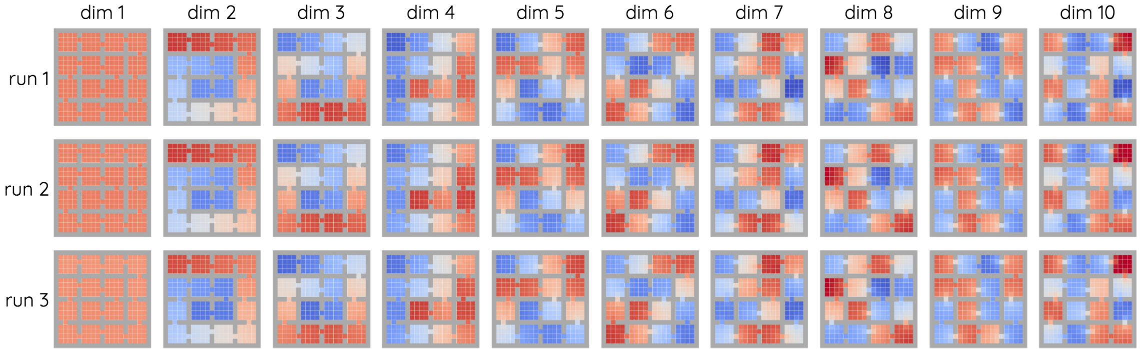

Figure 17. Visualization of the Laplacian representations learned by our method on `GridRoom` in 3 different runs.

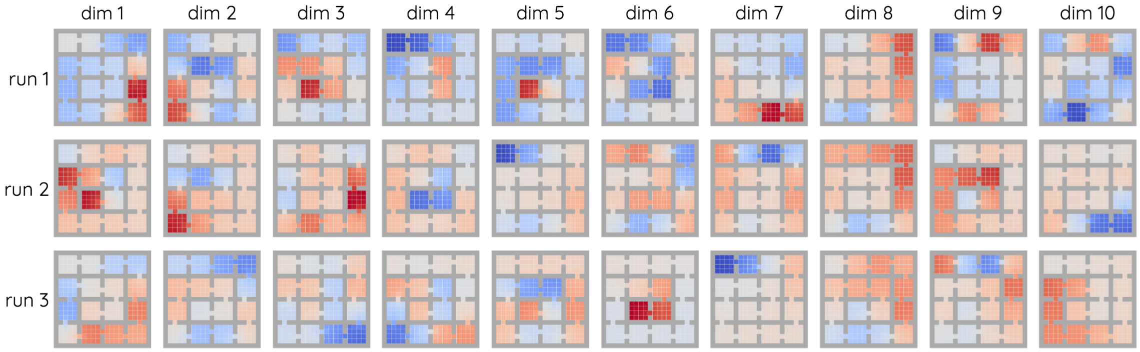

Figure 18. Visualization of the Laplacian representations learned by baseline method `GridRoom` in 3 different runs.

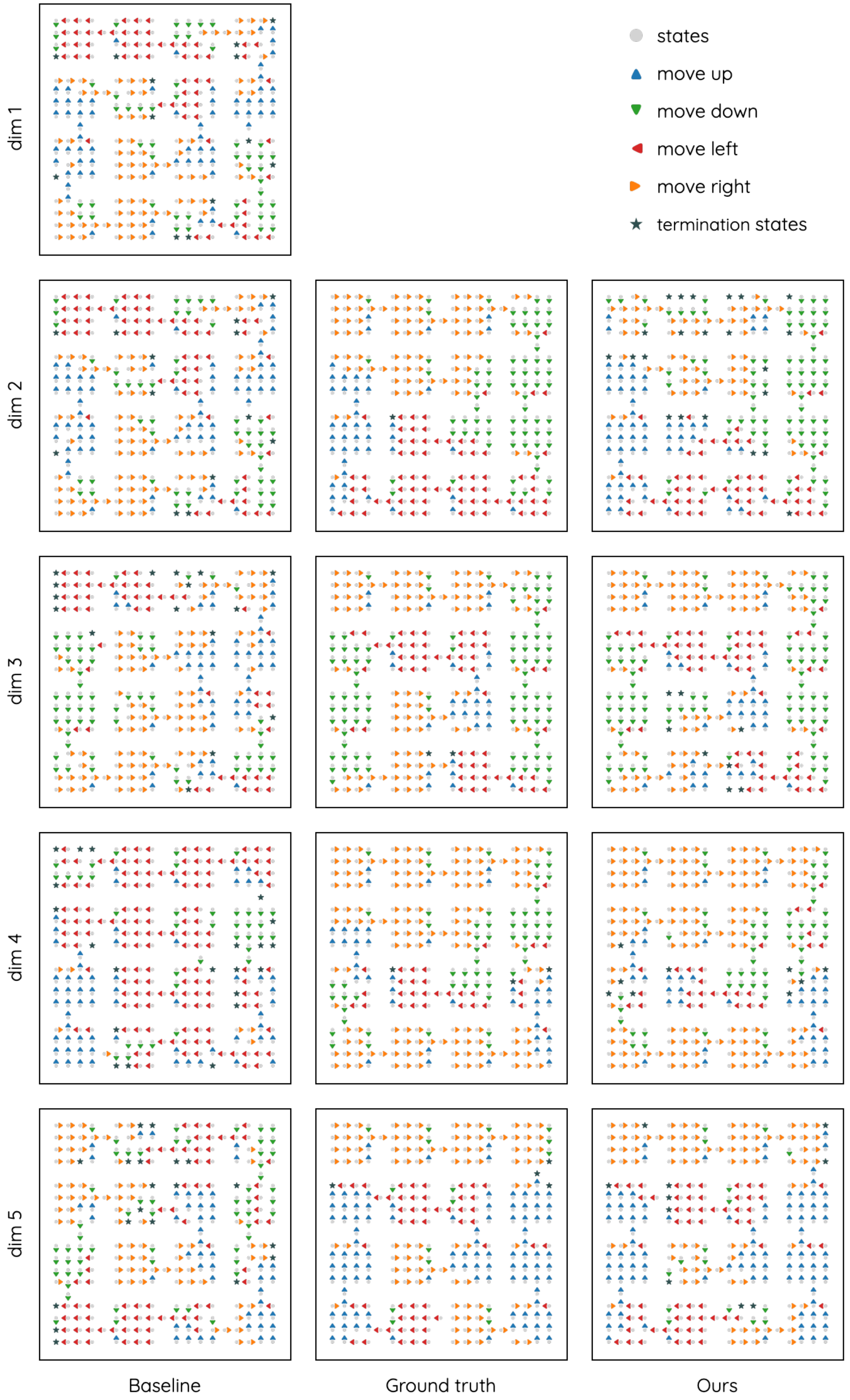

Figure 19. Visualization of the discovered options in `GridRoom`.

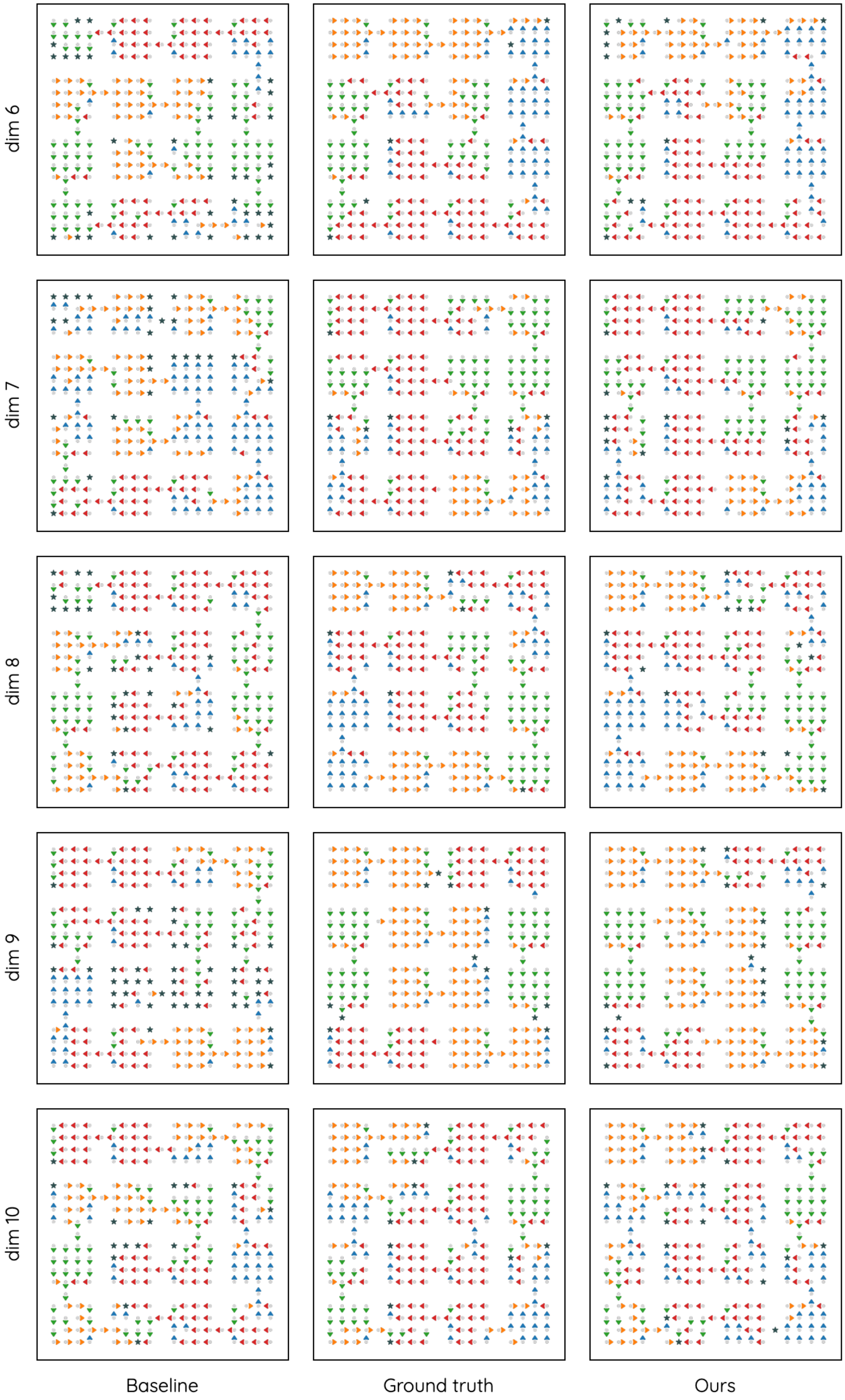

Figure 20. Visualization of the discovered options in `GridRoom` (continued).