Probabilistic World Modeling with Asymmetric Distance Measure

Meng Song [1]

Abstract

Representation learning is a fundamental task in machine learning, aiming at uncovering structures from data to facilitate subsequent tasks. However, what is a good representation for planning and reasoning in a stochastic world remains an open problem. In this work, we posit that learning a distance function is essential to allow planning and reasoning in the representation space. We show that a geometric abstraction of the probabilistic world dynamics can be embedded into the representation space through asymmetric contrastive learning. Unlike previous approaches that focus on learning single-step mutual similarity or compatibility measures, we instead learn an asymmetric similarity function that allows irreversible state reachability and multi-way probabilistic inference. Moreover, by conditioning on a common reference state (e.g. the observer's current state), the learned representation space allows us to discover the geometrically salient states that only a handful of paths can lead through. These states can naturally serve as subgoals to break down long-horizon planning tasks. We evaluate our method in gridworld environments with various layouts and demonstrate its effectiveness in discovering the subgoals.

1. Introduction

Learning good representations from the data plays a crucial role in the success of modern machine learning algorithms (Bengio et al., 2013). It requires an AI system to have the ability to extract rich structures from the data and build a model of the world. What structures should be preserved, summarized, and what should be abstracted out is heavily dependent on the downstream tasks and learning objectives. In this paper, we consider the problem of *what is a good representation for planning and reasoning in a stochastic world*. We will derive the problem formulation by delving into the definitions of planning and reasoning, along with the stochastic nature of the world.

On one hand, planning and reasoning typically involve an optimization process finding the most probable future outcomes or the most effective path to the goal. This optimization ability sets it apart from retrieving and interpolating good solutions seen in the training data, which imitation learning algorithms are good at. Instead, it enables problem-solving in new ways. *To allow for such optimization, a fundamental requirement for the representation space is to have a distance function that accurately measures the probability of an event occurring given another or the distance from the current state to the goal state.* On the other hand, the world inherently operates as a stochastic dynamical system, transiting from one state to another, often formulated as a Markov chain. By leveraging the transition graph induced from the Markov chain, we can estimate the probability of transitioning from state $\mathbf{x}$ to state $\mathbf{y}$ within C time steps, which we refer to as *C-step reachability from $\mathbf{x}$ to $\mathbf{y}$*. Rather than looking at the single-step transition like typical model-based RL methods, C-step reachability considers all possible paths within C steps from $\mathbf{x}$ to $\mathbf{y}$, thus allowing us to do multi-way probabilistic inference and answer questions such as "Given that event $\mathbf{x}$ has already occurred, how probable is it for event $\mathbf{y}$ to occur in the future, considering all possible ways the future could unfold?", "How likely to reach $\mathbf{y}$ from $\mathbf{x}$ considering all possible paths?"

By integrating the above principles, we formulate the representation learning problem for planning and reasoning in a stochastic world as the task of learning an embedding space. This space's distance function reflects the state reachability induced by a Markov chain. To address this challenge, we encode C-step reachability into an asymmetric similarity function by binary NCE (Gutmann & Hyvärinen, 2012; Ma & Collins, 2018). A significant volume of prior studies on representation learning in NLP (Mikolov et al., 2013), computer vision (Chen et al., 2020; He et al., 2020; Chen et al., 2023) and RL (Ghosh et al., 2018; Wu et al., 2018) have implicitly modeled the co-occurrence statistics on an undirected graph and embed it by a single mapping function. In contrast to these prior works, we use a pair of embedding mappings to model the dual roles of a state in a directed

[1] UC San Diego. Correspondence to: Meng Song <mes050@ucsd.edu>.

Extended Abstract Track of the Geometry-grounded Representation Learning and Generative Modeling Workshop at 41st International Conference on Machine Learning, Vienna, Austria.

graph: as an outgoing vertex and an incoming vertex. This approach assures an asymmetric similarity function reflecting the asymmetric even irreversible transition probabilities resulting from the temporal order or causalities (e.g. the transition of food from raw to cooked), which is crucial for planning and event modeling.

Furthermore, the learned representation space also provides a geometric abstraction of the underlying directed graph in a perspective-dependent way. We find that by conditioning on a common reference state, a symmetric distance measure can be recovered to measure the point density in the representation space. This naturally gives rise to the notion of subgoals, which denote the geometrically salient states that only a handful of paths can lead through based on the agent's current state. Our definition of subgoals aligns well with everyday experiences but has not been explored by the previous works (Goyal et al., 2019; Nachum et al., 2018). For instance, at a theme park entrance, a ticket is required for entry, but not for exit. Thus, the entrance serves as a subgoal upon arrival but not upon leaving. The proposed subgoals can be identified as low-density regions using any density-based clustering algorithms. We demonstrate the effectiveness of our approach to subgoal discovery in a variety of gridworld environments.

2. Problem formulation

Suppose that a Markov chain $\mathcal{M}$ on state space $\mathcal{S}$ has an initial distribution $\rho(S_0)$ and an **unknown** transition probability distribution $P(S_{t+1}|S_t) \triangleq P$. Unlike the previous works (Grover & Leskovec, 2016; Ghosh et al., 2018; Wu et al., 2018), we do not require a uniform initial distribution on the state space since enumerating an unknown or uncountable state space at the very beginning of a Markov chain is often infeasible. Instead, we study the typical cases where the initial distribution $\rho(S_0)$ is highly concentrated.

2.1. Vertex reachability

Let random variables U and W represent any vertex on the directed transition graph $G = (V, E, P)$ induced from $\mathcal{M}$. The reachability from vertex $\mathbf{u}$ to vertex $\mathbf{w}$ can be defined as

$$P(W = \mathbf{w}|U = \mathbf{u}) = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} P_t(S_t = \mathbf{w}|S_0 = \mathbf{u}) \tag{1}$$

where $P_t(S_t = \mathbf{w}|S_0 = \mathbf{u})$ is the probability of traveling from $\mathbf{u}$ to $\mathbf{w}$ in exactly t steps. The reachability from $\mathbf{u}$ to $\mathbf{w}$ can be therefore understood as the likelihood of reaching $\mathbf{w}$ from $\mathbf{u}$ within any number of steps.

Specifically, the sequence of vertices $S_0 \to S_1 \to \cdots \to S_t$ is a t-step walk on G starting from $\mathbf{u}$ to $\mathbf{w}$, whose probability can be factorized as the following by thinking of G as a Bayesian network:

$$\begin{aligned} &P_t(S_t = \mathbf{w}|S_0 = \mathbf{u}) \\ &= \sum_{S_1,\ldots,S_{t-1}} P(S_1, \ldots, S_{t-1}, S_t = \mathbf{w}|S_0 = \mathbf{u}) \\ &= \sum_{S_1,\ldots,S_{t-1}} \prod_{i=0}^{t-1} P(S_{i+1}|S_i) \end{aligned} \tag{2}$$

2.2. C-step approximation

Computing the vertex reachability is often intractable since it requires enumerating all possible paths over a near-infinite horizon. In practice, we approximate (1) by looking ahead C steps in a Markov chain. Formally, suppose that we have a T-step Markov chain $M = \{S_0, S_1, \ldots, S_T\}$ with transition probability distribution $P(S_{t+1}|S_t) \triangleq P, t = 0, 1, \ldots, T-1$. We denote any preceding random variables in the chain as the random variable $Y = \{S_i \mid 0 \leq i \leq T-1\}$, and any subsequent random variables within C steps from Y as the random variable $X = \{S_j \mid i < j \leq \min(i + C, T)\}$. The joint distribution $P(X = \mathbf{w}, Y = \mathbf{u})$ represents the occurrence frequency of **the ordered pair** $(\mathbf{u}, \mathbf{w})$ on all the possible paths within C steps. The reachability from $\mathbf{u}$ to $\mathbf{w}$ can be approximated as

$$\begin{aligned} P(W = \mathbf{w}|U = \mathbf{u}) &\approx P(X = \mathbf{w}|Y = \mathbf{u}) \\ &\approx \frac{1}{C} \sum_{t=1}^{C} (P^t)_{\mathbf{uw}} \end{aligned} \tag{3}$$

where P^t denotes the t-step transition probability distribution which is the product of t one-step transition matrices P, i.e. $P^t = \underbrace{PP \cdots P}_{t}$. $(P^t)_{\mathbf{uw}} = P(S_t = \mathbf{w} \mid S_0 = \mathbf{u})$. $P(X|Y)$ is also a stochastic matrix where each row sums to one and represents the reachability distribution starting from a specific state.

The first approximation in (3) results from the fact that we reduce the near-infinite look-ahead steps to C steps. The second line holds when $0 < C \ll T$ (See Appendix A.6 for the proof) and suggests that when $0 < C \ll T$, one should use a larger C to achieve better approximation precision.

3. Asymmetric contrastive representation learning

We now aim to learn representations whose distance function encodes the asymmetric state reachability distribution $P(X|Y)$. In this section, we show how this problem can be formulated and addressed within the framework of noise contrastive estimation (NCE) (Gutmann & Hyvärinen, 2012). Based on different objective functions, NCE methods fall into two categories: binary NCE which discriminates between the positive and negative classes, and ranking NCE which ranks the true labels above the negative ones for the positive inputs (Ma & Collins, 2018). Although ranking NCE is broadly used in the recent resurgence of contrastive representation learning (van den Oord et al., 2019; Chen et al., 2020; Srinivas et al., 2020; He et al., 2020), we adopt binary NCE in this paper since it establishes a direct relationship between the scoring function and probability ratio.

3.1. Conditional binary NCE

Suppose that there is a data set of N trajectories drawn from a T-step Markov chain $\mathcal{M}$, i.e. $\mathcal{T} = \{(\mathbf{s}_0^i, \mathbf{s}_1^i, \ldots, \mathbf{s}_T^i)\}_{i=1}^N \sim \mathcal{M}$. We construct the positive data set for binary NCE as $D^+ = \{(\mathbf{x}_i^+, \mathbf{y}_i) \sim P(X, Y)\}_{i=1}^{N^+}$, and negative data set as $D^- = \{\mathbf{x}_i^- \sim P_n(X)\}_{i=1}^{N^-}$. $P(X, Y)$ denotes the aforementioned joint distribution of preceding random variable Y and subsequent random variable X. $P_n(X)$ denotes a specified negative distribution on state space $\mathcal{S}$. We use K to denote the ratio of negative and positive samples, i.e. $N^- = KN^+$.

This yields a binary classification setting where a classifier is trained to discriminate between positive samples from the conditional distribution $P(X|Y = \mathbf{y})$ and negative samples from $P_n(X)$, $\forall \mathbf{y}$. The training objective is to correctly classify both positive samples D^+ and negative samples D^- using logistic regression:

$$\begin{aligned} &\max_\theta J_{BI\_NCE}(\theta) \\ =&\max_\theta \mathbb{E}_{\mathbf{x}^+, \mathbf{y} \sim P(X,Y)} \log \sigma(f_\theta(\mathbf{x}^+, \mathbf{y})) \\ &+ K\mathbb{E}_{\mathbf{y} \sim P_Y(Y)} \mathbb{E}_{\mathbf{x}^- \sim P_n(X)} \log(1 - \sigma(f_\theta(\mathbf{x}^-, \mathbf{y}))) \end{aligned} \tag{4}$$

where $P_Y(Y)$ denotes the marginal distribution of Y. $\sigma(\cdot)$ denotes the logistic function.

When the classifier is Bayes-optimal, we have

$$\exp(f(X, Y = \mathbf{y})) = \frac{P(X|Y = \mathbf{y})}{KP_n(X)}, \quad \forall \mathbf{y} \tag{5}$$

where $f(X, Y = \mathbf{y})$ is the scoring function. (See proof in Appendix A.7)

3.2. Asymmetric encoders

Our method is built upon the key observation that the scoring function $f(X = \mathbf{x}, Y = \mathbf{y})$ in the conditional binary NCE can be treated as a similarity function between the embeddings of $\mathbf{x}$ and $\mathbf{y}$. Given that the underlying graph of a Markov chain is *directed*, the reachability from vertex $\mathbf{x}$ to vertex $\mathbf{y}$ often differs from the reachability from vertex $\mathbf{y}$ to vertex $\mathbf{x}$, i.e. $P(X = \mathbf{x}|Y = \mathbf{y}) \neq P(X = \mathbf{y}|Y = \mathbf{x})$. Therefore, having an asymmetric similarity function becomes essential to mirror the inherent asymmetry in the reachability probabilities. In other words, the function form of f should allow $f(X = \mathbf{x}, Y = \mathbf{y}) \neq f(X = \mathbf{y}, Y = \mathbf{x})$.

One effective approach to accomplish this is by utilizing two separate encoders. Concretely, we use an encoding function $\phi : \mathcal{S} \to \mathcal{Z}$ to map the preceding random variable Y to the embedding space Z and use another encoding function $\psi : \mathcal{S} \to \mathcal{Z}$ to map the subsequent random variable X to the same embedding space, i.e.

$$f(X = \mathbf{x}, Y = \mathbf{y}) = s(\psi(\mathbf{x}), \phi(\mathbf{y})) \tag{6}$$

$s(\cdot, \cdot)$ could be an arbitrary similarity measure in space Z, e.g. cosine similarity or negative l_2 distance. In this paper, we opt for cosine similarity as $s(\cdot, \cdot)$ because it is well-bounded and facilitates stable convergence, i.e. $s(\psi(\mathbf{x}), \phi(\mathbf{y})) = \frac{\psi(\mathbf{x})}{\|\psi(\mathbf{x})\|_2} \cdot \frac{\phi(\mathbf{y})}{\|\phi(\mathbf{y})\|_2}$.

Employing separate encoders offers two advantages over a single encoder. Firstly, it guarantees the asymmetry regardless of the choice of $s(\cdot, \cdot)$. Secondly, it ensures the asymmetry in a general sense without imposing constraints on the relationship between $\phi(\cdot)$ and $\psi(\cdot)$. On the contrary, prior works (van den Oord et al., 2019; Srinivas et al., 2020; Eysenbach et al., 2023) employ a single encoder $\psi(\cdot)$ alongside a linear transformation A to model time-series data, which can be viewed as a specific case of our approach where $\phi(\cdot) = A\psi(\cdot)$.

4. Reference state conditioned distance measure

Building upon the learned asymmetric similarity function $s(\cdot, \cdot)$, we can recover symmetric distance measures by conditioning on a reference state. We term it as a *reference state conditioned distance measure*. Formally, given a reference state $\mathbf{r}$, we define the pairwise latent distance between $\mathbf{u}$ and $\mathbf{v}$ with respect to $\mathbf{r}$ as $d_\mathbf{r}(\mathbf{u}, \mathbf{v})$:

- If $s(\phi(\mathbf{r}), \psi(\mathbf{u})) \geq s(\phi(\mathbf{r}), \psi(\mathbf{v}))$,

$$d_\mathbf{r}(\mathbf{u}, \mathbf{v}) = 1 - s(\phi(\mathbf{u}), \psi(\mathbf{v})) \tag{7}$$

- Otherwise,

$$d_\mathbf{r}(\mathbf{u}, \mathbf{v}) = 1 - s(\phi(\mathbf{v}), \psi(\mathbf{u})) \tag{8}$$

In other words, for a pair of states $\mathbf{u}$ and $\mathbf{v}$, we always choose the one closer to $\mathbf{r}$ as the starting state and choose the other one as the ending state to evaluate their distance. This criterion allows us to measure pairwise distance without any ambiguity. Unlike its undirected graph counterpart, there is no single and universal distance measure in a directed graph;

instead, the measure varies depending on the observer's perspective $\mathbf{r}$. This definition of a reference-dependent measure $d_{\mathbf{r}}(\cdot,\cdot)$ allows us to find the salient geometric structure of the representation space from changing perspectives, which is meaningful in many real-world applications.

5. Subgoal discovery

Subgoal is a notion in RL referring to the intermediate objectives or states that an agent aims to achieve on the way to reaching the ultimate goal, and are often used to break down complex or long-horizon tasks into shorter, simpler, more manageable parts.

In this work, we propose to use the concept of subgoals to denote the key states that only a handful of actions can lead through, which may change as the observer's perspective varies. Formally, we identify subgoals as the states that cause a sharp decrease in pairwise reachability, as perceived from the agent's current state. According to (5), we can convert the reachability defined in the original state space into the point density in the representation space. We then perform DBSCAN (Deng, 2020) in the representation space to identify the points in the low-density regions as subgoals.

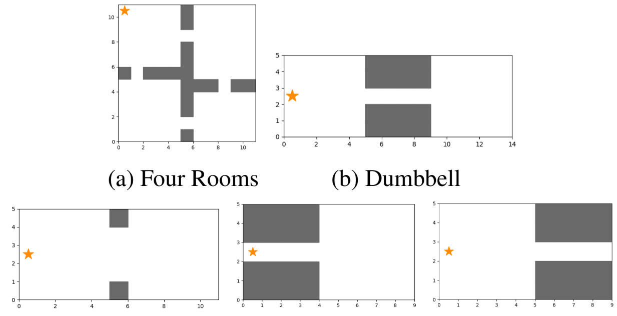
6. Experiments

We evaluate our representation learning method in five gridworld environments with various layouts: Four Rooms, Dumbbell, Wide door, Flask, and Nail as shown in Figure 1, which are designed to encompass basic geometric configurations. The states are 2D locations of the agent, $\mathbf{s} \in \mathbb{R}^2$. The action space includes five actions: left, right, up, down, and stop. In each environment, a data set of N trajectories $\mathcal{T} = \{(\mathbf{s}_0^i, \mathbf{s}_1^i, \ldots, \mathbf{s}_T^i)\}_{i=1}^N$ is collected by a uniform policy starting from a fixed initial state $\mathbf{s}_0$.

We train two encoders ϕ and ψ on $\mathcal{T}$ according to the learning objective (4). Both ϕ and ψ are parameterized as a 2-layer MLP with latent space $\mathbf{z} \in \mathbb{R}^{64}$. We set the ratio of negative and positive samples as $K = 1$ throughout all the experiments. We use the marginal distribution of X as the negative distribution for training, i.e. $P_n(X) = P_X(X)$, which we found performs the best.

6.1. Subgoal discovery results in Four Rooms

We now demonstrate that our learned distance measure $d_{\mathbf{r}}(\cdot,\cdot)$ enables easy identification of the subgoal states. We perform the DBSCAN clustering in the representation space according to $d_{\mathbf{r}}(\cdot,\cdot)$. Figure 2 shows the subgoals and clusters found by our algorithm in Four Rooms. The results in other environments and conditioning on more reference states can be found in Appendix A.3. It can be observed that the low-reachability regions such as doorways and passages have been successfully identified as subgoals and the rooms are decomposed into clusters.



(a) Four Rooms (b) Dumbbell

(c) Wide door (d) Flask (e) Nail

Figure 1: Gridworld environments: The grey areas indicate the walls. The yellow star indicates the initial state $\mathbf{s}_0$.

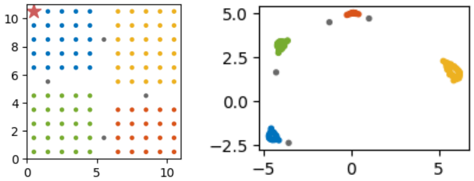


Figure 2: Subgoal discovery results in Four Rooms: The left shows the original states in 2D Euclidean space. The right shows the t-SNE projection of the learned representations. The states are colored according to the cluster labels in both two spaces. The red star indicates the reference state. The gray states are the subgoals found by our method. We set approximation step size $C = 16$, and train the encoders on a single episode of length $T = 153600$.

7. Conclusion

In this work, we study the problem of what is a good representation for planning and reasoning in a stochastic world and how to learn it. We discussed the importance of learning a distance measure in allowing planning and reasoning in the representation space. We modeled the world as a Markov chain and introduced the notion of C-step reachability on top of it to capture the geometric abstraction of the transition graph and allow multi-way probabilistic inference. We then showed how to embed the C-step abstraction of a Markov transition graph and encode the reachability into an asymmetric similarity function through conditional binary NCE. Based on this asymmetric similarity function, we developed a reference state conditioned distance measure, which enables the identification of geometrically salient states as subgoals. We demonstrated the effectiveness of the learned representations in subgoal discovery in the domain of gridworld.

References

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020.

Chen, Y., Bardes, A., Li, Z., and LeCun, Y. Bag of image patch embedding behind the success of self-supervised learning, 2023.

Deng, D. Dbscan clustering algorithm based on density. In *2020 7th international forum on electrical engineering and automation (IFEEA)*, pp. 949–953. IEEE, 2020.

Eysenbach, B., Salakhutdinov, R., and Levine, S. Search on the replay buffer: Bridging planning and reinforcement learning. *CoRR*, abs/1906.05253, 2019. URL http://arxiv.org/abs/1906.05253.

Eysenbach, B., Myers, V., Levine, S., and Salakhutdinov, R. Contrastive representations make planning easy. In *NeurIPS 2023 Workshop on Generalization in Planning*, 2023. URL https://openreview.net/forum?id=W0bhHvQK60.

Ghosh, D., Gupta, A., and Levine, S. Learning actionable representations with goal-conditioned policies. *arXiv preprint arXiv:1811.07819*, 2018.

Goyal, A., Islam, R., Strouse, D., Ahmed, Z., Botvinick, M., Larochelle, H., Bengio, Y., and Levine, S. Infobot: Transfer and exploration via the information bottleneck. *arXiv preprint arXiv:1901.10902*, 2019.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. *CoRR*, abs/1607.00653, 2016. URL http://arxiv.org/abs/1607.00653.

Gutmann, M. U. and Hyvärinen, A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2), 2012.

Gutmann, M. U. and Hyvärinen, A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(null):307–361, feb 2012. ISSN 1532-4435.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning, 2020.

Ma, Y. J., Yan, J., Jayaraman, D., and Bastani, O. How far i'll go: Offline goal-conditioned reinforcement learning via f-advantage regression. *arXiv preprint arXiv:2206.03023*, 2022.

Ma, Z. and Collins, M. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *CoRR*, abs/1809.01812, 2018. URL http://arxiv.org/abs/1809.01812.

McGovern, A. and Barto, A. G. Automatic discovery of subgoals in reinforcement learning using diverse density. *ICONIP 2008*, 2001.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

Nachum, O., Gu, S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning, 2018.

Srinivas, A., Laskin, M., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning, 2020.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding, 2019.

Wu, Y., Tucker, G., and Nachum, O. The laplacian in rl: Learning representations with efficient approximations, 2018.

Zhang, T., Eysenbach, B., Salakhutdinov, R., Levine, S., and Gonzalez, J. E. C-planning: An automatic curriculum for learning goal-reaching tasks. *CoRR*, abs/2110.12080, 2021. URL https://arxiv.org/abs/2110.12080.

A. Appendix

A.1. Preliminaries

A.1.1. Markov chain and the directed transition graph

A Markov chain on state space $\mathcal{S}$ can be thought of as a stochastic process traversing a directed graph where we start from vertex $\mathbf{s}_0 \sim \rho(S_0)$ and repeatedly follow an outgoing edge $\mathbf{s}_t \to \mathbf{s}_{t+1}$ with some probabilities. We denote the transition probability distribution of the Markov chain as $P(S_{t+1}|S_t)$ $(t = 0, 1, \dots)$. Thus, a Markov chain induces a weighted directed graph $G = (V, E, P)$ which is called the *transition graph*. $V = \{\mathbf{s} \in \mathcal{S}\}$ is the vertex set, and $E = \{\mathbf{s} \to \mathbf{s}' \mid \mathbf{s}, \mathbf{s}' \in \mathcal{S}\}$ is a set of directed edges where each edge $\mathbf{s} \to \mathbf{s}'$ has probability $P(S_{t+1} = \mathbf{s}'|S_t = \mathbf{s})$ as its weight. In this work, we consider the induced transition graph in the most general setting where loops are allowed, and a directed edge does not have to be paired with an inverse edge.

A.1.2. MDP and the environment graph

A Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho)$ is an extension of a Markov chain with the addition of actions and rewards. The induced transition graph G still has the states $\mathbf{s} \in \mathcal{S}$ as its vertex set, but the transition probability $P(S_{t+1} = \mathbf{s}'|S_t = \mathbf{s})$ from vertex $\mathbf{s}$ to its neighbor $\mathbf{s}'$ now involves a two-stage transition process: we move from $\mathbf{s}$ to $\mathbf{s}'$ through some actions $\mathbf{a}$ according to both the environment dynamics $P(S_{t+1}|S_t, A_t)$ and the agent's policy $\pi(A_t|S_t)$:

$$\begin{aligned} &P^{\pi}(S_{t+1} = \mathbf{s}'|S_t = \mathbf{s}) \\ =&\int P(S_{t+1} = \mathbf{s}'|S_t = \mathbf{s}, A_t = \mathbf{a}) \\ &\pi(A_t = \mathbf{a}|S_t = \mathbf{s})d\mathbf{a} \end{aligned} \tag{9}$$

where the policy probability $\pi(A_t = \mathbf{a}|S_t = \mathbf{s})$ acts as a weight of the environment transition probability $P(S_{t+1} = \mathbf{s}'|S_t = \mathbf{s}, A_t = \mathbf{a})$.

In particular, when the policy is a uniform distribution for any state $\mathbf{s}$, (9) is irrelevant to the policy π, i.e.

$$\begin{aligned} &P(S_{t+1} = \mathbf{s}'|S_t = \mathbf{s}) \\ =&\frac{1}{|\mathcal{A}|}\int P(S_{t+1} = \mathbf{s}'|S_t = \mathbf{s}, A_t = \mathbf{a})d\mathbf{a} \end{aligned} \tag{10}$$

We term this policy-agnostic transition graph as the *environment graph* since its edge weights encode the environment dynamics unbiasedly and the graph can be fully induced from an MDP. In addition, we ignore the rewards as they are task-specific and do not necessarily reflect the structure of the environment.

A.2. Related Work

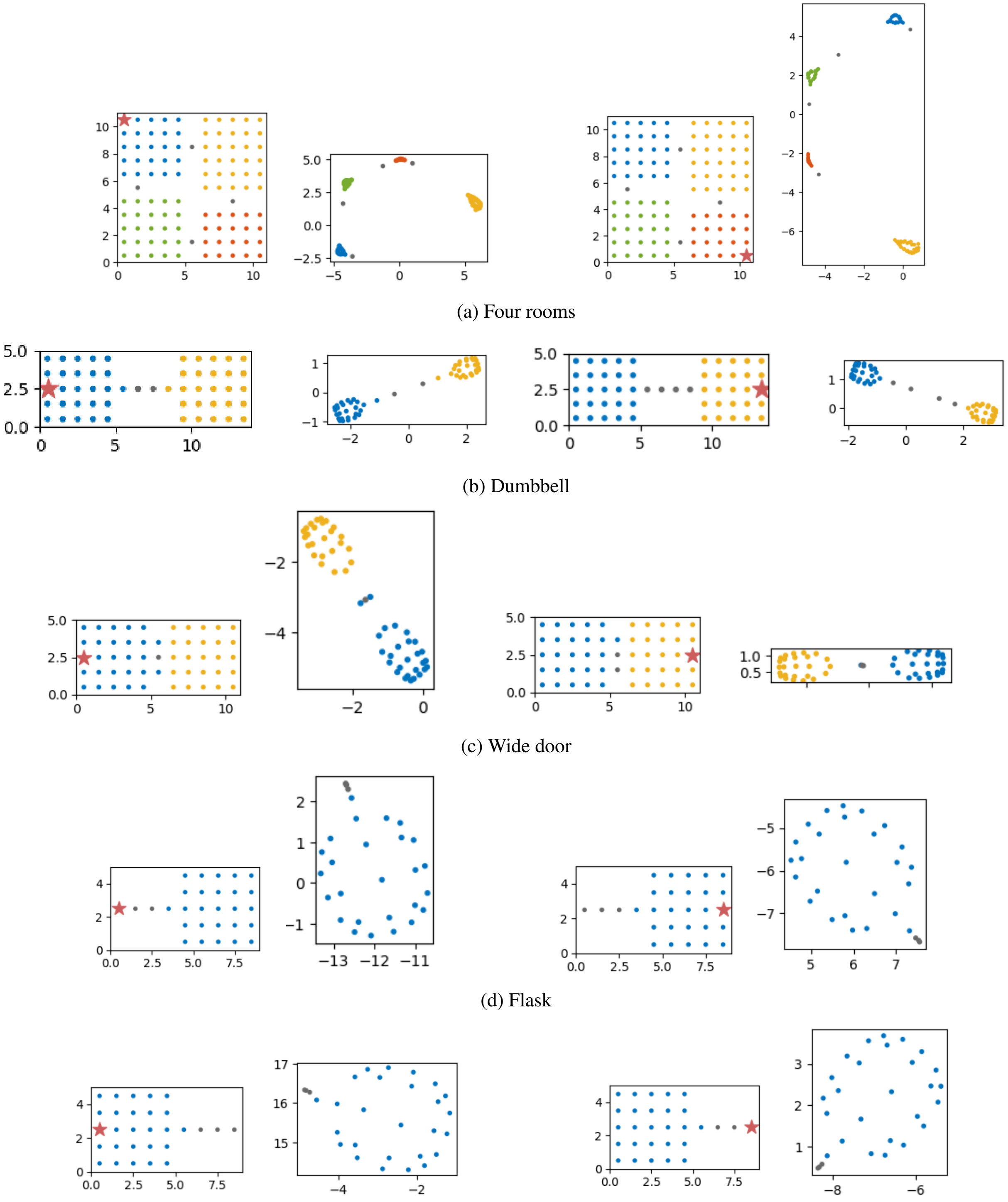
Subgoal discovery Subgoal is a fundamental concept introduced to tackle planning or decision-making problems using the divide-and-conquer strategy. While lacking a unified mathematical formulation, previous works have proposed various definitions of subgoals, each emphasizing its different roles in dividing the overall task, which include: midway states between the starting and goal state that are reachable by the current policy (Zhang et al., 2021; Eysenbach et al., 2019); common states shared by successful trajectories (McGovern & Barto, 2001; Ma et al., 2022); functionally salient states where policy distributions significantly change (Ghosh et al., 2018); and decision states where the goal state is informative to the decisions (Goyal et al., 2019). Different from the prior works, we define the subgoal as perspective-dependent geometrically salient states.

A.3. Subgoal discovery results for more environments

We show the subgoal discovery results in environments Four rooms, Dumbbell, Wide-door, Flask, and Nail in Figure 3.

A.4. t-SNE visualization of the learned representations

To examine the learned representations, we show the visualization of the original states and the learned representations in Figure 4. We use t-SNE to project the 64-dimensional learned representations onto 2D plots based on the distance



(a) Four rooms

(b) Dumbbell

(c) Wide door

(d) Flask

(e) Nail

Figure 3: Subgoal discovery results. The states are colored according to the cluster labels in both the original space and the learned representation space. The gray states are subgoals. In each environment, the clustering is performed based on two reference states, which are indicated by the red stars. In each case, the left subplot shows the original states in 2D Euclidean space, and the right subplot shows the t-SNE projection of the learned representations. In all the experiments, we set approximation step size $C = 16$, and train the encoders on a single episode of length $T = 153600$.

measure $d_{\mathbf{r}}(\cdot,\cdot)$ defined in (7) and (8). Across all the environments, the state space can be categorized into high-reachability regions such as rooms, and low-reachability regions such as long passages and various doorways (bottlenecks). Our learned representations can distinctly separate these regions according to their reachability with respect to the given reference state.

A.5. Ablation studies

A.5.1. The effect of approximation step size C

To better understand the effect of step size C in approximating the true reachability defined in (1), we compare the representations learned with $C = 1, 16, 64$ in the Four Rooms environment. When $C = 1$, the reachability is equivalent to the one-step transition probabilities. As the value of C increases, the C-step reachability considers more and more possible paths between two states and thus progressively approaches the true reachability. As a result, the abstraction level of the representations becomes higher. This trend has been successfully captured in the learned distance measure $d_{\mathbf{r}}(\cdot,\cdot)$. As shown in Figure 5, the rooms and doorways become further apart in the representation space as C goes up. This is because the reachability contrast becomes sharper as longer paths are considered. These empirical observations align well with our theoretical derivations on the influence of step size C in Section 2.2.

A.5.2. The choice of the negative distribution and the connection to PMI

Although the NCE framework holds for an arbitrary negative distribution $P_n(X)$, different choices of $P_n(X)$ affect the similarity function $f(X,Y)$ in a meaningful way. We experimented with several choices of the negative distribution $P_n(X)$: $P_X(\cdot)$ - the marginal distribution of X, $P_Y(\cdot)$ the marginal distribution of Y, and $U(X)$ - the uniform distribution of X in the Four Rooms environment in Figure 6 and 7. Among these options, $P_X(\cdot)$ is the only choice that consistently performs well across various values of chain length T and step size C. In contrast, $U(X)$ performs the worst in most cases. One hypothesis of why this phenomenon arises is that using a negative distribution resembling the positive distribution helps train the classifier to reach Bayes optimum. $P_X(X)$ is more similar to $P(X|Y=\mathbf{y})$ than $U(X)$.

It is worth noting that when $P_n(X) = P_X(X)$, the similarity function $f(X,Y)$ encodes the pointwise mutual information (PMI) of an ordered pair $(\mathbf{y},\mathbf{x})$ up to a constant offset.

$$\begin{aligned} f(X=\mathbf{x},Y=\mathbf{y}) &= \log\frac{P(X=\mathbf{x}|Y=\mathbf{y})}{KP_X(X=\mathbf{x})} \\ &= \log\frac{P(X=\mathbf{x}|Y=\mathbf{y})}{P_X(X=\mathbf{x})} + \log\frac{1}{K} \end{aligned} \tag{11}$$

where $\frac{1}{P_X(X=\mathbf{x})}$ can be viewed as a weight of the reachability $P(X=\mathbf{x}|Y=\mathbf{y})$, representing the inverse of the overall frequency of visits to the state $\mathbf{x}$. In other words, distribution $P(X=\mathbf{x}|Y=\mathbf{y})$ is adjusted according to the rarity of the arrival state $\mathbf{x}$. Especially, when $K = 1$ which is the default setting in this paper, the offset is removed from (11) which yields a direct relationship between the similarity function and PMI:

$$f(X=\mathbf{x},Y=\mathbf{y}) = \log\frac{P(X=\mathbf{x}|Y=\mathbf{y})}{P_X(X=\mathbf{x})} \tag{12}$$

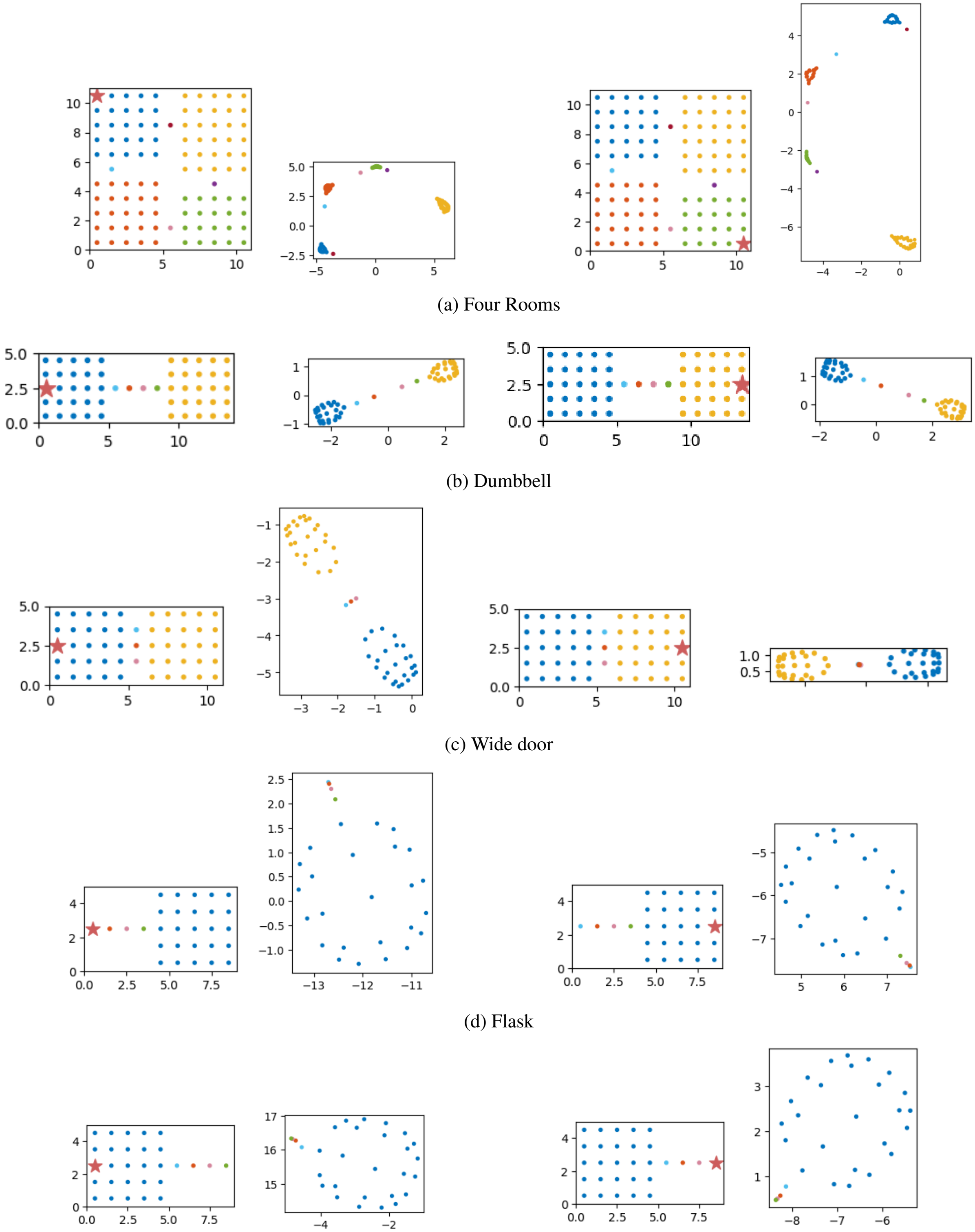
A.6. Proof of C-step approximation

Given a data set of N trajectories drawn from a T-step Markov chain M, i.e. $\mathcal{T} = \{(\mathbf{s}_0^i, \mathbf{s}_1^i, \dots, \mathbf{s}_T^i)\}_{i=1}^N \sim \mathcal{M}$. Let Y represent a preceding random variable in the chain $Y = \{S_i \mid 0 \le i \le T-1\}$ and X represent a random variable subsequent to Y within C time steps. $X = \{S_j \mid i < j \le \min(i+C, T)\}$. P denotes the one-step transition matrix. We now derive $P(X|Y)$ in terms of P as follows:

Based on the occurrences of the ordered pair $(\mathbf{y},\mathbf{x})$, we can derive the joint distribution of X and Y as

$$P(X=\mathbf{x},Y=\mathbf{y}) = \frac{\sum_{t=0}^{T-C}\sum_{i=t+1}^{t+c} n(S_i=\mathbf{x},S_t=\mathbf{y}) + \sum_{t=T-C+1}^{T-1}\sum_{i=t+1}^{T} n(S_i=\mathbf{x},S_t=\mathbf{y})}{N(T-C+1)C + N\sum_{t=1}^{C-1} t} \tag{13}$$

where $n(S_i=\mathbf{x},S_t=\mathbf{y})$ denotes the number of times $S_i=\mathbf{x}$ and $S_t=\mathbf{y}$ occur in data set $\mathcal{T}$.



(a) Four Rooms

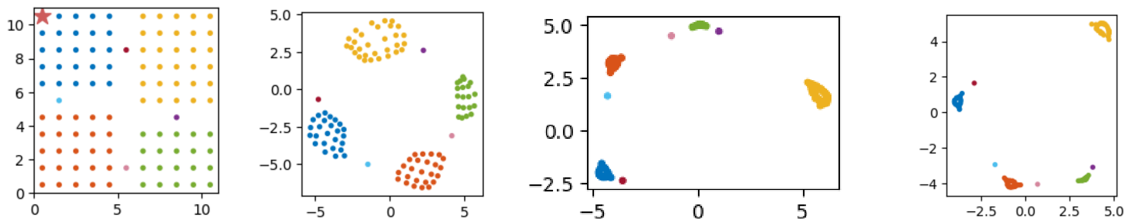
(b) Dumbbell

(c) Wide door

(d) Flask

(e) Nail

Figure 4: Visualization of the original states and the learned representations. The states are colored to visualize their position correspondences between two spaces. In each environment, we visualize the representation space from two different perspectives. The reference states $\mathbf{r}$ are indicated by the red stars. In each group, the left subplot shows the original states in 2D Euclidean space and the right subplot shows the t-SNE projection of the learned representations. In all the experiments, we set approximation step size $C = 16$, and train the encoders on a single episode of length $T = 153600$.



(a) The original states (b) $C = 1$ (c) $C = 16$ (d) $C = 64$

Figure 5: Visualization of the original states and the learned representations with different approximation step sizes C in Four Rooms environment. The embeddings are projected to 2D plots by t-SNE. In all the experiments, we train the encoders on a single episode of length $T = 153600$.

Similarly, by counting the occurrences of each specific state $\mathbf{y}$, the marginal distribution of Y can be derived as

$$P_Y(Y = \mathbf{y}) = \frac{C\sum_{t=0}^{T-C} n(S_t = \mathbf{y}) + \sum_{t=T-C+1}^{T-1}(T-t)n(S_t = \mathbf{y})}{N(T-C+1)C + N\sum_{t=1}^{C-1} t} \tag{14}$$

where $n(S_t = \mathbf{y})$ denotes the number of times $S_t = \mathbf{y}$ occurs in data set $\mathcal{T}$.

By the definition of the Markov chain, we have

$$n(S_i = \mathbf{x}, S_t = \mathbf{y}) = n(S_t = \mathbf{y})(P^{i-t})_{\mathbf{yx}} \quad (i = t+1, t+2, \dots) \tag{15}$$

Plugging (15) into (13), we have

$$P(X = \mathbf{x}, Y = \mathbf{y}) = \frac{\sum_{t=0}^{T-C} n(S_t = \mathbf{y}) \sum_{i=1}^{C} (P^i)_{\mathbf{yx}} + \sum_{t=T-C+1}^{T-1} \sum_{i=t+1}^{T} n(S_t = \mathbf{y})(P^{i-t})_{\mathbf{yx}}}{N(T-C+1)C + N\sum_{t=1}^{C-1} t} \tag{16}$$

Given (16) and (14), we can derive the conditional distribution as

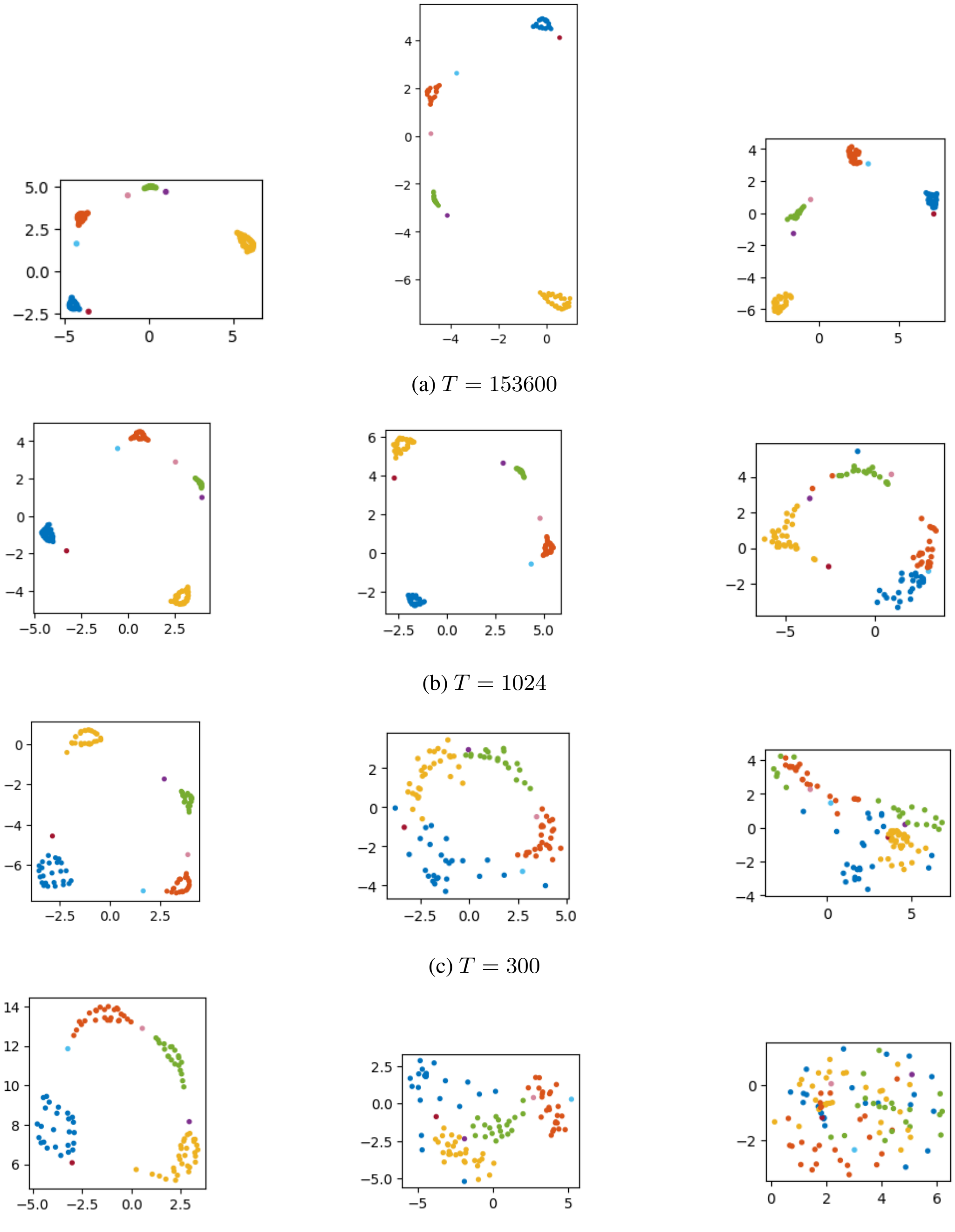
$$\begin{aligned} P(X = \mathbf{x} \mid Y = \mathbf{y}) &= \frac{P(X = \mathbf{x}, Y = \mathbf{y})}{P(Y = \mathbf{y})} \\ &= \frac{\sum_{t=0}^{T-C} n(S_t = \mathbf{y}) \sum_{i=1}^{C} (P^i)_{\mathbf{yx}} + \sum_{t=T-C+1}^{T-1} \sum_{i=t+1}^{T} n(S_t = \mathbf{y})(P^{i-t})_{\mathbf{yx}}}{C\sum_{t=0}^{T-C} n(S_t = \mathbf{y}) + \sum_{t=T-C+1}^{T-1}(T-t)n(S_t = \mathbf{y})} \end{aligned} \tag{17}$$

When $0 < C \ll T$, we can further drop the second term of the nominator and denominator, which is equivalent to changing the range of Y to $Y = \{S_i \mid 0 \le i \le T - C\}$. That is,

$$\begin{aligned} P(X = \mathbf{x} \mid Y = \mathbf{y}) &\approx \frac{\sum_{t=0}^{T-C} n(S_t = \mathbf{y}) \sum_{i=1}^{C} (P^i)_{\mathbf{yx}}}{C\sum_{t=0}^{T-C} n(S_t = \mathbf{y})} \\ &= \frac{1}{C}\sum_{t=1}^{C} (P^t)_{\mathbf{yx}} \end{aligned} \tag{18}$$

A.7. Proof of Bayes optimal scoring function

Suppose that we are training a binary classifier $P(C|X, Y = \mathbf{y}; \theta)$ to discriminate between positive data distribution $P(X|Y = \mathbf{y})$ and negative data distribution $P_n(X)$, $\forall \mathbf{y}$. The ratio of negative and positive samples is K. The Bayes



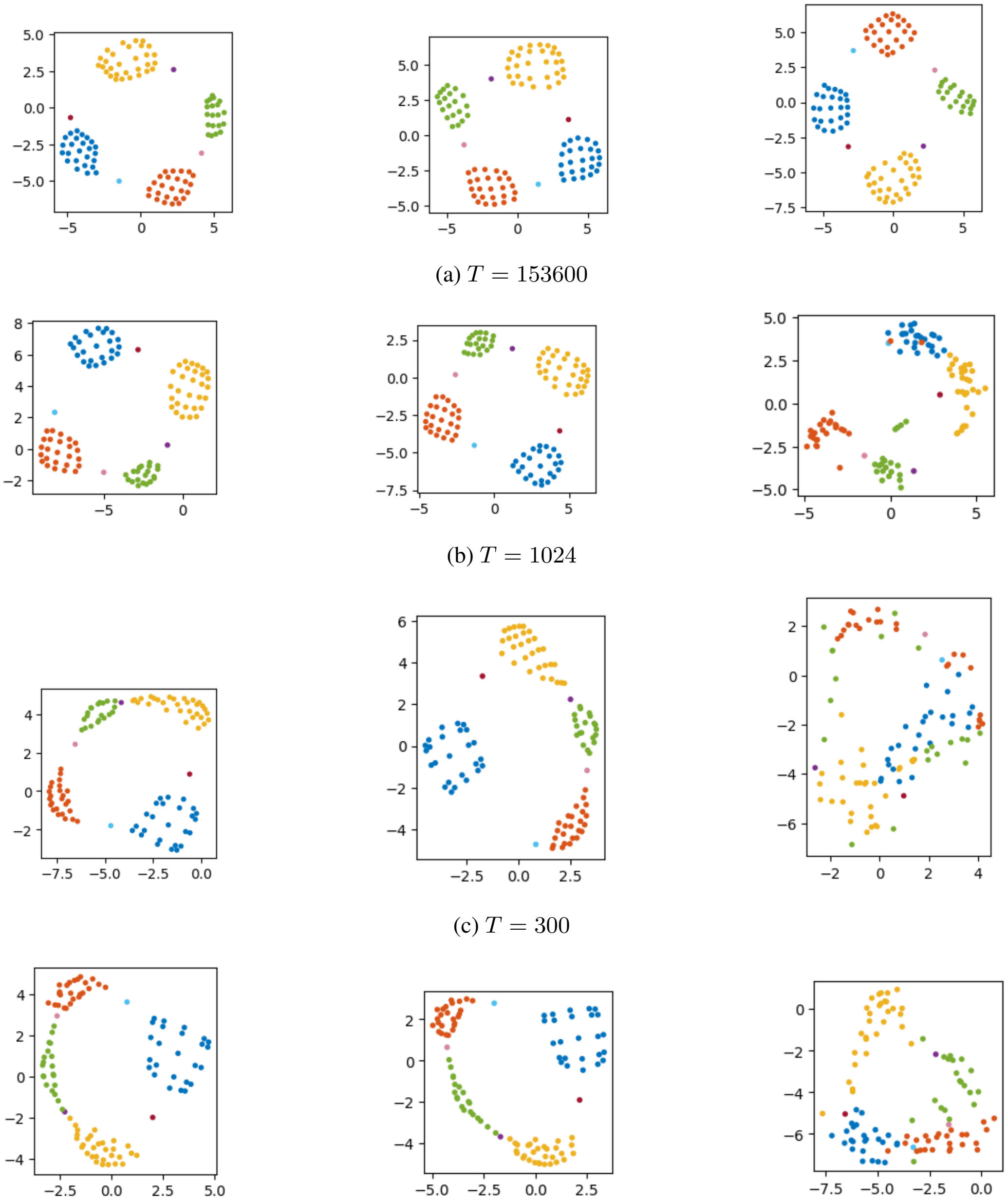
(a) $T = 153600$

(b) $T = 1024$

(c) $T = 300$

(d) $T = 150$

Figure 6: Learned representations with different negative distributions in Four Rooms environment when $C = 16$. Left column: $P_n(X) = P_X(X)$, Middle column: $P_n(X) = P_Y(X)$, Right column: $P_n(X) = U(X)$. Each row corresponds to the results with a different episode length T. All experiments have the same number of training environment steps 153600.



(a) $T = 153600$

(b) $T = 1024$

(c) $T = 300$

(d) $T = 150$

Figure 7: Learned representations with different negative distributions in Four Rooms environment when $C = 1$. Left column: $P_n(X) = P_X(X)$, Middle column: $P_n(X) = P_Y(X)$, Right column: $P_n(X) = U(X)$. Each row corresponds to the results with a different episode length T. All experiments have the same number of training environment steps 153600.

optimal classifier $P(C|X, Y = \mathbf{y}; \theta)$ is a maximum a posteriori (MAP) hypothesis satisfying

$$\begin{aligned}\frac{P(X|Y=\mathbf{y})}{P_n(X)} &= \frac{P(X|Y=\mathbf{y}, C=1)}{P(X|C=0)} \\ &= \frac{P(C=1|X, Y=\mathbf{y}; \theta)}{P(C=0|X, Y=\mathbf{y}; \theta)} \frac{P(C=0)}{P(C=1)} \\ &= \frac{P(C=1|X, Y=\mathbf{y}; \theta)}{P(C=0|X, Y=\mathbf{y}; \theta)} K \\ &= \frac{P(C=1|X, Y=\mathbf{y}; \theta)}{1-P(C=1|X, Y=\mathbf{y}; \theta)} K, \quad \forall \mathbf{y}\end{aligned} \tag{19}$$

$$P(C=1|X, Y=\mathbf{y}; \theta) = \frac{P(X|Y=\mathbf{y})}{P(X|Y=\mathbf{y}) + KP_n(X)}, \quad \forall \mathbf{y} \tag{20}$$

When the classifier is a logistic function, we have

$$P(C=1|X, Y=\mathbf{y}; \theta) = \frac{1}{1+\exp(-f_\theta(X, Y=\mathbf{y}))}, \quad \forall \mathbf{y} \tag{21}$$

Plugging (21) into (20), we have

$$\exp(f_\theta(X, Y=\mathbf{y})) = \frac{P(X|Y=\mathbf{y})}{KP_n(X)}, \quad \forall \mathbf{y} \tag{22}$$

A.8. A graph perspective of the single encoder embedding

With slight adjustments, our method can also *treat the directed graph as an undirected graph*. In such cases, we modify the range of random variable Y and X to $Y = \{S_i \mid 0 \leq i \leq T\}$ and $X = \{S_j \mid \max(i-C, 0) \leq j \leq \min(i+C, T), j \neq i\}$. Additionally, we use a single encoder $\phi(\cdot)$ to encode both X and Y. $s(\cdot, \cdot)$ could be any symmetric similarity measure, e.g. cosine similarity. As a result, we have

$$\begin{aligned}&f(X=\mathbf{x}, Y=\mathbf{y}) \\ =&f(X=\mathbf{y}, Y=\mathbf{x}) \\ =&s(\phi(\mathbf{x}), \phi(\mathbf{y})) \\ =&\log\frac{P(X=\mathbf{x}|Y=\mathbf{y}) + P(X=\mathbf{y}|Y=\mathbf{x})}{2KP_n(X=\mathbf{x})}\end{aligned} \tag{23}$$

We transform the directed graph into an undirected one by enforcing the similarity function $f(\cdot, \cdot)$ to encode the average of $P(X=\mathbf{x}|Y=\mathbf{y})$ and $P(X=\mathbf{y}|Y=\mathbf{x})$. Note that the reachability from $\mathbf{x}$ to $\mathbf{y}$ and the reachability from $\mathbf{y}$ to $\mathbf{x}$ may not be identical, meaning $P(X=\mathbf{x}|Y=\mathbf{y})$ may not be equal to $P(X=\mathbf{y}|Y=\mathbf{x})$.

In fact, word embedding methods such as word2vec (Mikolov et al., 2013) and graph embedding methods such as node2vec (Grover & Leskovec, 2016) have implicitly performed the operations in (23) through the Skip-gram model. Learning undirected representations is sufficient when the downstream tasks are clustering, classification, etc. These applications rely on mutual similarity or compatibility measures insensible to the directionality. However, incorporating transition direction into representation learning is crucial for planning, reasoning, and event modeling. This becomes especially important when the transition is irreversible due to temporal order or causalities. For example, consider two states: $\mathbf{y}$=young and $\mathbf{x}$=old, transiting from young to old is certain while the reverse is impossible. In probabilistic language, we can express it as $P(X=\mathbf{x}|Y=\mathbf{y}) = 1$ and $P(X=\mathbf{y}|Y=\mathbf{x}) = 0$. The expected embeddings of these two states should correctly reflect this difference in reachability. Ideally, the state 'young' should be very close to the state 'old' in one scenario and be infinitely far away in the other. Averaging these two probabilities into 0.5 would erroneously pull them together to the same distance in both cases.

A.9. Inference and planning using the learned asymmetric similarity function

After training $\phi(\cdot)$, $\psi(\cdot)$, we can use them to perform inference tasks on a given set of states χ. For instance, suppose that we are currently at state $\mathbf{s}$ and want to know which state is most likely to occur in the future. We can then identify it by finding the state closest to $\mathbf{s}$ in the latent space, i.e. solving $\arg\max_{\mathbf{x}\in\chi} s(\phi(\mathbf{s}), \psi(\mathbf{x}))$. Furthermore, with a defined action space, we can construct a directed graph on χ by evaluating $s(\phi(\cdot), \psi(\cdot))$. Feeding G to any search algorithm, such as Dijkstra's algorithm, enables us to plan a shortest path from an initial state $\mathbf{s}_0$ to a goal state $\mathbf{s}_g$ in the latent space.

A.10. Future work

We leave the following topics for future work: (1) Use the discovered subgoals in HRL setting to solve long-horizon tasks and improve sample efficiency. (2) Use the learned similarity function as an intrinsic reward function to avoid tedious reward engineering.