Contrastive Difference Predictive Coding

Chongyi Zheng
Carnegie Mellon University
chongyiz@andrew.cmu.edu

Ruslan Salakhutdinov
Carnegie Mellon University

Benjamin Eysenbach
Princeton University

Abstract

Predicting and reasoning about the future lie at the heart of many time-series questions. For example, goal-conditioned reinforcement learning can be viewed as learning representations to predict which states are likely to be visited in the future. While prior methods have used contrastive predictive coding to model time series data, learning representations that encode long-term dependencies usually requires large amounts of data. In this paper, we introduce a temporal difference version of contrastive predictive coding that stitches together pieces of different time series data to decrease the amount of data required to learn predictions of future events. We apply this representation learning method to derive an off-policy algorithm for goal-conditioned RL. Experiments demonstrate that, compared with prior RL methods, ours achieves $2\times$ median improvement in success rates and can better cope with stochastic environments. In tabular settings, we show that our method is about $20\times$ more sample efficient than the successor representation and $1500\times$ more sample efficient than the standard (Monte Carlo) version of contrastive predictive coding.

Code: `https://github.com/chongyi-zheng/td_infonce`
Website: `https://chongyi-zheng.github.io/td_infonce`

1 Introduction

Learning representations is important for modeling high-dimensional time series data. Many applications of time-series modeling require representations that not only contain information about the contents of a particular observation, but also about how one observation relates to others that co-occur in time. Acquiring representations that encode temporal information is challenging, especially when attempting to capture long-term temporal dynamics: the frequency of long-term events may decrease with the time scale, meaning that learning longer-horizon dependencies requires larger quantities of data.

In this paper, we study contrastive representation learning on time series data – positive examples co-occur nearby in time, so the distances between learned representations should encode the likelihood of transiting from one representation to another. Building on prior work that uses the InfoNCE [79, 67] loss to

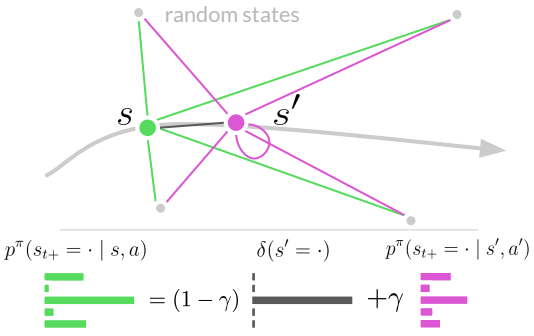


Figure 1: **TD InfoNCE** is a nonparametric version of the successor representation. *(Top)* The distances between learned representations indicate the probability of transitioning to a set of randomly-sampled states. *(Bottom)* We update these representations so they assign high likelihood to *(a)* the next state and *(b)* states likely to be visited after the next state. See Sec. 2 for details.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

learn representations of time-series data effectively, we will aim to build a temporal difference version of this loss. Doing so may allow us to optimize this objective with fewer samples, may enable us to stitch together pieces of different time series data, and may enable us to perform counterfactual reasoning – we should be able to estimate which representations we would have learned, if we had collected data in a different way. After a careful derivation, our resulting method can be interpreted as a non-parametric form of the successor representation [15], as shown in Fig. 1.

The main contribution of this paper is a temporal difference estimator for InfoNCE. We then apply this estimator to develop a new algorithm for goal-conditioned RL. Experiments on both state-based and image-based benchmarks show that our algorithm outperforms prior methods, especially on the most challenging tasks. Additional experiments demonstrate that our method can handle stochasticity in the environment more effectively than prior methods. We also demonstrate that our algorithm can be effectively applied in the offline setting. Additional tabular experiments demonstrate that TD InfoNCE is up to $1500\times$ more sample efficient than the standard Monte Carlo version of the loss and that it can effectively stitch together pieces of data.

2 Temporal Difference InfoNCE

In this section, we derive a new loss for estimating the discounted state occupancy measure for a fixed policy. This loss will be a temporal difference variant of the InfoNCE loss. We will use **temporal difference InfoNCE (TD InfoNCE)** to refer to our loss function. See Appendix B for the complete derivation.

In the off-policy setting, we aim to estimate the discounted state occupancy measure of the policy π given a dataset of transitions $\mathcal{D} = \{(s, a, s')_i\}_{i=1}^D$ collected by another behavioral policy $\beta(a \mid s)$. This setting is challenging because we do not obtain samples from the discounted state occupancy measure of the target policy π. Addressing this challenge involves two steps: *(i)* expanding the MC estimator (Eq. 10) via the recursive relationship of the discounted state occupancy measure (Eq. 8), and *(ii)* estimating the expectation over the discounted state occupancy measure via importance sampling. We first use the identity from Eq. 8 to express the MC InfoNCE loss as the sum of a next-state term and a future-state term:

$$\mathbb{E}_{\substack{(s,a)\sim p(s,a)\\ s_{t+}^{(2:N)}\sim p(s_{t+})}}\Bigg[(1-\gamma)\underbrace{\mathbb{E}_{s_{t+}^{(1)}\sim p(s'|s,a)}\left[\log\frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}}\right]}_{\mathcal{L}_1(f)} + \gamma\underbrace{\mathbb{E}_{\substack{s'\sim p(s'|s,a),a'\sim\pi(a'|s')\\ s_{t+}^{(1)}\sim p^\pi(s_{t+}|s',a')}}\left[\log\frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}}\right]}_{\mathcal{L}_2(f)}\Bigg].$$

While this estimate is similar to a TD target for Q-Learning [91, 28], the second term requires sampling from the discounted state occupancy measure of policy π. To avoid this sampling, we next replace the expectation over $p^\pi(s_{t+} \mid s', a')$ in $\mathcal{L}_2(f)$ by an importance weight,

$$\mathcal{L}_2(f) = \mathbb{E}_{\substack{s'\sim p(s'|s,a),a'\sim\pi(a'|s')\\ s_{t+}^{(1)}\sim p(s_{t+})}}\left[\frac{p^\pi(s_{t+}^{(1)} \mid s', a')}{p(s_{t+}^{(1)})}\log\frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}}\right].$$

If we could estimate the importance weight, then we could easily estimate this term by sampling from $p(s_{t+})$. We will estimate this importance weight by rearranging the expression for the optimal critic (Eq. 11) and substituting our estimate for the normalizing constant $c(s, a)$ (Eq. 6):

$$\frac{p^\pi(s_{t+}^{(1)} \mid s, a)}{p(s_{t+}^{(1)})} = c(s,a)\cdot\exp\left(f^\star(s,a,s_{t+}^{(1)})\right) = \frac{e^{f^\star(s,a,s_{t+}^{(1)})}}{\mathbb{E}_{p(s_{t+})}\left[e^{f^\star(s,a,s_{t+})}\right]}. \tag{1}$$

We will use $w(s, a, s_{t+}^{(1:N)})$ to denote our estimate of this, using f in place of $f^\star$ and using a finite-sample estimate of the expectation in the denominator:

$$w(s,a,s_{t+}^{(1:N)}) \triangleq \frac{e^{f(s,a,s_{t+}^{(1)})}}{\frac{1}{N}\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}} \tag{2}$$

This weight accounts for the effect of the discounted state occupancy measure of the target policy. Additionally, it corresponds to the categorical classifier that InfoNCE produces (without constant N). Taken together, we can now substitute the importance weight in $\mathcal{L}_2(f)$ with our estimate in Eq. 2, yielding a temporal difference (TD) InfoNCE estimator

$$\mathcal{L}_{\text{TD InfoNCE}}(f) \triangleq \mathbb{E}_{\substack{(s,a)\sim p(s,a) \\ s_{t+}^{(2:N)}\sim p(s_{t+})}} \left[(1-\gamma)\mathbb{E}_{s_{t+}^{(1)}\sim p(s'|s,a)} \left[\log \frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}} \right] \right.$$
$$\left. + \gamma \mathbb{E}_{\substack{s'\sim p(s'|s,a) \\ a'\sim\pi(a'|s') \\ s_{t+}^{(1)}\sim p(s_{t+})}} \left[\lfloor w(s',a',s_{t+}^{(1:N)}) \rfloor_{\text{sg}} \log \frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}} \right] \right], \quad (3)$$

where $\lfloor \cdot \rfloor_{\text{sg}}$ indicates the gradient of the importance weight should not affect the gradient of the entire objective. As shown in Fig. 1, we can interpret the first term as pulling together the representations of the current state-action pair $\phi(s,a)$ and the next state $\psi(s')$; the second term pulls the representations at the current step $\phi(s,a)$ similar to the (weighted) predictions from the future state $\psi(s_{t+})$. Importantly, the TD InfoNCE estimator is equivalent to the MC InfoNCE estimator for the optimal critic function: $\mathcal{L}_{\text{TD InfoNCE}}(f^\star) = \mathcal{L}_{\text{MC InfoNCE}}(f^\star)$.

Goal-conditioned policy learning. The TD InfoNCE method provides a way for estimating the discounted state occupancy measure. This section shows how this estimator can be used to derive a new algorithm for goal-conditioned RL. This algorithm will alternate between *(1)* estimating the occupancy measure using the TD InfoNCE objective and *(2)* optimizing the policy to maximize the likelihood of the desired goal under the estimated occupancy measure. Pseudo-code is shown in Algorithm 1, additional details are in Appendix B.3, and code is available online.[1]

While our TD InfoNCE loss in Sec. B.2 estimates the discounted state occupancy measure for policy $\pi(a \mid s)$, we can extend it to the goal-conditioned setting by replacing $\pi(a \mid s)$ with $\pi(a \mid s, g)$ and $f(s,a,s_{t+})$ with $f(s,a,g,s_{t+})$, resulting in a goal-conditioned TD InfoNCE estimator. This goal-conditioned TD InfoNCE objective estimates the discounted state occupancy measure of *any* future state for a goal-conditioned policy commanding *any* goal. Recalling that the discounted state occupancy measure corresponds to the Q function [24], the policy objective is to select actions that maximize the likelihood of the commanded goal:

$$\mathbb{E}_{\substack{p_g(g),p_0(s) \\ \pi(a_0|s,g)}} \left[\log p^\pi(s_{t+} = g \mid s,a,g) \right] = \mathbb{E}_{\substack{g\sim p_g(g), s\sim p_0(s) \\ a_0\sim\pi(a|s,g), s_{t+}^{(1:N)}\sim p(s_{t+})}} \left[\log \frac{e^{f^\star(s,a,g,s_{t+}=g)}}{\sum_{i=1}^N e^{f^\star(s,a,g,s_{t+}^{(i)})}} \right]. \quad (4)$$

In practice, we optimize both the critic function and the policy for one gradient step iteratively, using our estimated f in place of $f^\star$.

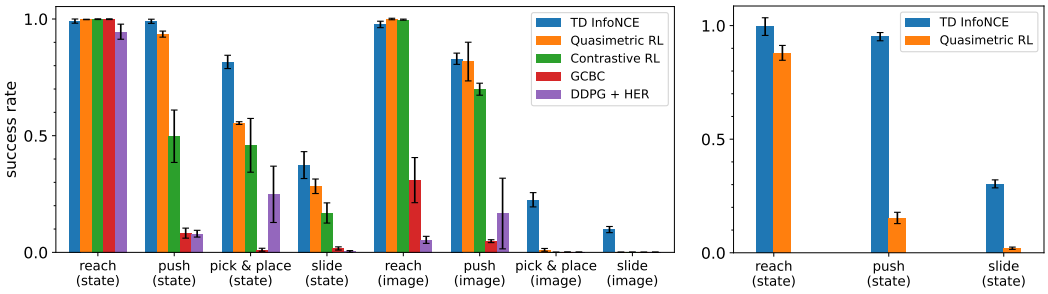
3 Experiments

We ran a wide range of experiments comparing goal-conditioned TD InfoNCE to prior goal-conditioned RL approaches on both online and offline GCRL benchmarks. Our findings include:

Evaluation on online GCRL benchmarks. TD InfoNCE matches or outperforms baselines on all Fetch manipulation tasks [69], both for state and image observations (Fig. 2a and Appendix Fig. 5). On those more challenging tasks (`pick & place (state / image)` and `slide (state / image)`), TD InfoNCE achieves a $2\times$ median improvement relative to the strongest baseline. On the most challenging tasks, image-based `pick & place` and `slide`, TD InfoNCE is the only method achieving non-negligible success rates. See Appendix F.1 for details.

Handling stochastic environments. TD InfoNCE continues achieving high success rates in environments with stochastic noise, while the performance of QRL decreases significantly (Appendix Fig. 2b). See Appendix F.1 for details.

[1] `https://github.com/chongyi-zheng/td_infonce`



(a) Fetch robotics benchmark from [69]

(b) Stochastic tasks.

Figure 2: **Evaluation on online GCRL benchmarks.** *(Left)* TD InfoNCE performs similarly to or outperforms all baselines on both state-based and image-based tasks. *(Right)* On stochastic versions of the state-based tasks, TD InfoNCE outperforms the strongest baseline (QRL). Appendix Fig. 5 shows the learning curves.

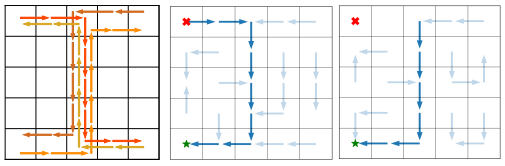


Figure 3: **Stitching trajectories in a dataset.** The behavioral policy collects "Z" style trajectories. Unlike the Monte Carlo method (contrastive RL) , our TD InfoNCE successfully "stitches" these trajectories together, navigating between pairs of (start ✖, goal ★) states unseen in the training trajectories. Appendix Fig. 7 shows additional examples.

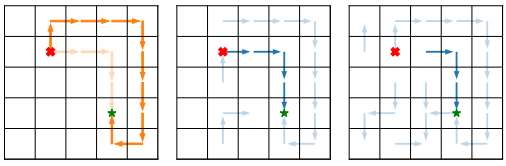


Figure 4: **Searching for shortcuts in skewed datasets.** *(Left)* Conditioned on different initial states ✖ and goals ★, we collect datasets with 95% long paths (dark) and 5% short paths (light). *(Center)* TD InfoNCE infers the shortest path, *(Right)* while contrastive RL fails to find this path. Appendix Fig. 8 shows additional examples.

Evaluation on offline D4RL benchmark. We also study whether the good performance of TD InfoNCE transfers to the setting without any interaction with the environment (i.e., offline RL). We evaluate on AntMaze tasks from the D4RL benchmark [27]. The results in Appendix Table 1 show that TD InfoNCE outperforms most baselines on most tasks. See Appendix F.2 for details.

Accuracy of the estimated discounted state occupancy measure. TD methods achieve lower errors than the Monte Carlo method, while TD InfoNCE converges faster than C-Learning (Appendix Fig. 6). Additional, TD InfoNCE is $1500\times$ more sample efficient (6.5×10^3 vs 10^7 transitions) than its Monte Carlo counterpart. Compared with the only other TD method applicable in continuous settings (C-learning), TD InfoNCE can achieve a comparable loss with $130\times$ less data (7.7×10^4 vs 10^7 transitions). Even compared with the strongest baseline (successor representations), TD InfoNCE can achieve a comparable error rate with almost $20\times$ fewer samples (5.2×10^5 vs 10^7 transitions). See Appendix F.3 for details.

Off-policy reasoning. We next study whether the resulting goal-conditioned policy is capable of performing dynamic programming with offline data, comparing TD InfoNCE to contrastive RL (i.e., Monte Carlo InfoNCE). Fig. 3 shows that TD InfoNCE successfully stitches together pieces of different trajectories to find a route between unseen (state, goal) pairs. Fig. 4 shows that TD InfoNCE can perform off-policy reasoning, finding a path that is shorter than the average path demonstrated in the dataset. See Appendix F.4 for details.

Representation interpolation. Prior work has shown that representations from self-supervised learning can reflect the geometry of the underlying data [88, 3]. We study this property for the representations learned by TD InfoNCE, interpolating between the learned representations of 29-dimensional observations from the offline AntMaze `medium-play-v2` task. Results in Appendix G.3 suggest that the learned representations are structured so that linear interpolation corresponds to planning a path from one state to another.

References

[1] Agarwal, A., Jiang, N., Kakade, S. M., and Sun, W. (2019). Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32.

[2] Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. (2017). Hindsight experience replay. *Advances in neural information processing systems*, 30.

[3] Arora, S., Khandeparkar, H., Khodak, M., Plevrakis, O., and Saunshi, N. (2019). A theoretical analysis of contrastive unsupervised representation learning. In *36th International Conference on Machine Learning, ICML 2019*, pages 9904–9923. International Machine Learning Society (IMLS).

[4] Barreto, A., Borsa, D., Hou, S., Comanici, G., Aygün, E., Hamel, P., Toyama, D., Mourad, S., Silver, D., Precup, D., et al. (2019). The option keyboard: Combining skills in reinforcement learning. *Advances in Neural Information Processing Systems*, 32.

[5] Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., and Silver, D. (2017). Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30.

[6] Bertsekas, D. P. and Tsitsiklis, J. N. (1995). Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE conference on decision and control*, volume 1, pages 560–564. IEEE.

[7] Blier, L., Tallec, C., and Ollivier, Y. (2021). Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*.

[8] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., et al. (2018). Jax: composable transformations of python+ numpy programs.

[9] Campos, V., Trott, A., Xiong, C., Socher, R., Giró-i Nieto, X., and Torres, J. (2020). Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, pages 1317–1327. PMLR.

[10] Chane-Sane, E., Schmid, C., and Laptev, I. (2021). Goal-conditioned reinforcement learning with imagined subgoals. In *International Conference on Machine Learning*, pages 1430–1440. PMLR.

[11] Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097.

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

[13] Choi, J., Sharma, A., Lee, H., Levine, S., and Gu, S. S. (2021). Variational empowerment as representation learning for goal-conditioned reinforcement learning. In *International Conference on Machine Learning*, pages 1953–1963. PMLR.

[14] Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

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

[16] Ding, Y., Florensa, C., Abbeel, P., and Phielipp, M. (2019). Goal-conditioned imitation learning. *Advances in neural information processing systems*, 32.

[17] Dubi, A. and Horowitz, Y. (1979). The interpretation of conditional monte carlo as a form of importance sampling. *SIAM Journal on Applied Mathematics*, 36(1):115–122.

[18] Durugkar, I., Tec, M., Niekum, S., and Stone, P. (2021). Adversarial intrinsic motivation for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:8622–8636.

[19] Emmons, S., Eysenbach, B., Kostrikov, I., and Levine, S. (2021). Rvs: What is essential for offline rl via supervised learning? In *International Conference on Learning Representations*.

[20] Ernst, D., Geurts, P., and Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6.

[21] Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. (2018). Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*.

[22] Eysenbach, B., Salakhutdinov, R., and Levine, S. (2019). Search on the replay buffer: Bridging planning and reinforcement learning. *Advances in Neural Information Processing Systems*, 32.

[23] Eysenbach, B., Salakhutdinov, R., and Levine, S. (2020). C-learning: Learning to achieve goals via recursive classification. *arXiv preprint arXiv:2011.08909*.

[24] Eysenbach, B., Zhang, T., Levine, S., and Salakhutdinov, R. R. (2022). Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620.

[25] Fang, K., Yin, P., Nair, A., and Levine, S. (2022). Planning to practice: Efficient online fine-tuning by composing goals in latent space. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4076–4083. IEEE.

[26] Fang, K., Yin, P., Nair, A., Walke, H. R., Yan, G., and Levine, S. (2023). Generalization with lossy affordances: Leveraging broad offline data for learning visuomotor tasks. In *Conference on Robot Learning*, pages 106–117. PMLR.

[27] Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. (2020). D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*.

[28] Fu, J., Kumar, A., Soh, M., and Levine, S. (2019). Diagnosing bottlenecks in deep q-learning algorithms. In *International Conference on Machine Learning*, pages 2021–2030. PMLR.

[29] Fujimoto, S. and Gu, S. S. (2021). A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145.

[30] Gao, T., Yao, X., and Chen, D. (2021). Simcse: Simple contrastive learning of sentence embeddings. In *2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics (ACL).

[31] Gershman, S. J. (2018). The successor representation: its computational logic and neural substrates. *Journal of Neuroscience*, 38(33):7193–7200.

[32] Ghosh, D., Gupta, A., Reddy, A., Fu, J., Devin, C. M., Eysenbach, B., and Levine, S. (2020). Learning to reach goals via iterated supervised learning. In *International Conference on Learning Representations*.

[33] Giles, M. B. (2015). Multilevel monte carlo methods. *Acta numerica*, 24:259–328.

[34] Gregor, K., Rezende, D. J., and Wierstra, D. (2016). Variational intrinsic control. *arXiv preprint arXiv:1611.07507*.

[35] Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al. (2020). Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284.

[36] Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. (2020). Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Conference on Robot Learning*, pages 1025–1037. PMLR.

[37] Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

[38] Hammersley, J. (1956). Conditional monte carlo. *Journal of the ACM (JACM)*, 3(2):73–76.

[39] Hansen-Estruch, P., Zhang, A., Nair, A., Yin, P., and Levine, S. (2022). Bisimulation makes analogies in goal-conditioned reinforcement learning. In *International Conference on Machine Learning*, pages 8407–8426. PMLR.

[40] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

[41] Henaff, O. (2020). Data-efficient image recognition with contrastive predictive coding. In *International conference on machine learning*, pages 4182–4192. PMLR.

[42] Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.

[43] Janner, M., Mordatch, I., and Levine, S. (2020). gamma-models: Generative temporal difference learning for infinite-horizon prediction. *Advances in Neural Information Processing Systems*, 33:1724–1735.

[44] Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

[45] Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

[46] Kostrikov, I., Nair, A., and Levine, S. (2021). Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*.

[47] Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. (2019). Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32.

[48] Kumar, A., Zhou, A., Tucker, G., and Levine, S. (2020). Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191.

[49] Laskin, M., Lee, K., Stooke, A., Pinto, L., Abbeel, P., and Srinivas, A. (2020a). Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895.

[50] Laskin, M., Srinivas, A., and Abbeel, P. (2020b). Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR.

[51] Levy, A., Konidaris, G., Platt, R., and Saenko, K. (2018). Learning multi-level hierarchies with hindsight. In *International Conference on Learning Representations*.

[52] Linsker, R. (1988). Self-organization in a perceptual network. *Computer*, 21(3):105–117.

[53] Logeswaran, L. and Lee, H. (2018). An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

[54] Lynch, C., Khansari, M., Xiao, T., Kumar, V., Tompson, J., Levine, S., and Sermanet, P. (2020). Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR.

[55] Ma, Y. J., Sodhani, S., Jayaraman, D., Bastani, O., Kumar, V., and Zhang, A. (2022). Vip: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*.

[56] Ma, Z. and Collins, M. (2018). Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *arXiv preprint arXiv:1809.01812*.

[57] Mazoure, B., Eysenbach, B., Nachum, O., and Tompson, J. (2022). Contrastive value learning: Implicit models for simple offline rl. *arXiv preprint arXiv:2211.02100*.

[58] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

[59] Nachum, O., Gu, S. S., Lee, H., and Levine, S. (2018). Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31.

[60] Nair, A., Bahl, S., Khazatsky, A., Pong, V., Berseth, G., and Levine, S. (2020a). Contextual imagined goals for self-supervised robotic learning. In *Conference on Robot Learning*, pages 530–539. PMLR.

[61] Nair, A. V., Pong, V., Dalal, M., Bahl, S., Lin, S., and Levine, S. (2018). Visual reinforcement learning with imagined goals. *Advances in neural information processing systems*, 31.

[62] Nair, S. and Finn, C. (2019). Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. In *International Conference on Learning Representations*.

[63] Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. (2022). R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*.

[64] Nair, S., Savarese, S., and Finn, C. (2020b). Goal-aware prediction: Learning to model what matters. In *International Conference on Machine Learning*, pages 7207–7219. PMLR.

[65] Oh, J., Guo, Y., Singh, S., and Lee, H. (2018). Self-imitation learning. In *International Conference on Machine Learning*, pages 3878–3887. PMLR.

[66] Oh Song, H., Xiang, Y., Jegelka, S., and Savarese, S. (2016). Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012.

[67] Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

[68] Pertsch, K., Rybkin, O., Ebert, F., Zhou, S., Jayaraman, D., Finn, C., and Levine, S. (2020). Long-horizon visual planning with goal-conditioned hierarchical predictors. *Advances in Neural Information Processing Systems*, 33:17321–17333.

[69] Plappert, M., Andrychowicz, M., Ray, A., McGrew, B., Baker, B., Powell, G., Schneider, J., Tobin, J., Chociej, M., Welinder, P., et al. (2018). Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*.

[70] Poole, B., Ozair, S., Van Den Oord, A., Alemi, A., and Tucker, G. (2019). On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR.

[71] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

[72] Rainforth, T., Cornish, R., Yang, H., Warrington, A., and Wood, F. (2018). On nesting monte carlo estimators. In *International Conference on Machine Learning*, pages 4267–4276. PMLR.

[73] Riedmiller, M., Hafner, R., Lampe, T., Neunert, M., Degrave, J., Wiele, T., Mnih, V., Heess, N., and Springenberg, J. T. (2018). Learning by playing solving sparse reward tasks from scratch. In *International conference on machine learning*, pages 4344–4353. PMLR.

[74] Rudner, T. G., Pong, V., McAllister, R., Gal, Y., and Levine, S. (2021). Outcome-driven reinforcement learning via variational inference. *Advances in Neural Information Processing Systems*, 34:13045–13058.

[75] Schaul, T., Horgan, D., Gregor, K., and Silver, D. (2015). Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR.

[76] Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

[77] Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S., and Brain, G. (2018). Time-contrastive networks: Self-supervised learning from video. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1134–1141. IEEE.

[78] Shah, D., Eysenbach, B., Rhinehart, N., and Levine, S. (2022). Rapid exploration for open-world navigation with latent goal models. In *Conference on Robot Learning*, pages 674–684. PMLR.

[79] Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

[80] Srivastava, R. K., Shyam, P., Mutz, F., Jaśkowski, W., and Schmidhuber, J. (2019). Training agents using upside-down reinforcement learning. *arXiv preprint arXiv:1912.02877*.

[81] Sun, H., Li, Z., Liu, X., Zhou, B., and Lin, D. (2019). Policy continuation with hindsight inverse dynamics. *Advances in Neural Information Processing Systems*, 32.

[82] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[83] Tian, S., Nair, S., Ebert, F., Dasari, S., Eysenbach, B., Finn, C., and Levine, S. (2020a). Model-based visual planning with self-supervised functional distances. In *International Conference on Learning Representations*.

[84] Tian, Y., Krishnan, D., and Isola, P. (2020b). Contrastive multiview coding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 776–794. Springer.

[85] Touati, A. and Ollivier, Y. (2021). Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23.

[86] Tsai, Y.-H. H., Zhao, H., Yamada, M., Morency, L.-P., and Salakhutdinov, R. R. (2020). Neural methods for point-wise dependency estimation. *Advances in Neural Information Processing Systems*, 33:62–72.

[87] Tschannen, M., Djolonga, J., Rubenstein, P. K., Gelly, S., and Lucic, M. (2019). On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*.

[88] Wang, T. and Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR.

[89] Wang, T., Torralba, A., Isola, P., and Zhang, A. (2023). Optimal goal-reaching reinforcement learning via quasimetric learning. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 36411–36430. PMLR.

[90] Warde-Farley, D., Van de Wiele, T., Kulkarni, T., Ionescu, C., Hansen, S., and Mnih, V. (2018). Unsupervised control through non-parametric discriminative rewards. *arXiv preprint arXiv:1811.11359*.

[91] Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine learning*, 8:279–292.

[92] Weinberger, K. Q. and Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2).

[93] Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742.

[94] Zhang, S., Liu, B., and Whiteson, S. (2020). Gradientdice: Rethinking generalized offline estimation of stationary values. In *International Conference on Machine Learning*, pages 11194–11203. PMLR.

[95] Zheng, C., Eysenbach, B., Walke, H., Yin, P., Fang, K., Salakhutdinov, R., and Levine, S. (2023). Stabilizing contrastive rl: Techniques for offline goal reaching. *arXiv preprint arXiv:2306.03346*.

A Related Work

This paper will study the problem of self-supervised RL, building upon prior methods on goal-condition RL, contrastive representation learning, and methods for predicting future state visitations. Our analysis will draw a connection between these prior methods, a connection which will ultimately result in a new algorithm for goal-conditioned RL. We discuss connections with unsupervised skill learning and mutual information in Appendix D.

Goal-conditioned reinforcement learning. Prior work has proposed many frameworks for learning goal-conditioned policies, including conditional supervised learning [16, 32, 36, 19, 54, 65, 81], actor-critic methods [2, 59, 10], semi-parametric planning [68, 25, 26, 22, 62, 36], and distance metric learning [89, 83, 64, 18]. These methods have demonstrated impressive results on a range of tasks, including real-world robotic tasks [55, 78, 95]. While some methods require manually-specified reward functions or distance functions, our work builds upon a self-supervised interpretation of goal-conditioned RL that casts this problem as predicting which states are likely to be visited in the future [23, 24, 7].

Contrastive representation learning. Contrastive learning methods have become a key tool for learning representations in computer vision and NLP [14, 76, 79, 66, 88, 67, 87, 92, 40, 71, 12, 84, 30]. These methods assign similar representations to positive examples and dissimilar representations to negative examples or outdated embeddings [35]. The two main contrastive losses are based on binary classification ("NCE") ranking loss ("InfoNCE") [56]. Modern contrastive learning methods typically employ the ranking-based objective to learn representations of images [12, 84, 41, 93], text [53, 44, 71] and sequential data [63, 77]. Prior works have also provided theoretical analysis for these methods from the perspective of mutual information maximization [52, 70], noise contrastive estimation [37, 56, 86, 3], and the geometry of the learned representations [88]. In the realm of RL, prior works have demonstrated that contrastive methods can provide effective reward functions and auxiliary learning objectives [50, 49, 39, 13, 60, 61], and can also be used to formulate the goal-reaching problem in an entirely self-supervised manner [55, 18, 23, 24]. Our method will extend these results by building a temporal difference version of the "ranking"-based contrastive loss; this loss will enable us to use data from one policy to estimate which states a different policy will visit.

Temporal difference learning and successor representation. Another line of work studies using temporal difference learning to predict states visited in the future, building upon successor representations and successor features [15, 5, 4, 7]. While learning successor representation using temporal difference bears a similarity to the typical Q-Learning algorithm [91, 28, 58] in the tabular setting, directly estimating this quantity is difficult with continuous states and actions [43, 5, 85, 7]. To lift this limitation, we will follow prior work [24, 23, 85] in predicting the successor representation indirectly: rather than learning a representation whose coordinates correspond to visitation probabilities, we will learn state representations such that their inner product corresponds to a visitation probability. Unlike prior methods, we will show how the common InfoNCE objective can be estimated in a temporal difference fashion, opening the door to off-policy reasoning and enabling our method to reuse historical data to improve data efficiency.

B Derivation of TD InfoNCE

We start by introducing notation and prior approaches to the contrastive representation learning and the goal-conditioned RL problems. We then propose a new self-supervised actor-critic algorithm that we will use in our analysis.

B.1 Preliminaries

We first review prior work in contrastive representation learning and goal-conditioned RL. Our method will use ideas from both.

Contrastive representation via InfoNCE. Contrastive representation learning aims to learn a representation space, pushing representations of positive examples together and pushing representations of negative examples away. InfoNCE (also known as contrastive predictive coding) [79, 45, 67, 41]

is a widely used contrastive loss, which builds upon noise contrastive estimation (NCE) [37, 56]. Given the distribution of data $p_{\mathcal{X}}(x), p_{\mathcal{Y}}(y)$ over data $x \in \mathcal{X}, y \in \mathcal{Y}$ and the conditional distribution of positive pairs $p_{\mathcal{Y}|\mathcal{X}}(y|x)$ over $\mathcal{X} \times \mathcal{Y}$, InfoNCE loss is defined as

$$\mathcal{L}_{\text{InfoNCE}}(f) \triangleq \mathbb{E}_{\substack{x \sim p_{\mathcal{X}}(x), y^{(1)} \sim p_{\mathcal{Y}|\mathcal{X}}(y|x) \\ y^{(2:N)} \sim p_{\mathcal{Y}}(y)}} \left[\log \frac{e^{f(x, y^{(1)})}}{\sum_{i=1}^{N} e^{f(x, y^{(i)})}} \right], \tag{5}$$

where $f : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ is a parametric function. Following prior work [24, 88, 85], we choose to parameterize $f(\cdot, \cdot)$ via the inner product of representations of data $f(x, y) = \phi(x)^\top \psi(y)$, where $\phi(\cdot)$ and $\psi(\cdot)$ map data to ℓ_2 normalized vectors of dimension d. We will call f the *critic function* and ϕ and ψ the *contrastive representations*. The Bayes-optimal critic for the InfoNCE loss satisfies [70, 56, 67]

$$\exp\left(f^\star(x, y)\right) = \frac{p(y \mid x)}{p(y)c(x)},$$

where $c(\cdot)$ is an arbitrary function. We can estimate this arbitrary function using the optimal critic $f^\star$ by sampling multiple negative pairs from the data distribution:

$$\mathbb{E}_{p(y)}\left[\exp\left(f^\star(x, y)\right)\right] = \int \cancel{p(y)} \frac{p(y \mid x)}{\cancel{p(y)} c(x)} dy = \frac{1}{c(x)} \underbrace{\int p(y \mid x) dy}_{=1} = \frac{1}{c(x)}. \tag{6}$$

Reinforcement learning and goal-conditioned RL. We will consider a Markov decision process defined by states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, rewards $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$. Using $\Delta(\cdot)$ denotes the probability simplex, we define an initial state distribution $p_0 : \mathcal{S} \mapsto \Delta(\mathcal{S})$, discount factor $\gamma \in (0, 1]$, and dynamics $p : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$. Given a policy $\pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$, we will use $p_t^\pi(s_{t+} \mid s, a)$ to denote the probability density of reaching state s_{t+} after exactly t steps, starting at state s and action a and then following the policy $\pi(a \mid s)$. We can then define the discounted state occupancy measure [42, 94, 23, 24, 95] starting from state s and action a as

$$p^\pi(s_{t+} \mid s, a) \triangleq (1 - \gamma) \sum_{t=1}^{\infty} \gamma^{t-1} p_t^\pi(s_{t+} \mid s, a). \tag{7}$$

Prior work [15] have shown that this discounted state occupancy measure follows a recursive relationship between the density at the current time step and the future time steps:

$$p^\pi(s_{t+} \mid s, a) = (1 - \gamma) p(s' = s_{t+} \mid s, a) + \gamma \mathbb{E}_{\substack{s' \sim p(s'|s,a) \\ a' \sim \pi(a'|s')}} \left[p^\pi(s_{t+} \mid s', a') \right]. \tag{8}$$

For goal-conditioned RL, we define goals $g \in \mathcal{S}$ in the same space as states and consider a goal-conditioned policy $\pi(a \mid s, g)$ and the corresponding goal-conditioned discounted state occupancy measure $p^\pi(s_{t+} \mid s, a, g)$. For evaluation, we will sample goals from a distribution $p_g : \mathcal{S} \mapsto \Delta(\mathcal{S})$. Following prior work [23, 74], we define the objective of the goal-reaching policy as maximizing the probability of reaching desired goals under its discounted state occupancy measure while commanding the same goals:

$$\max_{\pi(\cdot|\cdot,\cdot)} \mathbb{E}_{p_g(g), p_0(s), \pi(a|s,g)} \left[p^\pi(s_{t+} = g \mid s, a, g) \right]. \tag{9}$$

In tabular settings, this objective is the same as maximizing expected returns using a sparse reward function $r(s, a, s', g) = (1 - \gamma)\delta(s' = g)$ [24]. Below, we review two strategies for estimating the discounted state occupancy measure. Our proposed method (Sec. B.2) will combine the strengths of these methods while lifting their respective limitations.

Contrastive RL and C-Learning. Our focus will be on using contrastive representation learning to build a new goal-conditioned RL algorithm, following a template set in prior work [24, 23]. These *contrastive RL* methods are closely related to the successor representation [15]: they aim to learn representations whose inner products correspond to the likelihoods of reaching future states. Like the successor representation, representations from these contrastive RL methods can then be used to represent the Q function for any reward function [57]. Prior work [24] has shown how both NCE and the InfoNCE losses can be used to derive Monte Carlo algorithms for estimating the

discounted state occupancy measure. We review the Monte Carlo InfoNCE loss below. Given a policy $\pi(a \mid s)$, consider learning contrastive representations for a state and action pair $x = (s, a)$ and a potential future state $y = s_{t+}$. We define the data distribution to be the joint distribution of state-action pairs $p_{\mathcal{X}}(x) = p(s, a)$ and the marginal distribution of future states $p_{\mathcal{Y}}(y) = p(s_{t+})$, representing either the distribution of a replay buffer (online) or the distribution of a dataset (offline). The conditional distribution of positive pairs is set to the discounted state occupancy measure for policy π, $p_{\mathcal{Y}|\mathcal{X}}(y \mid x) = p^\pi(s_{t+} \mid s, a)$, resulting in a Monte Carlo (MC) estimator

$$\mathcal{L}_{\text{MC InfoNCE}}(f) = \mathbb{E}_{\substack{(s,a)\sim p(s,a), s_{t+}^{(1)} \sim p^\pi(s_{t+}|s,a) \\ s_{t+}^{(2:N)} \sim p(s_{t+})}} \left[\log \frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}} \right] \tag{10}$$

and an optimal critic function satisfying

$$\exp(f^\star(s, a, s_{t+})) = \frac{p^\pi(s_{t+} \mid s, a)}{p(s_{t+})c(s, a)}. \tag{11}$$

This loss estimates the discounted state occupancy measure in a Monte Carlo manner. While conceptually simple, computing this estimator requires sampling future states from the discounted state occupancy measure of the policy π, i.e., on-policy data. Such an estimate is potentially sample inefficient because collecting samples for different policies is expensive. That is, we cannot share experiences collected by one policy with the learning of the discounted state occupancy measure of another policy.

In the same way that temporal difference (TD) algorithms tend to be more sample efficient than Monte Carlo algorithms for reward maximization [82], we expect that TD contrastive methods are more sample efficient at estimating probability ratios than their Monte Carlo counterparts. Given that the InfoNCE tends to outperform the NCE objective in other machine learning disciplines, we conjecture that our TD InfoNCE objective will outperform the TD NCE objective [23] (see experiments in Sec. 3).

B.2 Temporal Difference InfoNCE

In this section, we derive a new loss for estimating the discounted state occupancy measure for a fixed policy. This loss will be a temporal difference variant of the InfoNCE loss. We will use **temporal difference InfoNCE (TD InfoNCE)** to refer to our loss function.

In the off-policy setting, we aim to estimate the discounted state occupancy measure of the policy π given a dataset of transitions $\mathcal{D} = \{(s, a, s')_i\}_{i=1}^D$ collected by another behavioral policy $\beta(a \mid s)$. This setting is challenging because we do not obtain samples from the discounted state occupancy measure of the target policy π. Addressing this challenge involves two steps: *(i)* expanding the MC estimator (Eq. 10) via the recursive relationship of the discounted state occupancy measure (Eq. 8), and *(ii)* estimating the expectation over the discounted state occupancy measure via importance sampling. We first use the identity from Eq. 8 to express the MC InfoNCE loss as the sum of a next-state term and a future-state term:

$$\mathbb{E}_{\substack{(s,a)\sim p(s,a) \\ s_{t+}^{(2:N)} \sim p(s_{t+})}} \Bigg[(1-\gamma) \underbrace{\mathbb{E}_{s_{t+}^{(1)} \sim p(s'|s,a)} \left[\log \frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}} \right]}_{\mathcal{L}_1(f)} + \gamma \underbrace{\mathbb{E}_{\substack{s'\sim p(s'|s,a), a'\sim\pi(a'|s') \\ s_{t+}^{(1)} \sim p^\pi(s_{t+}|s',a')}} \left[\log \frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}} \right]}_{\mathcal{L}_2(f)} \Bigg].$$

While this estimate is similar to a TD target for Q-Learning [91, 28], the second term requires sampling from the discounted state occupancy measure of policy π. To avoid this sampling, we next replace the expectation over $p^\pi(s_{t+} \mid s', a')$ in $\mathcal{L}_2(f)$ by an importance weight,

$$\mathcal{L}_2(f) = \mathbb{E}_{\substack{s'\sim p(s'|s,a), a'\sim\pi(a'|s') \\ s_{t+}^{(1)} \sim p(s_{t+})}} \left[\frac{p^\pi(s_{t+}^{(1)} \mid s', a')}{p(s_{t+}^{(1)})} \log \frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}} \right].$$

Algorithm 1 Temporal Difference InfoNCE

Input contrastive representations ϕ_θ and ψ_θ, target representations $\phi_{\bar{\theta}}$ and $\psi_{\bar{\theta}}$, and goal-conditioned policy π_ω.
for each iteration **do**
  Sample $\{(s_t^{(i)}, a_t^{(i)}, s_{t+1}^{(i)}, g^{(i)}, s_{t+}^{(i)})\}_{i=1}^N \sim$ replay buffer / dataset, $a^{(i)} \sim \pi(a \mid s_t^{(i)}, g^{(i)})$.
  Compute $F_{\text{next}}, F_{\text{future}}, F_{\text{goal}}$ using ϕ_θ and ψ_θ.
  Compute $\bar{F}_w$ using $\phi_{\bar{\theta}}$ and $\psi_{\bar{\theta}}$.
  $W \leftarrow N \cdot \texttt{stop\_grad}\left(\textsc{SoftMax}(\bar{F}_w)\right)$
  $\mathcal{L}(\theta) \leftarrow (1-\gamma)\mathcal{CE}(\text{logits} = F_{\text{next}}, \text{labels} = I_N) + \gamma\mathcal{CE}(\text{logits} = F_{\text{future}}, \text{labels} = W)$
  $\mathcal{L}(\omega) \leftarrow \mathcal{CE}(\text{logits} = F_{\text{goal}}, \text{labels} = I_N)$
  Update θ, ω by taking gradients of $\mathcal{L}(\theta), \mathcal{L}(\omega)$.
  Update $\bar{\theta}$ using an exponential moving average.
Return ϕ_θ, ψ_θ, and π_ω.

If we could estimate the importance weight, then we could easily estimate this term by sampling from $p(s_{t+})$. We will estimate this importance weight by rearranging the expression for the optimal critic (Eq. 11) and substituting our estimate for the normalizing constant $c(s,a)$ (Eq. 6):

$$\frac{p^\pi(s_{t+}^{(1)} \mid s,a)}{p(s_{t+}^{(1)})} = c(s,a) \cdot \exp\left(f^\star(s,a,s_{t+}^{(1)})\right) = \frac{e^{f^\star(s,a,s_{t+}^{(1)})}}{\mathbb{E}_{p(s_{t+})}\left[e^{f^\star(s,a,s_{t+})}\right]}. \tag{12}$$

We will use $w(s,a,s_{t+}^{(1:N)})$ to denote our estimate of this, using f in place of $f^\star$ and using a finite-sample estimate of the expectation in the denominator:

$$w(s,a,s_{t+}^{(1:N)}) \triangleq \frac{e^{f(s,a,s_{t+}^{(1)})}}{\frac{1}{N}\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}}$$

This weight accounts for the effect of the discounted state occupancy measure of the target policy. Additionally, it corresponds to the categorical classifier that InfoNCE produces (without constant N). Taken together, we can now substitute the importance weight in $\mathcal{L}_2(f)$ with our estimate in Eq. 2, yielding a temporal difference (TD) InfoNCE estimator

$$\mathcal{L}_{\text{TD InfoNCE}}(f) \triangleq \mathbb{E}_{\substack{(s,a)\sim p(s,a) \\ s_{t+}^{(2:N)} \sim p(s_{t+})}} \left[(1-\gamma)\mathbb{E}_{s_{t+}^{(1)} \sim p(s' \mid s,a)} \left[\log \frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}} \right] \right.$$
$$\left. + \gamma \mathbb{E}_{\substack{s' \sim p(s' \mid s,a) \\ a' \sim \pi(a' \mid s') \\ s_{t+}^{(1)} \sim p(s_{t+})}} \left[\lfloor w(s', a', s_{t+}^{(1:N)}) \rfloor_{\text{sg}} \log \frac{e^{f(s,a,s_{t+}^{(1)})}}{\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}} \right] \right],$$

where $\lfloor \cdot \rfloor_{\text{sg}}$ indicates the gradient of the importance weight should not affect the gradient of the entire objective. As shown in Fig. 1, we can interpret the first term as pulling together the representations of the current state-action pair $\phi(s,a)$ and the next state $\psi(s')$; the second term pulls the representations at the current step $\phi(s,a)$ similar to the (weighted) predictions from the future state $\psi(s_{t+})$. Importantly, the TD InfoNCE estimator is equivalent to the MC InfoNCE estimator for the optimal critic function: $\mathcal{L}_{\text{TD InfoNCE}}(f^\star) = \mathcal{L}_{\text{MC InfoNCE}}(f^\star)$.

Convergence and connections. In Appendix C, we prove that optimizing a variant of the TD InfoNCE objective is equivalent to perform one step policy evaluation with a new Bellman operator; thus, repeatedly optimizing this objective yields the correct discounted state occupancy measure. This analysis considers the tabular setting and assumes that the denominators of the softmax functions and w in Eq. 3 are computed using an exact expectation. We discuss the differences between TD InfoNCE and C-learning [23] (a temporal difference estimator of the NCE objective) in Appendix G.2. Appendix E discusses how TD InfoNCE corresponds to a nonparametric variant of the successor representation.

B.3 The Complete Algorithm for Goal-Conditioned RL

The complete algorithm of TD InfoNCE (Algorithm 1) alters between estimating the discounted state occupancy measure of the current goal-conditioned policy via contrastive learning (Eq. 3)

and updating the policy using the actor loss (Eq. 4), while collecting more data. Given a batch of N transitions of $\{(s_t^{(i)}, a_t^{(i)}, s_{t+1}^{(i)}, g^{(i)}, s_{t+}^{(i)})\}_{i=1}^N$ sampled from $p(s_t, a_t, g)$, $p(s_{t+1} \mid s_t, a_t)$, and $p(s_{t+})$, we can first compute the critic function for different combinations of goal-conditioned state-action pairs and future states by computing their contrastive representations $\phi(s_t, a_t, g)$, $\psi(s_{t+})$, and $\psi(s_{t+})$, and then construct two critic matrices $F_{\text{next}}, F_{\text{future}} \in \mathbb{R}^{N \times N}$ using those representations:

$$F_{\text{next}}[i, j] = \phi(s_t^{(i)}, a_t^{(i)}, g^{(i)})^\top \psi(s_{t+1}^{(j)}), F_{\text{future}}[i, j] = \phi(s_t^{(i)}, a_t^{(i)}, g^{(i)})^\top \psi(s_{t+}^{(j)})$$

Note that the inner product parameterization of the critic function $f(s_t, a_t, g, s_{t+}) = \phi(s_t, a_t, g)^\top \psi(s_{t+})$ helps compute these matrices efficiently. Using these critic matrices, we rewrite the TD InfoNCE estimate as a sum of two cross entropy losses. The first cross entropy loss involves predicting which of the N next states $s_{t+1}^{(1:N)}$ is the correct next state for the corresponding goal-conditioned state and action pair:

$$(1 - \gamma)\mathcal{CE}(\text{logits} = F_{\text{next}}, \text{labels} = I_N),$$

where $\mathcal{CE}(\text{logits} = F_{\text{next}}, \text{labels} = I_N) = -\sum_{i=1}^N \sum_{j=1}^N I_N[i, j] \cdot \log \textsc{SoftMax}(F_{\text{next}})[i, j]$, $\textsc{SoftMax}(\cdot)$ denotes row-wise softmax normalization, and I_N is a N dimensional identity matrix. For the second cross entropy term, we first sample a batch of N actions from the target policy at the *next* time step, $a_{t+1}^{(1:N)} \sim \pi(a_{t+1} \mid s_{t+1}, g)$, and then estimate the importance weight matrix $W \in \mathbb{R}^{N \times N}$ that serves as labels as

$$F_w[i, j] = \phi(s_{t+1}^{(i)}, a_{t+1}^{(i)}, g^{(i)})^\top \psi(s_{t+}^{(j)}), W = N \cdot \textsc{SoftMax}(F_w).$$

Thus, the second cross entropy loss takes as inputs the critic F_{future} and the importance weight W:

$$\gamma\mathcal{CE}(\text{logits} = F_{\text{future}}, \text{labels} = W). \tag{13}$$

Regarding the policy objective (Eq. 4), it can also be rewritten as the cross entropy between a critic matrix F_{goal} with $F_{\text{goal}}[i, j] = \phi(s_t^{(i)}, a^{(i)}, g^{(i)})^\top \psi(g^{(j)})$, where $a^{(i)} \sim \pi(a \mid s_t^{(i)}, g^{(i)})$, and the identity matrix I_N:

$$\mathcal{CE}(\text{logits} = F_{\text{goal}}, \text{labels} = I_N)$$

In practice, we use neural networks with parameters $\theta = \{\theta_\phi, \theta_\psi\}$ to parameterize (normalized) contrastive representations ϕ and ψ and use a neural network with parameters ω to parameterize the goal-conditioned policy π and optimize them using gradient descent.

C Theoretical Analysis

Our convergence proof will focus on the tabular setting with known $p(s' \mid s, a)$ and $p(s_{t+})$, and follows the fitted Q-iteration strategy [28, 20, 6]: at each iteration, an optimization problem will be solved exactly to yield the next estimate of the discounted state occupancy measure. One key step in the proof is to employ a preserved invariant; we will define the classifier derived from the TD InfoNCE objective (Eq. 3) and show that this classifier always represents a valid probability distribution (over future states). We then construct a variant of the TD InfoNCE objective using this classifier and prove that optimizing this objective is exactly equivalent to perform policy evaluation, resulting in the convergence to the discounted state occupancy measure.

Definition of the classifier. We start by defining the classifier derived from the TD InfoNCE as

$$C(s, a, s_{t+}) \triangleq \frac{p(s_{t+})e^{f(s,a,s_{t+})}}{\mathbb{E}_{p(s'_{t+})}\left[e^{f(s,a,s'_{t+})}\right]} = \frac{p(s_{t+})e^{f(s,a,s_{t+})}}{\sum_{s'_{t+} \in \mathcal{S}} p(s_{t+})e^{f(s,a,s'_{t+})}}, \tag{14}$$

suggesting that $C(s, a, \cdot)$ is a valid distribution over future states: $C(s, a, \cdot) \in \Delta(\mathcal{S})$.

A variant of TD InfoNCE. Our definition of the classifier (Eq. 14) allows us to rewrite the importance weight $w(s, a, s_{t+})$ and softmax functions in $\mathcal{L}_{\text{TD InfoNCE}}$ (Eq. 3) as Monte Carlo estimates of the classifier using samples of $s_{t+}^{(1:N)}$,

$$w(s, a, s_{t+}^{(1:N)}) = \frac{e^{f(s,a,s_{t+}^{(1)})}}{\frac{1}{N}\sum_{i=1}^N e^{f(s,a,s_{t+}^{(i)})}} \approx \frac{C(s, a, s_{t+})}{p(s_{t+})}.$$

Thus, we construct a variant of the TD InfoNCE objective using C:

$$\bar{\mathcal{L}}_{\text{TD InfoNCE}}(C) \triangleq \mathbb{E}_{p(s,a)}\left[(1-\gamma)\mathbb{E}_{p(s'=s_{t+}|s,a)}\left[\log C(s,a,s_{t+})\right] + \gamma\mathbb{E}_{\substack{p(s'|s,a),\pi(a'|s')\\ p(s_{t+})}}\left[\frac{\lfloor C(s',a',s_{t+})\rfloor_{\text{sg}}}{p(s_{t+})}\log C(s,a,s_{t+})\right]\right].$$

This objective is similar to $\mathcal{L}_{\text{TD InfoNCE}}$, but differs in that *(a)* softmax functions are replaced by $C(s,a,s_{t+})$ up to constant $\frac{1}{N\cdot p(s_{t+})}$ and *(b)* $w(s',a',s_{t+}^{(1:N)})$ is replaced by $\frac{C(s',a',s_{t+})}{p(s_{t+})}$. Formally, $\mathcal{L}_{\text{TD InfoNCE}}(C)$ is a nested Monte Carlo estimator of $\bar{\mathcal{L}}_{\text{TD InfoNCE}}$ [72, 33] and we leave the analysis of the gap between them as future works. We now find the solution of $\bar{\mathcal{L}}_{\text{TD InfoNCE}}(C)$ analytically by rewriting it using the cross entropy and ignore the stop gradient operator to reduce clutter: $\bar{\mathcal{L}}_{\text{TD InfoNCE}}(C) =$

$$\begin{aligned}
&\mathbb{E}_{p(s,a)}\left[(1-\gamma)\mathbb{E}_{p(s'=s_{t+}|s,a)}\left[\log C(s,a,s_{t+})\right] + \gamma\mathbb{E}_{\substack{p(s'|s,a),\pi(a'|s',g)\\ C(s',a',s_{t+})}}\left[\log C(s,a,s_{t+})\right]\right]\\
&= -\mathbb{E}_{p(s,a)}\left[(1-\gamma)\mathcal{CE}\left(p(s'=\cdot\mid s,a), C(s,a,\cdot)\right)\right.\\
&\qquad\left. +\gamma\mathcal{CE}\left(\mathbb{E}_{p(s'|s,a),\pi(a'|s')}\left[C(s',a',\cdot)\right], C(s,a,\cdot)\right)\right]\\
&= -\mathbb{E}_{p(s,a)}\left[\mathcal{CE}\left(C(s,a,\cdot), (1-\gamma)p(s'=\cdot\mid s,a) + \gamma\mathbb{E}_{p(s'|s,a),\pi(a'|s')}\left[C(s',a',\cdot)\right]\right)\right], \qquad (15)
\end{aligned}$$

where the cross entropy for $p, q \in \Delta(\mathcal{X})$ is defined as

$$\mathcal{CE}(p(\cdot), q(\cdot)) = -\mathbb{E}_{p(x)}[\log q(x)] = -\sum_{x\in\mathcal{X}} p(x)\log q(x),$$

with the minimizer $q^\star = \arg\min_{q\in\Delta(\mathcal{X})}\mathcal{CE}(p(\cdot),q(\cdot)) = p$. Note that $p(s'=\cdot\mid s,a) \in \Delta(\mathcal{S})$ and $\mathbb{E}_{p(s'|s,a)\pi(a'|s')}[C(s',a',\cdot)] \in \Delta(\mathcal{S})$ in Eq. 15 indicate that their convex combination is also a distribution in $\Delta(\mathcal{S})$. This objective suggests a update for the classifier given any (s,a,s_{t+}):

$$C(s,a,s_{t+}) \leftarrow (1-\gamma)p(s'=s_{t+}\mid s,a) + \gamma\mathbb{E}_{p(s'|s,a)\pi(a'|s')}[C(s',a',s_{t+})], \qquad (16)$$

which bears a resemblance to the standard Bellman equation.

InfoNCE Bellman operator. We define the InfoNCE Bellman operator for any function $Q(s,a,s_{t+}) : \mathcal{S}\times\mathcal{A}\times\mathcal{S}\mapsto\mathbb{R}$ with policy $\pi(a\mid s)$ as

$$\mathcal{T}_{\text{InfoNCE}}Q(s,a,s_{t+}) \triangleq (1-\gamma)p(s'=s_{t+}\mid s,a) + \gamma\mathbb{E}_{p(s'|s,a)\pi(a'|s',g)}[Q(s',a',s_{t+})], \qquad (17)$$

and write the update of the classifier as $C(s,a,s_{t+}) \leftarrow \mathcal{T}_{\text{InfoNCE}}C(s,a,s_{t+})$. Like the standard Bellman operator, this InfoNCE Bellman operator is a γ-contraction. Unlike the standard Bellman operator, $\mathcal{T}_{\text{InfoNCE}}$ replaces the reward function with the discounted probability of the future state being the next state $(1-\gamma)p(s'=s_{t+}\mid s,a)$ and applies to a function depending on a state-action pair and a future state (s,a,s_{t+}).

Proof of convergence. Using the same proof of convergence for policy evaluation with the standard Bellman equation [82, 1], we conclude that repeatedly applying $\mathcal{T}_{\text{GC InfoNCE}}$ to C results in convergence to a unique $C^\star$ regardless of initialization,

$$C^\star(s,a,s_{t+}) = (1-\gamma)p(s'=s_{t+}\mid s,a) + \gamma\mathbb{E}_{p(s'|s,a)\pi(a'|s')}[C^\star(s',a',s_{t+})].$$

Since $C^\star(s,a,s_{t+})$ and $p^\pi(s_{t+}\mid s,a)$ satisfy the same identity (Eq. 8), we have $C^\star(s,a,s_{t+}) = p^\pi(s_{t+}\mid s,a)$, i.e., the classifier of the TD InfoNCE estimator converges to the discounted state occupancy measure. To recover $f^\star$ from $C^\star$, we note that $f^\star$ satisfies

$$\begin{aligned}
f^\star(s,a,s_{t+}) &= \log C^\star(s,a,s_{t+}) - \log p(s_{t+}) + \log\mathbb{E}_{p(s'_{t+})}[\exp(f^\star(s,a,s'_{t+}))]\\
&= \log p^\pi(s_{t+}\mid s,a) - \log p(s_{t+}) + \log\mathbb{E}_{p(s'_{t+})}[\exp(f^\star(s,a,s'_{t+}))]
\end{aligned}$$

by definition. Since the (expected) softmax function is invariant to translation, we can write $f^\star(s,a,s_{t+}) = \log p^\pi(s_{t+}\mid s,a) - \log p(s_{t+}) - \log c(s,a)$, where $c(s,a)$ is an arbitrary function that does not depend on s_{t+} [2]. Thus, we conclude that TD InfoNCE objective converges to the same solution as that of MC InfoNCE (Eq. 11), i.e. $\bar{\mathcal{L}}_{\text{TD InfoNCE}}(f^\star) = \mathcal{L}_{\text{MC InfoNCE}}(f^\star)$.

[2] Technically, $f^\star$ should be a set of functions satisfying $\left\{f : \frac{e^{f(s,a,s_{t+})}}{\mathbb{E}_{p(s'_{t+})}\left[e^{f(s,a,s'_{t+})}\right]} = \frac{C^\star(s,a,s_{t+})}{p(s_{t+})}\right\}$.

It is worth noting that the same proof applies to the goal-conditioned TD InfoNCE objective. After finding an exact estimate of the discounted state occupancy measure of a goal-conditioned policy $\pi(a \mid s, g)$, maximizing the policy objective (Eq. 4) is equivalent to doing policy improvement. We can apply the same proof as in the Lemma 5 of [23] to conclude that $\pi(a \mid s, g)$ converges to the optimal goal-conditioned policy $\pi^\star(a \mid s, g)$.

D Connection with mutual information and skill learning.

The theoretical analysis in Appendix C has shown that the TD InfoNCE estimator has the same effect as the MC InfoNCE estimator. As the (MC) InfoNCE objective corresponds to a lower bound on mutual information [70], we can interpret our goal-conditioned RL method as having both the actor and the critic jointly optimize a lower bound on mutual information. This perspective highlights the close connection between unsupervised skill learning algorithms [21, 9, 90, 34], and goal-conditioned RL, a connection previously noted in Choi et al. [13]. Seen as an unsupervised skill learning algorithm, goal-conditioned RL lifts one of the primary limitations of prior methods: it can be unclear which skill will produce which behavior. In contrast, goal-conditioned RL methods learn skills that are defined as optimizing the likelihood of reaching particular goal states.

E Connection with Successor Representations

In settings with tabular states, the successor representation [15] is a canonical method for estimating the discounted state occupancy measure (Eq. 7). The successor representation has strong ties to cognitive science [31] and has been used to accelerate modern RL methods [5, 85].

Successor representation $M^\pi : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ is a long-horizon, policy dependent model that estimates the discounted state occupancy measure for every $s \in \mathcal{S}$ via the recursive relationship (Eq. 8). Given a policy $\pi(a \mid s)$, the successor representation satisfies

$$M^\pi(s, a) \leftarrow (1-\gamma)\textsc{OneHot}_{|\mathcal{S}|}(s') + \gamma M^\pi(s', a'), \tag{18}$$

where $s' \sim p(s' \mid s, a)$ and $a' \sim \pi(a' \mid s')$. Comparing this update to the TD InfoNCE update shown in Fig. 1 and Eq. 16, we see that this successor representation update is a special case of TD InfoNCE where *(a)* every state is used instead of randomly-sampling the states, and *(b)* the probabilities are encoded directed in a matrix M, rather than encoding the probabilities as the inner product between two learned vectors.

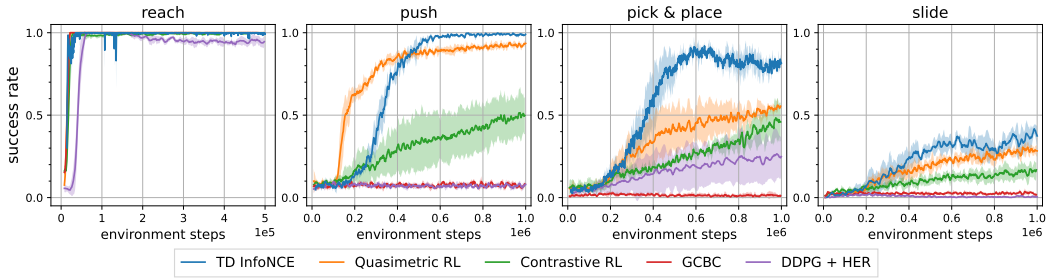
This connection is useful because it highlights how and why the learned representations can be used to solve fully-general reinforcement learning tasks. In the same way that the successor representation can be used to express the value function of a reward ($M^\pi(s, a)^\top r(\cdot)$), the representations learned by TD InfoNCE can be used to recover value functions:

$$\begin{aligned}
Q^\pi(s, a) &= \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \\
&= r(s, a) + \frac{\gamma}{1-\gamma}\mathbb{E}_{p^\pi(s_{t+}|s,a),\pi(a_{t+}|s_{t+})}[r(s_{t+}, a_{t+})] \\
&= r(s, a) + \frac{\gamma}{1-\gamma}\mathbb{E}_{p(s_{t+}),\pi(a_{t+}|s_{t+})}\left[\frac{e^{f^\star(s,a,s_{t+})}}{\frac{1}{N}\sum_{i=1}^N e^{f^\star(s,a,s_{t+}^{(i)})}} r(s_{t+}, a_{t+})\right].
\end{aligned}$$

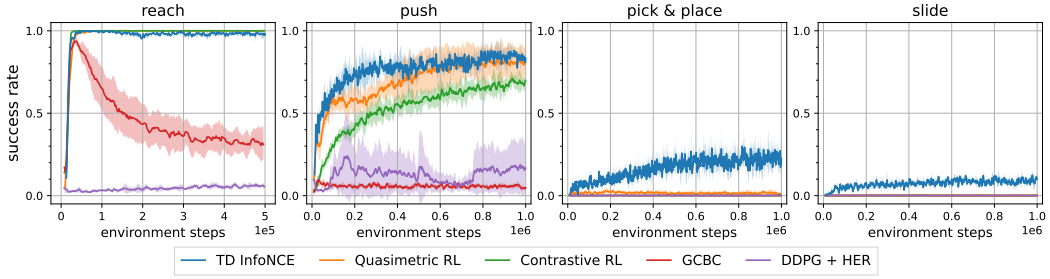
F Experimental Details

F.1 Online Goal-conditioned RL Experiments

We compare TD InfoNCE to four baselines on an online GCRL benchmark [69] containing four manipulation tasks for the Fetch robot. The observations and goals of those tasks can be either a state of the robot and objects or a 64×64 RGB image. We will evaluate using both versions. The first baseline, Quasimetric Reinforcement Learning (QRL) [89], is a state-of-the-art approach that uses quasimetric models to learn the optimal goal-conditioned value functions and the corresponding



(a) State-based tasks



(b) Image-based tasks

Figure 5: **Evaluation on online GCRL benchmarks.** TD InfoNCE matches or outperforms all baselines on both state-based and image-based tasks.

policies. The second baseline is contrastive RL [24], which estimates the discounted state occupancy measure using $\mathcal{L}_{\text{MC InfoNCE}}$ (Eq. 10). Our third baseline is the goal-conditioned behavioral cloning (GCBC) [16, 19, 32, 54, 81, 80]. We also include a comparison with an off-the-shelf actor-critic algorithm augmented with hindsight relabeling [2, 51, 73, 75] to learn a goal-conditioned policy (DDPG + HER).

Results in Fig. 5 show that TD InfoNCE matches or outperforms other baselines on all tasks, both for state and image observations. On those more challenging tasks (`pick & place (state / image)` and `slide (state / image)`), TD InfoNCE achieves a $2\times$ median improvement relative to the strongest baseline. On the most challenging tasks, image-based `pick & place` and `slide`, TD InfoNCE is the only method achieving non-negligible success rates. We speculate this observation is because TD InfoNCE estimates the discounted state occupancy measure more accurately, a hypothesis we will investigate in Appendix. F.3.

Among those baselines, QRL is the strongest one. Unlike TD InfoNCE, the derivation of QRL assumes the dynamics are deterministic. This difference motivates us to study whether TD InfoNCE continues achieving high success rates in environments with stochastic noise. To study this, we compare TD InfoNCE to QRL on a variant of the Fetch benchmark where observations are corrupted with probability 0.1. As shown in Fig. 2b, TD InfoNCE maintains high success rates while the performance of QRL decreases significantly, suggesting that TD InfoNCE can better cope with stochasticity in the environment.

F.2 Offline Goal-conditioned RL Experiments

Similar to prior works [24, 89], we adopt an additional goal-conditioned behavioral cloning regularization to prevent the policy from sampling out-of-distribution actions [29, 48, 47] during policy optimization (Eq.9):

$$\arg\max_{\pi(\cdot|\cdot,\cdot)} \mathbb{E}_{\substack{(s,a_{\text{orig}},g)\sim p(s,a_{\text{orig}},g) \\ a\sim\pi(a|s,g), s_{t+}^{(1:N)}\sim p(s_{t+})}} \left[(1-\lambda)\cdot \log \frac{e^{f(s,a,g,s_{t+}=g)}}{\sum_{i=1}^{N} e^{f(s,a,g,s_{t+}^{(i)})}} + \lambda \cdot \|a - a_{\text{orig}}\|_2^2 \right],$$

where λ is the coefficient for regularization. Note that we use a supervised loss based on the mean squared error instead of the maximum likelihood estimate of a_{orig} under policy π used in prior works.

Table 1: Evaluation on offline D4RL AntMaze benchmarks.

	TD InfoNCE	QRL	Contrastive RL	GCBC	DT	IQL	TD3 + BC
umaze-v2	**85.8 ± 0.9**	77.2 ± 2.3	79.8 ± 1.4	65.4	65.6	**87.5**	78.6
umaze-diverse-v2	**92.1 ± 1.1**	79.4 ± 1.5	77.6 ± 2.8	60.9	51.2	62.2	71.4
medium-play-v2	**87.5 ± 1.2**	74.9 ± 1.9	72.6 ± 2.9	58.1	1.0	71.2	10.6
medium-diverse-v2	**82.3 ± 2.8**	73.1 ± 1.1	71.5 ± 1.3	67.3	0.6	70.0	3.0
large-play-v2	47.3 ± 2.9	**52.3 ± 3.2**	48.6 ± 4.4	32.4	0.0	39.6	0.2
large-diverse-v2	**56.2 ± 3.8**	50.9 ± 4.6	**54.1 ± 5.5**	36.9	0.2	47.5	0.0

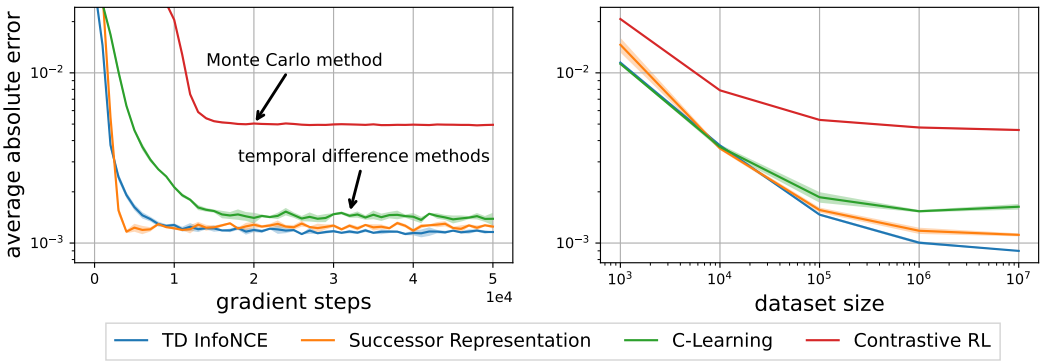


Figure 6: **Estimating the discounted state occupancy measure in a tabular setting.** *(Left)* Temporal difference methods have lower errors than the Monte Carlo method. Also note that our TD InfoNCE converges as fast as the best baseline (successor representation). *(Right)* TD InfoNCE is more data efficient than other methods. Using a dataset of size 10M, TD InfoNCE achieves an error rate 25% lower than the best baseline; TD InfoNCE also matches the performance of C-learning with 130× less data.

We compare TD InfoNCE to the state-of-the-art QRL [89] and its Monte Carlo counterpart (contrastive RL [24]). We also compare to the pure goal-conditioned behavioral cloning implemented in [19] as well as a recent baseline that predicts optimal actions via sequence modeling using a transformer (DT [11]). Our last two baselines are offline actor-critic methods trained via TD learning: TD3 + BC [29] and IQL [46], not involving goal-conditioned relabeling. We use the result for baselines except QRL from [24].

As shown in Table 1, TD InfoNCE matches or outperforms all baselines on 5 / 6 tasks. On tasks (`medium-play-v2` and `medium-diverse-v2`), TD InfoNCE achieves a $+13\%$ improvement over contrastive RL, showing the advantage of temporal difference learning over the Monte Carlo approach with a fixed dataset. We conjecture that this benefit comes from the dynamic programming property of the TD method and will investigate this property further in later experiments (Sec. F.4). Additionally, TD InfoNCE performs $1.4\times$ better than GCBC and retains a $3.8\times$ higher scores than DT on average, where these baselines use (autoregressive) supervised losses instead of TD learning. These results suggest that TD InfoNCE is also a competitive goal-conditioned RL algorithm in the offline setting.

F.3 Accuracy of the estimated discounted state occupancy measure

This section tests the hypothesis that our TD InfoNCE loss will be more accurate and sample efficient than alternative Monte Carlo methods (namely, contrastive RL [24]) in predicting the discounted state occupancy measure. We will use the tabular setting so that we can get a ground truth estimate. We compare TD InfoNCE to three baselines. Successor representations [15] can also be learned in a TD manner, though can be challenging to apply beyond tabular settings. C-learning is similar to TD InfoNCE in that it uses a temporal difference method to optimize a contrastive loss, but differs in using a binary cross entropy loss instead of a softmax cross entropy loss. Contrastive RL is the MC counterpart of TD InfoNCE. We design a 5×5 gridworld with 125 states and 5 actions (up, down, left, right, and no-op) and collect 100K transitions using a uniform random policy, $\mu(a \mid s) = \textsc{Unif}(\mathcal{A})$. We evaluate each method by measuring the absolute error between the

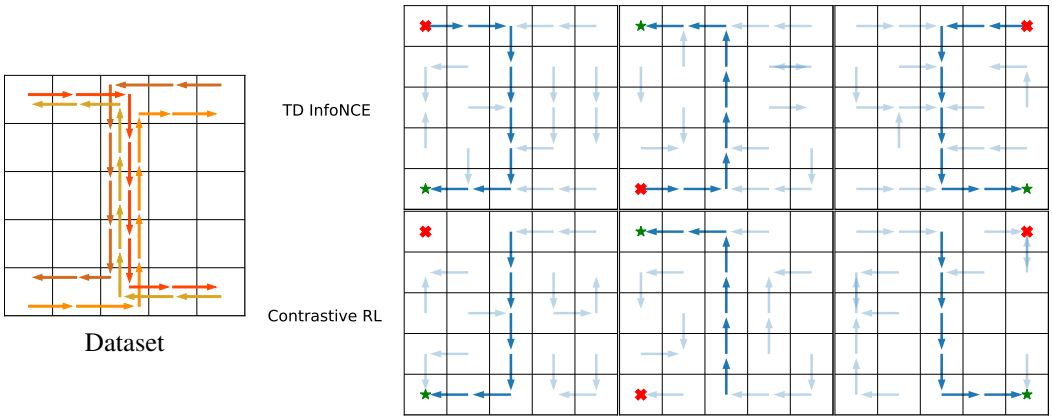


Figure 7: **Stitching trajectories in a dataset.** The behavioral policy collects "Z" style trajectories. Unlike the Monte Carlo method (contrastive RL) , our TD InfoNCE successfully "stitches" these trajectories together, navigating between pairs of (start ✖, goal ★) states unseen in the training trajectories.

predicted probability $\hat{p}$ and the ground truth probability p^μ, averaging over all pairs of (s, a, s_{t+}):

$$\frac{1}{|\mathcal{S}||\mathcal{A}||\mathcal{S}|} \sum_{s,a,s_{t+}} |\hat{p}(s_{t+} \mid s, a) - p^\mu(s_{t+} \mid s, a)|.$$

For the three TD methods, we compute the TD target in a SARSA manner [82]. For those methods estimating a probability ratio, we convert the prediction to a probability by multiplying by the empirical state marginal. Results in Fig. 6 show that TD methods achieve lower errors than the Monte Carlo method, while TD InfoNCE converges faster than C-Learning. Appendix G.1 discusses why all methods plateau above zero.

Our next experiments studies sample efficiency. We hypothesize that the softmax in the TD InfoNCE loss may provide more learning signal than alternative methods, allowing it to achieve lower error on a fixed budget of data. To test this hypothesis, we run experiments with dataset sizes from 1K to 10M on the same gridworld, comparing TD InfoNCE to the same set of baselines. We report results in Fig. 6 with errors showing one standard deviation after training for 50K gradient steps for each approach. These results suggest that methods based on temporal difference learning predict more accurately than Monte Carlo method when provided with the same amount of data. Compared with its Monte Carlo counterpart, TD InfoNCE is $1500\times$ more sample efficient (6.5×10^3 vs 10^7 transitions). Compared with the only other TD method applicable in continuous settings (C-learning), TD InfoNCE can achieve a comparable loss with $130\times$ less data (7.7×10^4 vs 10^7 transitions). Even compared with the strongest baseline (successor representations), which makes assumptions (tabular MDPs) that our method avoids, TD InfoNCE can achieve a comparable error rate with almost $20\times$ fewer samples (5.2×10^5 vs 10^7 transitions).

F.4 Off-Policy Reasoning Experiments

The explicit temporal difference update (Eq. 3) in TD InfoNCE is similar to the standard Bellman backup, motivating us to study whether the resulting goal-conditioned policy is capable of performing dynamic programming with offline data. To answer these questions, we conduct two experiments on the same gridworld environment as in Sec. F.3, comparing TD InfoNCE to contrastive RL (i.e., Monte Carlo InfoNCE).

Stitching trajectories. The first set of experiments investigate whether TD InfoNCE successfully stitches pieces of trajectories in a dataset to find complete paths between (start, goal) pairs unseen together in the dataset. We collect a dataset with size 20K consisting of "Z" style trajectories moving in diagonal and off-diagonal directions (Fig. 7), while evaluating the learned policy on reaching goals on the same edge as starting states after training both methods for 50K gradient steps. Figure 7 shows that TD InfoNCE succeeds in stitching parts of trajectory in the dataset, moving along "C"

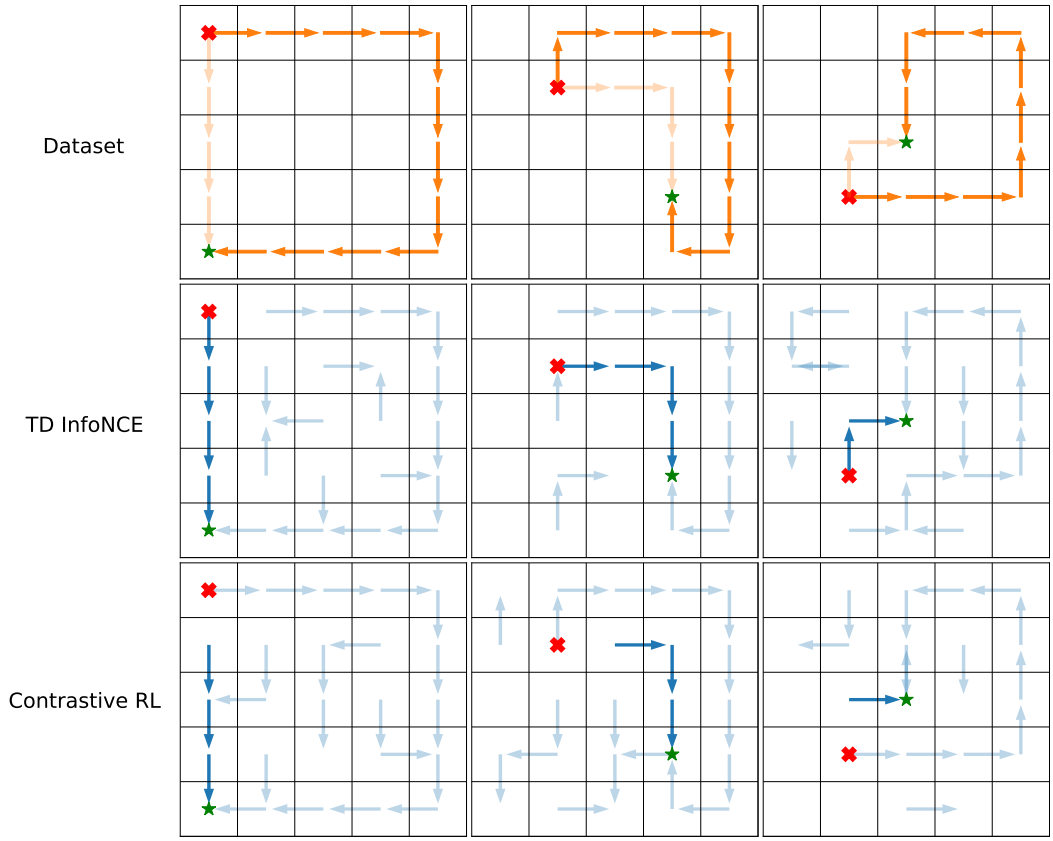


Figure 8: **Searching for shortcuts in skewed datasets.** *(Left)* Conditioned on different initial states ✖ and goals ★, we collect datasets with 95% long paths (dark) and 5% short paths (light). *(Center)* TD InfoNCE infers the shortest path, *(Right)* while contrastive RL fails to find this path.

style paths towards goals, while contrastive RL fails to do so. These results justify our hypothesis that TD InfoNCE performs dynamic programming and contrastive RL instead naively follows the behavior defined by the data.

Searching for shortcuts. Our second set of experiments aim to compare the performance of TD InfoNCE against contrastive RL on searching shortcuts in skewed datasets. To study this, we collect different datasets of size 20K with trajectories conditioned on the same pair of initial state and goal, with 95% of the time along a long path and 5% of the time along a short path. Using these skewed datasets, we again train both methods for 50K gradient steps and then evaluate the policy performance on reaching the same goal starting from the same state. We show the goal-conditioned policies learned by the two approaches in Fig. 8. The observation that TD InfoNCE learns to take shortcuts even though those data are rarely seen, while contrastive RL follows the long paths dominating the entire dataset, demonstrates the advantage of temporal difference learning over its Monte Carlo counterpart in improving data efficiency.

F.5 Implementations and Hyperparameters

We implement TD InfoNCE, contrastive RL, and C-Learning using JAX [8] building upon the official codebase of contrastive RL[3]. For baselines QRL, GCBC, and DDPG + HER, we use implementation provided by the author of QRL[4]. We summarize hyperparameters for TD InfoNCE in Table 2. Whenever possible, we used the same hyperparameters as contrastive RL [24]. Since TD InfoNCE computes the loss with N^2 negative examples, we increase the capacity of the goal-conditioned

[3]https://github.com/google-research/google-research/tree/master/contrastive_rl
[4]https://github.com/quasimetric-learning/quasimetric-rl

Table 2: Hyperparameters for TD InfoNCE.

Hyperparameters	Values
actor learning rate	5×10^{-5}
critic learning rate	3×10^{-4}
using ℓ_2 normalized representations	yes
hidden layers sizes (for both actor and representations)	$(512, 512, 512, 512)$
contrastive representation dimensions	16

Table 3: Changes to hyperparameters for offline RL experiments. (Table 1)

Hyperparameters	Values
batch size (on `large-` tasks)	256 $\rightarrow$ 1024
hidden layers sizes (for both actor and representations on `large-` tasks)	(512, 512, 512, 512) $\rightarrow$ (1024, 1024, 1024, 1024)
behavioral cloning regularizer coefficient λ	0.1
goals for actor loss	random states $\rightarrow$ future states

state-action encoder and the future state encoder to 4 layers MLP with 512 units in each layer applying ReLU activations. We find that a ℓ_2 normalized representation space is important for TD InfoNCE. For offline RL experiments, we make some changes to hyperparameters (Table 3).

G Additional Experiments

G.1 Predicting the discounted state occupancy measure

Our experiments estimating the discounted state occupancy measure in the tabular setting (Sec. F.3) observed a small "irreducible" error. To test the correctness of our implementation, we applied the successor representation with a known model (Fig. 9), finding that its error does go to zero. This gives us confidence that our implementation of the successor representation baseline is correct, and suggests that the error observed in Fig. 6 arises from sampling the transitions (rather than having a known model).

G.2 Understanding the Differences between TD InfoNCE and C-Learning

While conceptually similar, our method is a temporal difference estimator building upon InfoNCE whereas C-learning instead bases on the NCE objective [37]. There are mainly three distinctions between TD InfoNCE and C-Learning: *(a)* C-Learning uses a binary cross entropy loss, while TD

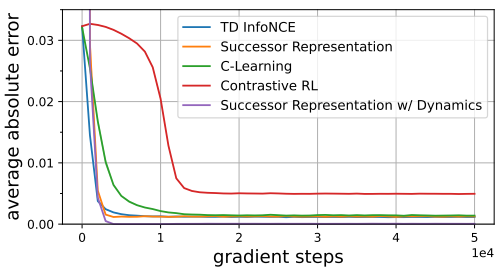


Figure 9: Errors of discounted state occupancy measure estimation in a tabular setting.

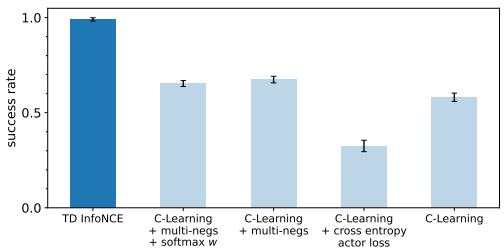
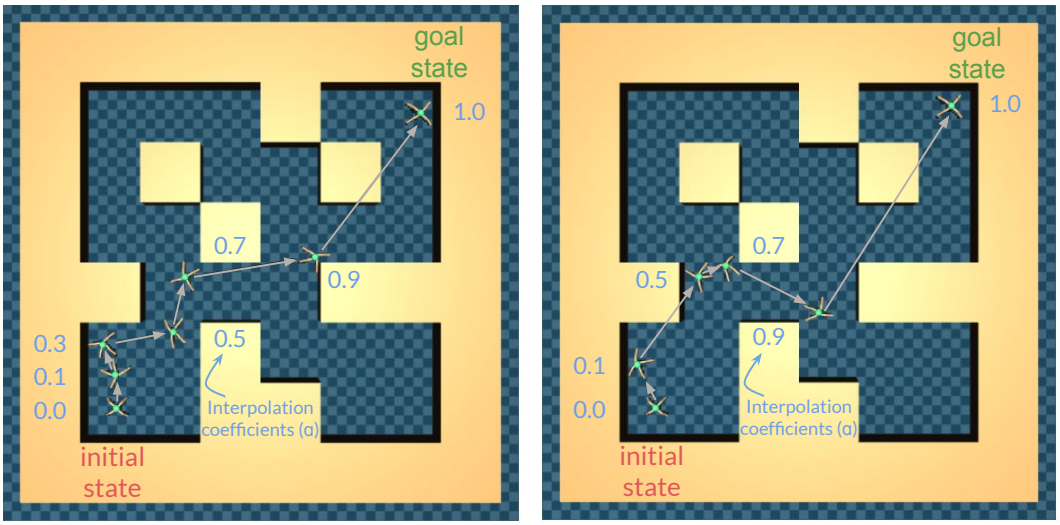


Figure 10: Differences between TD InfoNCE and C-Learning.



(a) Parametric interpolation

(b) Non-parametric interpolation

Figure 11: **Visualizing representation interpolation.** Using spherical interpolation of representations *(Left)* or linear interpolation of softmax features *(Right)*, TD InfoNCE learns representations that capture not only the content of states, but also the causal relationships.

InfoNCE uses a categorical cross entropy loss. *(b)* C-Learning uses importance weights of the form $\exp(f(s, a, g))$; if these weights are self-normalized [17, 38], they corresponds to the softmax importance weights in our objectives (Eq. 2). *(c)* For the same batch of N transitions, TD InfoNCE updates representations of N^2 negative examples (Eq. 13), while C-Learning only involves N negative examples. We ablate these decisions in Fig. 10, finding that differences (b) and (c) have little effect. Thus, we attribute the better performance of TD InfoNCE to its use of the categorical cross entropy loss.

G.3 Representation Interpolation

Prior works have shown that representations learned by self-supervised learning incorporate structure of the data [88, 3], motivating us to study whether the representations acquired by TD InfoNCE contain task-specific information. To answer this question, we visualize representations learned by TD InfoNCE via interpolating in the latent space following prior work [95]. We choose to interpolate representations learned on the offline AntMaze `medium-play-v2` task and compare a parametric interpolation method against a non-parametric variant. Importantly, the states and goals of this task are 29 dimensions and we visualize them in 2D from a top-down view.

Parametric interpolation. Given a pair of start state and goal (s_0, g), we compute the normalized representations $\phi(s_0, a_{\text{no-op}}, g)$ and $\phi(g, a_{\text{no-op}}, g)$, where $a_{\text{no-op}}$ is an action taking no operation. Applying spherical linear interpolation to both of them results in blended representations,

$$\frac{\sin(1-\alpha)\eta}{\sin\eta}\phi(s_0, a_{\text{no-op}}, g) + \frac{\sin\alpha\eta}{\sin\eta}\phi(g, a_{\text{no-op}}, a),$$

where $\alpha \in [0, 1]$ is the interpolation coefficient and η is the angle subtended by the arc between $\phi(s_0, a_{\text{no-op}}, g)$ and $\phi(g, a_{\text{no-op}}, g)$. These interpolated representations can be used to find the spherical nearest neighbors in a set of representations of validation states $\{\phi(s_{\text{val}}, a_{\text{no-op}}, g)\}$ and we call this method parametric interpolation.

Non-parametric interpolation. We can also sample another set of random states and using their representations $\{\phi(s^{(i)}_{\text{rand}}, a_{\text{no-op}}, g)\}_{i=1}^{S}$ as anchors to construct a softmax feature for a state s, $\text{feat}(s; g, \{s_{\text{rand}}\}) =$

$$\textsc{SoftMax}\left(\left[\phi(s, a_{\text{no-op}}, g)^\top \phi(s^{(1)}_{\text{rand}}, a_{\text{no-op}}, g), \cdots, \phi(s, a_{\text{no-op}}, g)^\top \phi(s^{(S)}_{\text{rand}}, a_{\text{no-op}}, g)\right]\right).$$

We compute the softmax features for representations of start and goal states and then construct the linear interpolated features,

$$\alpha \text{feat}(s_0; g, \{s_{\text{rand}}\}) + (1 - \alpha)\text{feat}(g; g, \{s_{\text{rand}}\}).$$

Those softmax features of interpolated representations are used to find the ℓ_2 nearest neighbors in a softmax feature validation set. We call this method non-parametric interpolation.

Results in Fig. 11 suggest that when interpolating the representations using both methods, the intermediate representations correspond to sequences of states that the optimal policy should visit when reaching desired goals. Therefore, we conjecture that TD InfoNCE encodes causality in its representations while the policy learns to arrange them in a temporally correct order.