Self-Predictive Representations for Combinatorial Generalization in Behavioral Cloning

Anonymous authors
Paper under double-blind review

Abstract

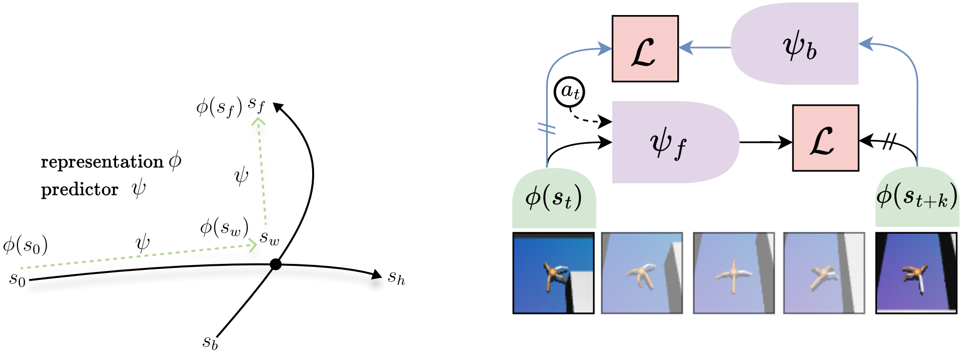
While goal-conditioned behavior cloning (GCBC) methods can perform well on in-distribution training tasks, they do not necessarily generalize zero-shot to tasks that require conditioning on novel state-goal pairs, i.e. *combinatorial generalization*. In part, this limitation can be attributed to a lack of temporal consistency in the state representation learned by BC; if temporally correlated states are properly encoded to similar latent representations, then the out-of-distribution gap for novel state-goal pairs would be reduced. We formalize this notion by demonstrating how encouraging long-range temporal consistency via successor representations (SR) can facilitate generalization. We then propose a simple yet effective representation learning objective, BYOL-γ for GCBC, which theoretically approximates the successor representation in the finite MDP case through self-predictive representations, and achieves competitive empirical performance across a suite of challenging tasks requiring combinatorial generalization.

1 Introduction

Generalization has been a long-standing goal in machine learning and robotics. Recently, large-scale supervised models for language and vision have demonstrated impressive generalization when trained over vast amounts of data. In robotics, this has motivated large-scale behavior cloning (BC) models trained on offline datasets of diverse demonstrations (Ghosh et al., 2024; Kim et al., 2024). However, these models still suffer from a lack of generalization. In particular, while BC methods can perform well on tasks directly observed in the dataset, they often fail to perform zero-shot transfer to tasks requiring novel combinations of in-distribution behavior, known as *combinatorial generalization*. In the robotics domain, where demonstration data is time-intensive and costly to produce, simply scaling the dataset is often not possible. Hence, achieving this type of generalization algorithmically will be critical to unlocking the potential for large-scale supervised policy training.

The property of combinatorial generalization has been previously formalized as the ability to “stitch” (Ghugare et al., 2024). Here, stitching refers to the ability of a policy to reach a goal state from a start state when trained on a dataset, which provides sufficient coverage of the path to the goal, but which does not contain a single complete trajectory of the path. The lack of stitching observed in goal-conditioned behavioral cloning (GCBC) and, more generally, supervised learning, can be understood through the inductive biases of the model. By construction, BC methods do not encode the inductive bias that the observed data are generated from a Markov decision process (MDP). In contrast, reinforcement learning (RL) policies that are trained via temporal difference (TD) learning directly utilize the structure of the MDP to pass information through time using dynamic programming. Offline RL (Levine et al., 2020) has been proposed as a method for achieving stitching in policies trained on offline datasets. However, these methods are challenging to scale due to the instability of bootstrapping in TD learning when combined with fully offline training. Scaling has been more successful with supervised methods, such as in robotics, where training robot foundation models with BC (Ghosh et al., 2024; Kim et al., 2024) on large-scale datasets (O’Neill et al., 2024; Khazatsky et al., 2024) can lead to more general-purpose policies.

A goal-conditioned policy being general-purpose implies it has learned an implicit world model of the environment (Richens et al., 2025). From this intuition, a key desiderata is to make a policy’s representations align with the (latent) dynamics of the underlying environment, in order to obtain a more robust goal-conditioned policy. However, an open question here is which representation



(a)

(b)

Figure 1: **(a) Self-predictive Representations.** We consider training on trajectories like, $s_0 \rightarrow s_h$ and $s_b \rightarrow s_f$, which intersect at w, and then evaluate evaluate on a task like $s_0 \rightarrow s_f$, requiring combinatorial generalization. **(b) Representation learning with BYOL-γ.** We predict future state representations $\phi(s_{t+k})$ via $\psi_f(\phi(s_t), a)$, and also predict backwards with $\psi_b(\phi(s_{t+k}))$. The target offset is sampled geometrically: $k \sim \text{geom}(1-\gamma)$. Stop-gradients are denoted by //. We provide more details on the training procedure $\mathcal{L}$ in Section 4.2.

learning objective best achieves this property. We begin to investigate this question with *Bootstrap Your Own Latent* (BYOL) framework (Grill et al., 2020), which in RL, learns a representation space through predicting future latent states (Schwarzer et al., 2020), without requiring negative samples nor TD-learning. While the standard BYOL objective has been shown to learn representations capturing spectral information about the one-step transition dynamics (Khetarpal et al., 2025), we find that a key property is to capture temporally extended information, leading us to (1) propose a novel objective, **BYOL-γ** which predicts future states geometrically (Figure 1b), and (2) present a unifying framework (Table 1) for understanding objectives related to the successor representation (Blier et al., 2021), including contrastive learning, BYOL, BYOL-γ, and novel applications for TD-based approximation of the SR as an auxiliary loss for BC. Namely, we quantity how these methods uniquely behave when applied to data collected by a mixture of policies which is encountered in practical BC settings.

In the finite, single-policy, MDP case, we show that, in fact, BYOL-γ approximates the successor representation. While in the mixture-policy case, BYOL-γ corresponds to approximating a mixture of SRs, but less with pessimism than existing contrastive objectives. Qualitatively, BYOL-γ objective learns representations that encodes long-range temporal distance between states on mixture datasets more faithfully when compared to TD than contrastive learning (Figure 2). Empirically, we demonstrate on the challenging OGBench suite (Park et al., 2025) that BYOL-γ augmented GCBC outperforms all other methods (Table 2) on average, and is robust to combinatorial generalization with increasing horizons (Figure 3). Our representation can also be extended to hierarchical setups (Appendix C), which leads to further improvements in combinatorial generalization.

2 Related Work

Stitching in Supervised Methods. Outcome (goals or return)-conditioned behavioral cloning (OCBC) methods (Schmidhuber, 2020; Chen et al., 2021; Emmons et al., 2022) provide a simple and scalable alternative to traditional offline RL (Levine et al., 2020) methods. However, these methods do not properly "stitch" and generalize to unseen outcomes (Brandfonbrener et al., 2022; Ghugare et al., 2024). To reduce this problem, various works have proposed augmenting training data used by BC methods. Some work incorporates methodlogy from offline RL to label returns or goals for downstream SL (Char et al., 2022; Yamagata et al., 2023). Other work has considered relabeling goals through clustering states (Ghugare et al., 2024), which relies on a good distance metric, or utilized planing Zhou et al. (2024) for goal relabeling, or generative models to synthesize new trajectories (Lu et al., 2023; Lee et al., 2024). Rather than using models to generate data, combinatorial generalization can be achieved by planning with generative models (Luo et al., 2025). In this work, we neither require explicit Q-learning, generative models, or perform explicit planning.

Representation learning in RL. Our objective is most closely related to approaches using auxiliary **BYOL objectives in online RL** (Gelada et al., 2019; Schwarzer et al., 2020; Ni et al., 2024; Voelcker et al., 2024). These objectives can help with sample-efficiency, such as in challenging, partially observed environments with sparse rewards, or with noisy states. Additionally, self-predictive dynamics models are used in planning and model-based RL (François-Lavet et al., 2019; Ye et al., 2021; Hansen et al., 2022). Various works have also characterized the dynamics of BYOL objectives in the RL setting, showing that BYOL objectives capture spectral information about the policy's transitions (Tang et al., 2023; Khetarpal et al., 2025). In the offline setting, how well Joint Embedding Predictive Architecture (JEPA) world models generalize when used for explicit planning has been studied Sobal et al. (2025), however not for combinatorial generalization. Additionally, certain representation structures for value functions, namely quasimetrics (Liu et al., 2023; Wang et al., 2023; Wang and Isola, 2022; Myers et al., 2024) can also lead to policies that better generalize to longer horizons (Myers et al., 2025a).

Successor Representation (SR) (Dayan, 1993) objectives, such as successor features (SF) (Barreto et al., 2017), and the successor measure (SM) (Blier et al., 2021) have been widely used for generalization and transfer in reinforcement learning (Carvalho et al., 2024). Similarly to BYOL, these objectives have been used for representation learning in RL (Lan et al., 2022; Farebrother et al., 2023). While prior BYOL methods either perform 1-step, or relatively short fixed n-step prediction, neither of these choices directly approximate the successor measure. Our setup is most related to temporal representation alignment (TRA) (Myers et al., 2025b), which recently proposed using contrastive learning as an auxiliary objective for BC to improve combinatorial generalization. In this work, we further build on the relationship between the SM and combinatorial generalization, and propose new objectives which can lead to better performance.

3 Background

Controlled Markov Process. We consider goal-conditioned decision-making, with states $\mathcal{S}$, actions $\mathcal{A}$, goals $g \in S$, initial state distribution $p_0(s)$, dynamics $p(s_{t+1} \,|s_t, a)$, and with policies $\pi(a|s, g)$.

Successor Representation (SR) and Successor Measure (SM). In a finite MDP, the *successor representation* (SR) (Dayan, 1993) of a policy is: $M^\pi(s, s') := \mathbb{E}\left[\sum_{t \geq 0} \gamma^t \mathbb{1}_{(s_{t+1}=s')} \mid s_0 = s, \pi\right]$ We use the convention of counting from s_{t+1}, writing in matrix form $M^\pi = \sum_{t\geq 0} \gamma^t (P^\pi)^{t+1}$. The transition matrix transition for policy π is P^π, with $P^\pi_{i,j} = \sum_a \pi(a|s = i) P_{i,a,j}$, where $P_{i,a,j} = p(s_{t+1} = j \mid s_t = i, a)$. The successor representation also satisfies the bellman equation, $M^\pi = P^\pi + \gamma P^\pi M^\pi = P^\pi (I - \gamma P^\pi)^{-1}$. For a fixed policy, the successor representation describes a type of temporal distance between states. The *successor measure* (SM) (Blier et al., 2021) extends SR to continuous spaces S: $M^\pi(s, X) := \sum_{t\geq 0} \gamma^t P(s_{t+1} \in X \mid s) \, \forall X \subset S$. We also define the *normalized* successor representation, or measure $\tilde{M}^\pi = (1 - \gamma) M^\pi$. In the finite case, the normalized successor representation $\tilde{M}^\pi$ has rows that sum to one like transitions P^π. We also define the state occupancy via $M^\pi(s') = \mathbb{E}_{s\sim p_0(s)} [M^\pi(s_0, s')]$. Another quantity, *successor features* (SF) (Barreto et al., 2017) are the expected discounted sum of future features $\phi(s) \in \mathbb{R}^d$: $\psi^\pi(s) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t \phi(s_{t+1}) \mid s_0 = s, \pi\right]$. We can relate SFs to the SM with $\psi^\pi(s) = \int_{s'} M^\pi(s, s') \phi(s')$. These quantities can also condition an action, e.g. $M^\pi(s, a, s')$.

3.1 Representation Learning

We begin with two representation learning methods that approximate the density of the SM.

Contrastive Learning. Temporal contrastive learning used in MDPs (Eysenbach et al., 2022) is related to a Monte Carlo (MC) approximation of the (discounted) successor measure. This can be implemented with a InfoNCE (van den Oord et al., 2019) loss that maximizes the similarity of a positive pair between a state s_t and a future state from the same trajectory s_+, and minimizing the similarity of s_t and random states s_-:

$$\min_{\phi,\psi} \mathbb{E}_{\substack{s_t \sim p(s) \\ k \sim \text{geom}(1-\gamma) \\ s_+ = s_{t+k}, s_-^{2:N} \sim p(s)}} \left[-\log \frac{e^{f(\psi(s_t), \phi(s_+))}}{\sum_{i=2}^N e^{f(\psi(s_t), \phi(s^i_-))}} \right] \tag{1}$$

A common choice for the energy function f is the inner product $f(\psi(s)\phi(s_+)) = \psi(s)^T\phi(s_+)$. A key aspect to note is that the positive sample s_+ comes from an MC sample from $s_+ \sim M^\pi(s_t, s_+)$. The optimal solution to (1) gives $\tilde{M}^\pi(s, s_+) \approx C\exp(\psi(s_t)^T\phi(s_+)) \cdot p(s_+)$.

Temporal-Difference Approximation of SR (TD-SR) We consider a Forward-Backward (Touati and Ollivier, 2021)-like loss that approximates the successor measure for a fixed policy π using TD learning, discussed by Touati et al. (2023), which we call TD-SR.

$$\min_{\phi,\psi} \mathbb{E}_{\substack{s_t\sim p(s), s'\sim p(s)\\ s_{t+1}\ \sim p^\pi(s_{t+1}|s_t)}} \left[(\psi(s_t)^T\phi(s')\ -\ \gamma\bar{\psi}(s_{t+1})^T\bar{\phi}(s'))^2\right] - 2\mathbb{E}_{\substack{s_t\sim p(s)\\ s_{t+1}\sim p^\pi(s_{t+1}|s_t)}} \left[\psi(s_t)^T\phi(s_{t+1})\right] \tag{2}$$

TD-SR learns an approximation of the successor measure with factorization $M^\pi(s, s_+) \approx \psi(s_t)^T\phi(s_+) \cdot p(s_+)$ using TD learning. Given transitions (s_t, s_{t+1}) sampled by a policy π, the second term relates to fitting $M^\pi(s_t, s_{t+1})$. Given an independently sampled state s', the first term bootstraps an estimate of $M^\pi(s_t, s')$ from $\bar{M}^\pi(s_{t+1}, s')$, where $\bar{\phi}, \bar{\psi}$ denote stop-gradient operations. In Appendix D, we further elaborate on the relationship between the TD-SR loss, and CL. Particularly, in the limit, an n-step version of TD-SR is related to CL.

BYOL. We now look at an objective that captures information about single-step transition instead of the successor measure. In the context of RL, self-predictive models jointly learn a latent space and a dynamics model through predicting future latent representations. Self-predictive models rely on latent bootstrapped targets (BYOL) (Grill et al., 2020), avoiding reconstruction (generative models), or negative samples (contrastive learning). Self-predictive models are also a type of joint-embedding predictive architectures (JEPAs) (LeCun, 2022; Garrido et al., 2024).

Given an encoder which produces a representation $z_t = \phi(s_t)$, and dynamics $\psi(z_{t+1}|z_t)$ for a fixed policy π, we minimize the difference between our prediction and target representation in latent-space:

$$\min_{\phi,\psi} \mathbb{E}_{s_t\sim\ p(s), s_{t+1}\sim p^\pi(s_{t+1}|s_t)}, \left[f(\psi(\phi(s_t)), \bar{\phi}(s_{t+1}))\right] \tag{3}$$

Where f measures the discrepancy between representations, such as the squared l_2 norm, and $\bar{\phi}$ refers to an EMA target, or stop-gradient. Variants of this BYOL objective have been widely used to learn state abstractions, and work as an auxiliary loss to value-function learning (Gelada et al., 2019; Schwarzer et al., 2020; Ni et al., 2024). In the finite MDP, this objective captures spectral information about the policy's transitions P^π (Tang et al., 2023; Khetarpal et al., 2025) , as in Appendix E.1.

3.2 Combinatorial Generalization from Offline Data

We now shift focus on how we can learn policies from offline data using behavioral cloning, and then introduce a combinatorial generalization gap that arises in this setting.

We consider a **dataset** $\mathcal{D} = \{(s_0^i, a_0^i, \cdots, s_T^i, a_T^i)\}_{i=1}^N$, composed of trajectories generated by a set of unknown policies $\{\beta_j(a|s)\}$. **Goal Conditioned Behavioral Cloning (GCBC)** trains a policy π_Θ with maximum likelihood to reproduce the behaviors from the dataset. After sampling a current state, a goal is sampled as a future state from the same trajectory:

$$\max_{\pi_\Theta} \mathcal{L}_{\text{BC}}(\pi_\Theta) = \max_{\pi} \mathbb{E}_{\substack{\beta_j\sim p(\beta_j),\ s\sim M^{\beta_j}(s)\\ a\sim\beta_j(a\ |\ s),\ s_+\sim M^{\beta_j}(s,s_+)}} \left[\log \pi_\Theta(a|s, g = s_+)\right] \tag{4}$$

Generalization gap. While this policy can perform well in-distribution, the behavior cloning policy struggles to generalize to reach goals from states that are not in matching training trajectories. We now review a more formal definition of this type of generalization gap.

We consider Lemma 3.1 from Ghugare et al. (2024), which says there exists a single Markovian policy $\beta(a|s)$ that has the same occupancy as the mixture of j policies: $M^\beta(s) = \mathbb{E}_{p(\beta_j)}\left[M^{\beta_j}(s)\right]$. This policy also has construction: $\beta(a \mid s) := \sum_j \beta_j(a \mid s)p(\beta_j \mid s)$, where $p(\beta_j \mid s)$ is the distribution over policies in s as reflected by the dataset.

Using the successor measure of the individual policies, and the mixture policy, we can quantify a gap between accomplishing out-of-distribution tasks versus in-distribution training tasks (Ghugare et al., 2024):

$$\underbrace{\mathbb{E}_{\substack{s_0\sim M^{\beta}(s_0)\\ s_g\sim M^{\beta}(s_0,s_g)}}\left[u^{\pi_\Theta}(s_0,s_g)\right]}_{\text{tasks requiring combinatorial generalization}} - \underbrace{\mathbb{E}_{\substack{\beta_j\sim p(\beta_j),\, s_0\sim M^{\beta_j}(s_0)\\ s_g\sim M^{\beta_j}(s_0,s_g)}}\left[u^{\pi_\Theta}(s_0,s_g)\right]}_{\text{in-distribution training tasks}} \tag{5}$$

Here, u is a performance metric of the policy π_Θ such as the success rate to reach s_g from s_0. As we perform well on in-distribution tasks due to a correspondence to Equation (4), the BC policy has no guarantees for the first term. This is because after sampling a state, the goal is sampled from the successor measure of the mixture policy.

4 Using Representations for Combinatorial Generalization

In this section, we aim to reduce the aforementioned generalization gap. We consider a policy trained with the BC objective π_Θ to be made more robust to the tasks requiring combinatorial generalization through representation learning. We begin with a setup similar to Equation (5), but with a shared initial state s_0 for both the in-distribution and out-of-distribution task. For the in-distribution task, we sample a goal as before, labeled as s_w. However, for the out-of-distribution task, we sample a goal s_f to be a state that can be reached by the mixture policy β after s_w. (6):

$$\mathbb{E}_{\substack{\beta_j\sim p(\beta_j),s_0\sim M^{\beta_j}(s)\\ s_w\sim M^{\beta_j}(s_0,s_w)}}\left[\underbrace{\mathbb{E}_{s_f\sim M^{\beta}(s_w,s_f)}\left[u^{\pi_\Theta}(s_0,s_f))\right]}_{\text{extended task requiring generalization}} - \underbrace{u^{\pi_\Theta}(s_0,s_w)}_{\text{in-distribution task}}\right] \tag{6}$$

$$= \mathbb{E}_{\substack{\beta_j\sim p(\beta_j),s_0\sim M^{\beta_j}(s)\\ s_w\sim M^{\beta_j}(s_0,s_w)}}\left[\underbrace{\mathbb{E}_{s_f\sim M^{\beta}(s_w,s_f)}\left[u^{\pi_\Theta}(s_0,\phi(s_f))\right]}_{\text{want invariance with respect to future goals through }\phi} - u^{\pi_\Theta}(s_0,\phi(s_w))\right] \tag{7}$$

Then, in Equation (7) we add a goal representation ϕ that processes the goal before going to policy π_Θ. Intuitively, a policy could achieve the out-of-distribution task by first going from s_0 to s_w (in-distribution), and then completing the remaining task s_w to s_f. In essence, we want that when conditioning on $\phi(s_f)$, the policy should first go to s_w, which can be achieved by learning ϕ, where $\phi(s_w)$ is similar to $\phi(s_f)$ (Myers et al., 2025b). More formally, for $s_f \sim M^{\beta}(s_w, s_f)$ we want an invariance $\phi(s_f) \approx \phi(s_w)$. From this observation, we can understand obtaining a representation ϕ related to the successor measure of the mixture policy β can be beneficial.

The BYOL would be a simple framework to learn these representations to capture temporal dependencies. However, in Section 5 we demonstrate that a simple BYOL objective empirically leads to limited generalization when used as an auxiliary loss. Intuitively, a standard BYOL objectives directly approximate one-step transition dynamics, not the successor measure so struggles to capture relationships between distant states, separated by several trajectories.

4.1 BYOL-γ: Connecting self-predictive objectives to the successor representation

To build better self-predictive representations, we propose BYOL-γ which allows us to use the BYOL framework to capture temporally extended information, i.e. *successor representations*. Given a state s_t, a BYOL objective samples prediction targets from one-step transition as in Equation (3). However, we make a modification to predict empirical samples from the normalized successor measure:

$$\mathcal{L}_{\text{BYOL-}\gamma}(\phi,\psi) = \mathbb{E}_{s_t\sim\, p(s),k\sim\text{geom}(1-\gamma),s_{t+k}\sim p^{\pi}(s_{t+k}|s_t)}\left[f(\psi(\phi(s_t)),\bar{\phi}(s_{t+k}))\right] \tag{8}$$

Where f refers to an energy function, ϕ refers to the encoder, and ψ the predictor. With $\gamma = 0$, we have $s_{t+k} = s_{t+1}$ corresponding to an approximation of the one-step transitions, recovering the base BYOL objective. Figure 1b depicts our overall representation learning objective. We can view this objective as iteratively minimizing an upper-bound on the error between $\psi(\phi((s))$ and a target of the true successor features of the policy ψ^{π} with changing basis features $\bar{\phi}$. With convex f, by Jensen's inequality we have:

$$\mathcal{L}_{\text{BYOL-}\gamma} \geq \mathbb{E}_{s_t}\left[f(\psi(\phi(s_t)),\mathbb{E}_{s_+\sim\tilde{M}^{\pi}(s_t,s_+)}\bar{\phi}(s_+))\right] = \mathbb{E}_{s_t}\left[f(\psi(\phi(s_t)),(1-\gamma)\psi^{\pi}_{\bar{\phi}}(s_t)\right] \tag{9}$$

Method	**Approx.** $\tau \sim \beta$	**Approx.** $\tau \sim \{\beta_j\}$	**Batch**
TRA (CL)	$\tilde{M}^\beta(s, s_+)/p^\beta(s_+)$	$\sum_j p(\beta_j \mid s)\tilde{M}^{\beta_j}(s, s_+)/p^\beta(s_+)$	$(s_t, s_+)^B, (s_i, s_j)^{B^2}$
TD-SR	$\tilde{M}^\beta(s, s_+)/p^\beta(s_+)$	$\tilde{M}^\beta(s, s_+)/p^\beta(s_+)$	$(s_t, s_{t+1})^B, (s_i, s_j)^{B^2}$
BYOL	$p^\beta(s_{t+1} \mid s_t)$	$p^\beta(s_{t+1} \mid s_t)$	$(s_t, s_{t+1})^B$
BYOL-γ (ours)	$\tilde{M}^\beta(s, s_+)$	$\sum_j p(\beta_j \mid s)\tilde{M}^{\beta_j}(s_t, s_+)$	$(s_t, s_+)^B$

Table 1: **Auxiliary Representation Objectives**. We provide an overview of the representation objectives we consider. In the first two columns, we label the quantities which representations are approximating in finite MDPs, either from datasets with trajectories collected from a single policy $\tau \sim \beta$, or a mixture of policies $\tau \sim \{\beta_j\}$. We provide additional derivations for mixture datasets in Appendix F. In the last column, we list samples used for each objective, where the superscript denotes the number of loss terms for a pair of samples.

Specifically, we precisely show the relationship of BYOL-γ to the SR with the following result:

Theorem 4.1. *Given a finite MDP with linear representations $\Phi \in \mathbb{R}^{|\mathcal{S}| \times d}$, and predictor $\Psi \in \mathbb{R}^{d \times d}$, under assumptions of orthogonal initialization for Φ (Ass. E.1), a uniform initial state distribution $p_0(s)$ (Ass. E.2), and symmetric transition dynamics (Ass. E.3), minimizing the self-predictive learning objective $\mathcal{L}_{BYOL\text{-}\gamma}(\phi, \psi)$ approximates a spectral decomposition of the successor representation $\tilde{M}^\pi \approx \Phi\Psi\Phi^T$, corresponding to successor features $(1-\gamma)\Psi^\pi \approx \Psi\Phi$.*

Proof is in Appendix E.2, where we show that existing theory (Khetarpal et al., 2025) also translates to the proposed BYOL-γ objective. Finally, we can see the relation between this objective and CL (1), with the most striking difference being the removal of the denominator involving negative samples. Surprisingly, we reveal that this simplified system still captures similar information and also can lead to empirical generalization in Section 5.1 while neither relying on TD learning nor negative samples.

BYOL-γ Variants. We discuss a few variants on our base objective, namely, we evaluate **bidirectional prediction** (Guo et al., 2020; Tang et al., 2023) where we add an additional backwards predictor ψ_b which predicts a past representation from the future. We also utilize an **action-conditioned** variant of the forward predictor $\psi_f(\phi(s_t), a_t)$, which can be interpreted as a temporally extended latent dynamics model, or capturing information about $\tilde{M}^\pi(s, a, s_+)$, giving:

$$\mathcal{L}_{\text{BYOL-}\gamma}(\phi, \psi) = \mathbb{E}_{s_t \sim\, p(s), s_+ \sim \tilde{M}^\pi(s_t, s_+)}\left[f(\psi_f(\phi(s_t), a_t), \bar{\phi}(s_+)) + f(\bar{\phi}(s_t), \psi_b(\phi(s_+))\right] \quad (10)$$

For f, we choose a cross-entropy loss between (softmax) normalized representations, similar to DINO (Caron et al., 2021): $f_{\text{CE}}(a, b) = \text{softmax}(b) \cdot \log \text{softmax}(a)$. We also find a normalized l_2 loss, $f_{l_2} = \|\frac{a}{\|a\|} - \frac{b}{\|b\|}\|_2^2$, commonly used in BYOL setups (Grill et al., 2020; Schwarzer et al., 2020) also works, which we ablate in Section 5.4.

4.2 Training a policy with auxiliary representation

We consider BYOL-γ and other objectives as auxiliary losses for BC policies $\pi_\Theta(a|s, g)$ to improve the generalization of policies. We label all parameters $\Theta = (\theta, \phi, \psi)$, with the parameters of the encoder and predictor correspond to ϕ, ψ. The policy-head (θ) transforms representations to actions via an MLP: $\pi_\Theta(a|s, g) = \text{MLP}_\theta(\text{concat}(\phi(s), \phi(g)))$. With this policy, we train with the objective:

$$\mathbb{E}_{\beta_j \sim p(\beta_j), \tau \sim \beta_j}\left[\mathcal{L}_{\text{BC}}(\Theta) + \alpha\mathcal{L}_{\text{aux}}(\phi, \psi)\right] \quad (11)$$

The term $\mathcal{L}_{\text{BC}}$ updates the parameters of both the policy head θ and its inputs, i.e., the encoder ϕ, while $\mathcal{L}_{\text{aux}}$ updates ψ, ϕ but not θ. With ϕ affected by both terms, the BC loss ensures that the representation is sufficient for action prediction, and prevents collapse which can occur in certain representation learning, such as in BYOL objectives. Additionally, the auxiliary loss prevents overfitting and help generalization for the policy. We provide additional details about the architecture in Appendix A.

We wish to learn representations related to the successor measure of the mixture policy as motivated Equation (7). However, there are trade-offs with the representation learning objectives in terms of the quantities they approximate and the data they use, as shown in Table 1. When we directly have full MC samples from a single policy ($\tau \sim \beta$), TRA, TD-SR, and BYOL-γ each capture information related to its SM. However, in practice, we only have MC samples from individual policies $\tau \sim \{\beta_j\}$, rather than the mixture.

First, we consider using TD-SR as an auxiliary loss for BC, as we can see it still approximates the correct quantity. To our knowledge, we are the first to study this objective as an auxiliary loss for BC. TD-SR explicitly can "stitch" across policies, i.e. $\psi(s_t)\psi(s')$ via $\gamma\bar{\psi}(s_{t+1})\bar{\phi}(s')$ for $s_t, s_{t+1} \sim p^{\beta_i}(s_t)p^{\beta}(s_{t+1}|s_t)$ and $s' \sim p^{\beta_j}(s)$. However, we wish to understand if we can obtain *representations that help with generalization without TD*, as this may be more scalable when applied with policy learning. This leads us to quantify how MC methods behave when trained on mixture datasets.

While 1-step MC methods like BYOL are consistent across dataset composition, we can see that TRA and BYOL-γ approximate different quantities than TD-SR. Namely, *rather than approximating the SM of the mixture policy, MC methods capture a mixture of SRs* as shown in Table 1 and Appendix F. In practice, MC methods still learn relationships between states encountered in different policies by effectively approximating many SRs in a single representation space, which is qualitatively shown in Figure 2. Surprisingly, we show that BYOL-γ can lead to representations that are similar, or even better than TD-SR without utilizing TD. However, we find that CL as used in TRA leads to pessimism in the relationship between states sampled by different policies. Namely, for states that are not in the same trajectory, they will only be paired as negative examples, whose representations are pushed apart. This also shows up in the denominator of its approximation, with normalization from $p^{\beta}(s_+)$ in $\frac{\sum_j p(\beta_j|s)\tilde{M}^{\beta_j}(s,s_+)}{p^{\beta}(s_+)}$. On the other hand, this pessimism is not encountered with BYOL-γ, which does not utilize negative examples, giving an approximation of $\sum_j p(\beta_j|s)\tilde{M}^{\beta_j}(s, s_+)$ by simply predicting latents. Finally, we highlight that BYOL-γ only computes $O(B)$ loss terms, while CL compute $O(B^2)$ negatives (s_i, s_j), and we utilize $O(B^2)$ bootstrap terms with TD-SR.

5 Experiments

Now that we have shown a theoretical basis for studying choices of representations, including CL, TD-SR, BYOL, and our new objective (BYOL-γ), we study how these methods behave empirically. We compare representation learning algorithms across three axes: (1) First, we compare qualitatively whether the representations appear to capture temporal relationships (2) Second, we assess representations quantitatively by measuring zero-shot generalization performance on unseen tasks that require combinatorial generalization (3) Third, we assess generalization performance over an increasing generalization horizon. Finally, we perform ablations on the various components of our proposed method to demonstrate the relative importance of each algorithmic choice.

Environments. We empirically evaluate how well our approach can help with combinatorial generalization on offline goal-reaching tasks on OGBench (Park et al., 2025), which contains both navigation and manipulation tasks, across low-dimensional and visual observations. We focus on navigation environments, where OGBench provides `stitch` datasets, that assess combinatorial generalization by training on trajectories that span at most 4 maze cells, while evaluating on tasks that are longer, requiring combining information from multiple smaller trajectories.

Baselines. We benchmark against non-hierarchical methods that perform control from state to low-level actions (e.g. joint-control). In addition to BYOL-γ used as an auxiliary loss for BC, we evaluate several baselines: **GCBC** is the standard BC baseline, which we aim to improve upon with representation learning. **Offline RL** from OGBench, including implicit {V,Q}-learning (**IVL, IQL**) (Kostrikov et al., 2022), Quasimetric RL (**QRL**)(Wang et al., 2023), and Contrastive RL (**CRL**) (Eysenbach et al., 2022). **BYOL** is a minimal version of our setup with 1-step prediction ($\gamma = 0$), only forwards prediction (ψ_f) without action-conditioning ($\psi_f(\phi(s_t))$), and loss f_{l_2}. **TRA** (Myers et al., 2025b) is an auxiliary representation objective using contrastive learning related to an MC approximation of the SM. **TD-SR** is a TD-based approximation of the SM as in Equation (2) used as an auxiliary objective for BC. We also compare to an n-step version of BYOL in Appendix B.2 and the Forward-Backward (FB) (Touati and Ollivier, 2021; Touati et al., 2023) in Appendix B.3.

Experimental Setup. We match the training details of OGBench, and consider a similar representation learning setup to TRA. We found it was beneficial to add action conditioning to TD-SR, but did not see an overall improvement for TRA, so we use the original setup without action-conditioning.

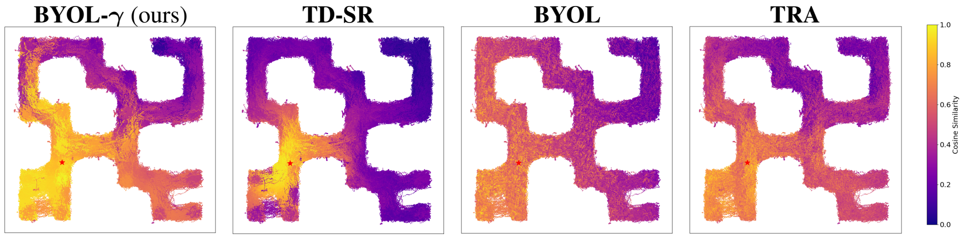


Figure 2: **Visualization of the Learned Representation: depicts the similarity between the prediction of the current state representation to the goal representation.** For **BYOL-γ** and **TD-SR**, we visualize the cosine similarity between $\psi(\phi(s), \cdot)$ or $\psi(s, \cdot)$, to $\phi(g)$ $\forall s \in D$ for a fixed goal g which is indicated by the star marked in red.

While we use policy $\pi(\phi(s), \phi(g))$ and train with action-conditioning for BYOL-γ and TD-SR, TRA originally uses a parameterization $\pi(\psi(s), \phi(g))$ and does not condition on actions. We provide a full comparison for changing $\psi(s)$ to $\phi(s)$ and action-conditioning in Appendix B.1 for TRA. However we obtain similar performance on average with the original setup. For clarity, in Table 2, we utilize superscript a to denote methods with action-conditioning. Notably, we find that the weight of the auxiliary representation learning objectives (α) can be sensitive to both the embodiment, and size of environment (`medium` vs `large`). For each method, we perform a hyperparameter sweep over 4 α values, and report the best result for each environment in Table 2. We hold other hyperparameters constant, except with variation between non-visual and visual noted in Appendix A.

5.1 Qualitative analysis of representations

In Figure 2, we display a qualitative analysis of the representations. We visualize the similarity between the future prediction ψ for each state to $\phi(g)$ for a fixed goal g. We can see that **BYOL-γ** seems to learn a representation that encodes reachability between states, and has a similar structure to **TD-SR**, which is known to approximate the successor measure. **TRA** and base **BYOL** seem to both capture similar structure and learn a less well-defined latent space. However, BYOL-γ and TD-SR have more distinct similarity, and have visible "paths" of similar states. **BYOL-γ** also appears to capture the most similarity among more distant pairs of states. Compared to **TRA**, our hypothesis here is that BYOL-γ has more optimistic similarity between distant states due to the lack of a negative term in the loss, pushing representations apart. We show additional environments in Appendix H.1. We also check the correlation of distance in representation space with shortest paths in the maze in Appendix H.2, showing that BYOL-γ best captures the structure of the environment.

5.2 Zero-shot performance on combinatorial generalization tasks

In Table 2, we provide the performance results across all methods. Overall, our proposed method **BYOL-γ**, shows improved performance vs. **GCBC** across most environments, and is either competi-

Dataset	**BYOL-γ^a**	**BYOL**	**TRA**	**TD-SR^a**	**GCBC**	**GCIVL**	**GCIQL**	**QRL**	**CRL**
`antmaze-medium-stitch`	58 ± 5	59 ± 4	54 ± 6	**64 ± 6**	45 ± 11	44 ± 6	29 ± 6	59 ± 7	53 ± 6
`antmaze-large-stitch`	19 ± 7	17 ± 6	11 ± 8	**23 ± 4**	3 ± 3	18 ± 2	7 ± 2	18 ± 2	11 ± 2
`humanoidmaze-medium-stitch`	**51 ± 6**	23 ± 3	45 ± 8	42 ± 4	29 ± 5	12 ± 2	12 ± 3	18 ± 2	36 ± 2
`humanoidmaze-large-stitch`	**13 ± 3**	3 ± 1	5 ± 4	11 ± 3	6 ± 3	1 ± 1	0 ± 0	3 ± 1	4 ± 1
`antsoccer-arena-stitch`	**25 ± 5**	12 ± 7	14 ± 4	22 ± 10	**24** ± 8	21 ± 3	2 ± 0	1 ± 1	1 ± 0
`visual-antmaze-medium-stitch`	**68 ± 4**	57 ± 8	52 ± 3	49 ± 2	**67** ± 4	6 ± 2	2 ± 0	0 ± 0	69 ± 2
`visual-antmaze-large-stitch`	26 ± 5	26 ± 5	17 ± 1	**29 ± 2**	24 ± 3	1 ± 1	0 ± 0	1 ± 1	11 ± 3
`visual-scene-play`	**17 ± 1**	13 ± 3	**16** ± 3	14 ± 1	12 ± 2	25 ± 3	12 ± 2	10 ± 1	11 ± 2
`average-state`	**33**	23	26	**32**	21	19	10	20	21
`average-visual`	**37**	32	28	31	34	11	5	4	30
`average-all`	**35**	26	27	32	26	16	8	14	25

Table 2: **OGBench: We find that BYOL-γ performs better overall compared to prior methods**. We report mean and standard deviation over 10 training seeds in non-visual environments, and 4 seeds in visual environments. We match the OGBench evaluation setup of 5 evaluation (state,goal) tasks, and 50 episodes per task. The success rate is then averaged over the last 3 checkpoints. We color the best non-RL method, and **bold** values within 95% of its value in the same row. We use superscript a to denote methods utilizing action-conditioning.

tive with or outperforms **TD-SR** and **TRA**. Importantly, we find that a minimal **BYOL** setup does not confer significant benefit over the base **GCBC** except in non-visual `antmaze` environments. Generally, auxiliary representation learning with **GCBC** outperforms existing offline RL methods.

Within the auxillary loss methods, we find that **TD-SR** and **BYOL-γ** tend to outperform **TRA** on most environments. While we find that **TD-SR** outperforms **BYOL-γ** on environments with smaller state spaces (`antmaze-{medium, large}`), we find that **BYOL-γ**'s simpler training procedure is beneficial in environments with larger state spaces (`humanoidmaze-{medium, large}`, `visual-antmaze-medium` and `visual-scene-play`).

Interestingly, in `visual-antmaze` **TRA** and **TD-SR** actually seem to hurt performance in comparison to base **GCBC**. On the other hand, with **BYOL-γ** we see no performance degradation over **GCBC** on the visual environments, a considerable improvement over other methods. In Appendix C, we extend BYOL-γ to the hierarchical setting (**HBYOL-γ**), where we obtain significant improvement over BC baselines, including on visual maze environments.

We also find a relationship between success and representation quality (Section 5.1). Namely, in Table 12 we calculate the correlation of representations to shortest path distances and success rate over these same checkpoints. We see that the ranking of methods in terms of average correlation to shortest path (`average maze correlation`) in representation space matches the ordering of methods in terms of average empirical policy success (`average maze success`).

5.3 Evaluating generalization with increasing horizon

We conduct experiments to understand how success rate changes as an agent has to reach more challenging goals further away from its starting position. For each `maze` environment, we consider the same base 5 evaluation tasks used in Table 2, but construct intermediate waypoints along the shortest path to the final goal determined by breadth-first search. We also include an additional `maze` environment, `giant` on which all methods have zero success rates to reach distant goals. This gives a more holistic view on an agent's performance.

We display results in Figure 3 and Appendix G, where we can see how performance drops off for all methods after a generalization threshold denoted by the red bar. While all methods cannot fully reach distant goals on `giant`, we see that **BYOL-γ** has the slowest drop-off in performance. We note that this is a challenging task, that requires stitching up to approximately 8 different trajectories.

5.4 Components affecting generalization

We ablate key components of the **BYOL-γ** objective in Table 3. This includes removing action conditioning for forward predictor ψ_f $(-a)$, swapping the loss from cross-entropy to normalized squared l_2 norm (f_{l_2}), removing backwards predictor ψ_b, and predicting the representation of the adjacent state $(\gamma = 0)$. Both removing action-conditioning, and backwards prediction overall lead to similar results, but variability per-environment. For f_{l2}, we obtain slightly worse average performance, and for $\gamma = 0$, we see the largest drop-off, especially on `humanoidmaze`.

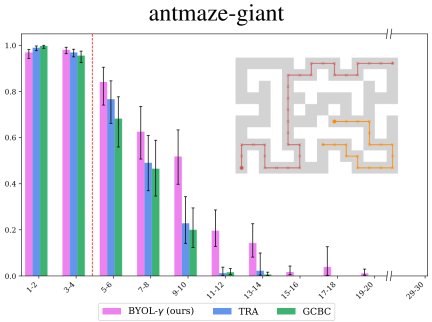


Figure 3: **Evaluating Generalization with Increasing Horizons**: shows that **BYOL-γ** not only performs well on goals in the near horizon, but also, helps to generalize well to goals requiring stitching, after the red bar (> 4).

6 Discussion

Limitations. While we demonstrate that BYOL-γ and other representation learning objectives offer a promising recipe for obtaining combinatorial generalization, we find that there still exists a generalization gap, especially on challenging navigation environments e.g. `giant`. We also find a less significant improvement over BC on visual environments, which may motivate additional

Dataset	**BYOL-γ^a**	$-a$	f_{l_2}	$-\psi_b$	$\gamma = 0$
antmaze-medium-stitch	61 ± 6	63 ± 9	56 ± 4	**67** ± 2	59 ± 5
antmaze-large-stitch	21 ± 5	**27** ± 7	24 ± 6	19 ± 7	8 ± 4
humanoidmaze-medium-stitch	**54** ± 5	48 ± 5	49 ± 6	**52** ± 5	18 ± 2
humanoidmaze-large-stitch	**14** ± 2	12 ± 6	**15** ± 7	13 ± 2	3 ± 1
antsoccer-arena-stitch	21 ± 4	20 ± 5	11 ± 5	**27** ± 7	**25** ± 7
visual-antmaze-medium	**68** ± 4	**65** ± 3	63 ± 5	61 ± 4	54 ± 9
visual-antmaze-large	26 ± 5	25 ± 8	**27** ± 7	**28** ± 2	**28** ± 1
average	33	33	31	33	24

Table 3: **BYOL-γ ablations.** For each ablation, we perform sweep over α, and report the best result per-environment. For all environments, we report results over 4 seeds (for **BYOL-γ**, we use the first 4 of 10 in Table 2).

investigation. Additionally, we may anticipate more benefit from representation learning when applied to larger visual datasets, which has been fruitful in other domains.

Conclusion. In this work, we provide a stronger understanding of the relationship between quantities related to successor representations and the generalization of policies trained with behavioral cloning through a unified understanding of objectives. We propose a new self-predictive representation learning objective, BYOL-γ, and show that it captures information related to the successor measure, resulting in a competitive choice of an auxiliary loss for better generalization. We demonstrate that augmenting behavior cloning with meaningful representations results in new capabilities such as improved combinatorial generalization, especially in larger and more complex environments.

References

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017. URL https://arxiv.org/abs/1606.05312.

Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint, 2021. URL https://arxiv.org/abs/2101.07123.

David Brandfonbrener, Alberto Bietti, Jacob Buckman, Romain Laroche, and Joan Bruna. When does return-conditioned supervised learning work for offline reinforcement learning? In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=XByg4kotW5.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021. doi: 10.1109/ICCV48922.2021.00951.

Wilka Carvalho, Momchil S. Tomov, William de Cothi, Caswell Barry, and Samuel J. Gershman. Predictive representations: Building blocks of intelligence. *Neural Computation*, 36(11):2225–2298, 10 2024. ISSN 0899-7667. doi: 10.1162/neco_a_01705. URL https://doi.org/10.1162/neco_a_01705.

Yash Chandak, Shantanu Thakoor, Zhaohan Daniel Guo, Yunhao Tang, Remi Munos, Will Dabney, and Diana L Borsa. Representations and exploration for deep reinforcement learning using singular value decomposition. In *International Conference on Machine Learning*, pages 4009–4034. PMLR, 2023. URL https://arxiv.org/abs/2305.00654.

Ian Char, Viraj Mehta, Adam Villaflor, John M. Dolan, and Jeff Schneider. Bats: Best action trajectory stitching, 2022. URL https://arxiv.org/abs/2204.12026.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=a7APmM4B9d.

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4):613–624, 1993. doi: 10.1162/neco.1993.5.4.613.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. Rvs: What is essential for offline RL via supervised learning? In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=S874XAIpkR-.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

Jesse Farebrother, Joshua Greaves, Rishabh Agarwal, Charline Le Lan, Ross Goroshin, Pablo Samuel Castro, and Marc G Bellemare. Proto-value networks: Scaling representation learning with auxiliary tasks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=oGDKSt9JrZi.

Vincent François-Lavet, Yoshua Bengio, Doina Precup, and Joelle Pineau. Combined reinforcement learning via abstract representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3582–3589, 2019. URL https://arxiv.org/abs/1809.04506.

Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Diffusion guidance is a controllable policy improvement operator. *arXiv preprint arXiv:2505.23458*, 2025.

Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann LeCun. Learning and leveraging world models in visual representation learning, 2024. URL https://arxiv.org/abs/2403.00504.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International conference on machine learning*, pages 2170–2179. PMLR, 2019. URL https://arxiv.org/abs/1906.02736.

Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R. Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems*, 2024. URL https://doi.org/10.15607/RSS.2024.XX.090.

Raj Ghugare, Matthieu Geist, Glen Berseth, and Benjamin Eysenbach. Closing the gap between TD learning and supervised learning - a generalisation point of view. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2401.11237.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020. URL https://arxiv.org/abs/2006.07733.

Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Remi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3875–3886. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/guo20g.html.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning (ICML)*, 2022. URL https://arxiv.org/abs/2203.04955.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Xinyu Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Khuong Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A large-scale in-the-wild robot manipulation dataset. In *RSS 2024 Workshop: Data Generation for Robotics*, 2024. URL `https://openreview.net/forum?id=Ml2pTYLNLi`.

Khimya Khetarpal, Zhaohan Daniel Guo, Bernardo Avila Pires, Yunhao Tang, Clare Lyle, Mark Rowland, Nicolas Heess, Diana L Borsa, Arthur Guez, and Will Dabney. A unifying framework for action-conditional self-predictive reinforcement learning. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. URL `https://arxiv.org/abs/2406.02035`.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024. URL `https://arxiv.org/abs/2406.09246`.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL `https://arxiv.org/abs/2110.06169`.

Charline Le Lan, Stephen Tu, Adam Oberman, Rishabh Agarwal, and Marc G. Bellemare. On the generalization of representations in reinforcement learning, 2022. URL `https://arxiv.org/abs/2203.00543`.

Yann LeCun. A path towards autonomous machine intelligence version, 2022. URL `https://openreview.net/forum?id=BZ5a1r-kVsf`.

Jaewoo Lee, Sujin Yun, Taeyoung Yun, and Jinkyoo Park. GTA: Generative trajectory augmentation with guidance for offline reinforcement learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=kZpNDbZrzy`.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020. URL `https://arxiv.org/abs/2005.01643`.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Bo Liu, Yihao Feng, Qiang Liu, and Peter Stone. Metric residual network for sample efficient goal-conditioned reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8799–8806, 2023. URL `https://arxiv.org/abs/2208.08133`.

Cong Lu, Philip J. Ball, Yee Whye Teh, and Jack Parker-Holder. Synthetic experience replay. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=6jNQ1AY1Uf`.

Yunhao Luo, Utkarsh A. Mishra, Yilun Du, and Danfei Xu. Generative trajectory stitching through diffusion composition, 2025. URL `https://arxiv.org/abs/2503.05153`.

Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=xQiYCmDrjp.

Vivek Myers, Catherine Ji, and Benjamin Eysenbach. Horizon Generalization in Reinforcement Learning. In *International Conference on Learning Representations*, January 2025a. URL https://arxiv.org/pdf/2501.02709.

Vivek Myers, Bill Chunyuan Zheng, Anca Dragan, Kuan Fang, and Sergey Levine. Temporal representation alignment: Successor features enable emergent compositionality in robot instruction following, 2025b. URL https://arxiv.org/abs/2502.05454.

Tianwei Ni, Benjamin Eysenbach, Erfan Seyedsalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive rl. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2401.08898.

Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen

Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions. *Advances in Neural Information Processing Systems*, 36:34866–34891, 2023.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2007.05929.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. URL https://arxiv.org/abs/2103.00020.

Jonathan Richens, David Abel, Alexis Bellot, and Tom Everitt. General agents contain world models, 2025. URL https://arxiv.org/abs/2506.01622.

Juergen Schmidhuber. Reinforcement learning upside down: Don't predict rewards – just map them to actions, 2020. URL https://arxiv.org/abs/1912.02875.

Max Schwarzer, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron C. Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/2007.05929.

Vlad Sobal, Wancong Zhang, Kynghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models, 2025. URL https://arxiv.org/abs/2502.14819.

Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Avila Pires, Yash Chandak, Rémi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, et al. Understanding self-predictive learning for reinforcement learning. In *International Conference on Machine Learning*, pages 33632–33656. PMLR, 2023. URL https://arxiv.org/abs/2212.03319.

Andrea Tirinzoni, Ahmed Touati, Jesse Farebrother, Mateusz Guzek, Anssi Kanervisto, Yingchen Xu, Alessandro Lazaric, and Matteo Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models. *arXiv preprint arXiv:2504.11054*, 2025.

Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23, 2021.

Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MYEap_OcQI.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

Claas A. Voelcker, Tyler Kastner, Igor Gilitschenski, and Amir-massoud Farahmand. When does self-prediction help? understanding auxiliary tasks in reinforcement learning. *Reinforcement Learning Conference*, August 2024. URL https://arxiv.org/abs/2406.17718.

Tongzhou Wang and Phillip Isola. Improved representation of asymmetrical distances with interval quasimetric embeddings. In *NeurIPS 2022 Workshop on Symmetry and Geometry in Neural Representations*, 2022. URL https://arxiv.org/abs/2211.15120.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*. PMLR, 2023. URL https://arxiv.org/abs/2304.01203.

Taku Yamagata, Ahmed Khalil, and Raúl Santos-Rodríguez. Q-learning decision transformer: leveraging dynamic programming for conditional sequence modelling in offline rl. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in neural information processing systems*, 34:25476–25488, 2021. URL https://arxiv.org/abs/2111.00210.

Zhaoyi Zhou, Chuning Zhu, Runlong Zhou, Qiwen Cui, Abhishek Gupta, and Simon Shaolei Du. Free from bellman completeness: Trajectory stitching via model-based return-conditioned supervised learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.19308.

A Experimental Setup

Table 4: Hyperparameters for BYOL-γ

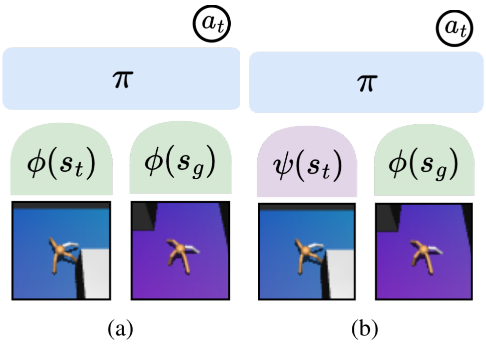
Hyperparameter	Shared
actor head	MLP (512,512,512)
representation encoder (ϕ)	MLP (64,64,64)
predictor (ψ)	MLP (64,64,64)
encoder ensemble	2
learning rate	3×10^{-4}
optimizer	Adam

	Non-visual	Visual
Gradient steps	1000k	500k
Batch size	1024	256
τ (EMA)	1.0	0.99
γ	0.99	{0.66, 0.99}
α (alignment)	{1,6,40,100}	{1,6,10,20}
additional encoder	n/a	impala_small
encoder output dimension	$\|s\|$	64

A.1 Implementation Details

In this section we provide more training details for BYOL-γ, and representation learning baselines. We match the training details of OGBench, including gradient steps, batch size, learning rate.

Network Architecture. We follow the same general setup as TRA, where we utilize MLP-based encoders, and action head. For the output dimension of the encoder, we use the state dimension for non-visual experiments, and 64 for visual experiments. For the predictor ψ, we utilize an MLP of the same architecture as the encoder. For image-based tasks, there is an additional CNN, which then passes output to the MLP encoder.



(a) (b)

Figure 4: **Encoder Variation**. When training with BYOL, BYOL-γ and TD-SR, we utilize policies with architecture (a) which uses ϕ to process states and goals. We utilize architecture (b) for TRA to match prior implementation, however in Appendix B.1 we train TRA with architecture (a) and action-conditioning.

Representation Ensemble. We follow the setup of TRA which utilizes representation ensembling, such that two copies of the encoder ϕ_1, ϕ_2 are in parallel. We also have two distinct predictors ψ_1, ψ_2 for each ensemble. As input to the policy head, we average the representations, $\bar{z} = \frac{\phi_1(s_t)+\phi_2(s_2)}{2}$. Each representation is trained independently for the BYOL loss, but the BC loss differentiates through both ϕs.

Alignment. We find that the choice of weight of the auxiliary loss for the representation learning objective is sensitive to both the robot embodiment and the environment size. For comparison, we perform a hyperparameter search over four alignment values for BYOL-γ, TRA, and TD-SR, and then report the best value for each environment in Table 2.

Discount. For sampling the next-state, we utilize a discount factor of $\gamma = 0.99$ for all non-visual environments. For visual environments, we perform a hyperparameter search over $\{0.66, 0.99\}$, however all representation learning methods performed better at $\gamma = 0.66$.

A.2 BYOL-γ

Target network. For BYOL, we find that exponential moving average (EMA) target networks for the encoder ϕ are not necessary for non-visual environments ($\tau = 1$), but for visual environments, we find that a fast target stabilizes training ($\tau = 0.99$):

$$\phi_{\text{target}} = \tau \phi_{\text{online}} + (1 - \tau)\phi_{\text{target}}$$

A.3 TRA

In practice, TRA uses a symmetric version (Radford et al., 2021) of the InfoNCE objective discussed in Equation 1. We write this in batch form, $\mathcal{B} = \{(s_i, s_{+,i})\}_{i=1}^{|\mathcal{B}|}$ rather than in expectation:

$$\mathcal{L}_{\text{TRA}} = \mathbb{E}_{\mathcal{B}} \left[-\frac{1}{B} \sum_{i=1}^{|\mathcal{B}|} \log \frac{e^{f(\psi(s_i), \phi(s_{+,i}))}}{\sum_{j=1}^{|\mathcal{B}|} e^{f(\psi(s_i), \phi(s_{+,j}))}} - \frac{1}{B} \sum_{i=1}^{|\mathcal{B}|} \log \frac{e^{f(\psi(s_i), \phi(s_{+,i}))}}{\sum_{j=1}^{|\mathcal{B}|} e^{f(\psi(s_j), \phi(s_{+,i}))}} \right] \tag{12}$$

Additionally, TRA minimizes the squared norm of representations $\min_{\phi,\psi} \lambda \mathbb{E}_s[\frac{\|\phi(s)\|^2}{d} + \frac{\|\psi(s)\|^2}{d}]$ with $\lambda = 10^{-6}$. For TRA, we search over $\alpha = \{10, 40, 60, 100\}$.

A.4 TD-SR

Prior work similar to TD-SR, which trains FB for zero-shot policy optimization (Touati et al., 2023) typically normalizes ϕ with an additional loss term so that $\mathbb{E}\left[\phi\phi^T\right] \approx I_d$. However, we found that adding this loss term was not beneficial to performance in our setting and hence do not include it.

TD-SR uses an EMA target network as described in A.2 with $\tau = 0.005$. For TD-SR, we search over $\alpha = \{0.01, 0.05, 0.001, 0.005\}$.

A.5 Code.

We utilize the OGBench (Park et al., 2025) codebase and benchmark, and its extensions in the TRA codebase (Myers et al., 2025b) for equal comparison.

A.6 Compute Requirements

We perform all experiments utilizing single GPUs, predominately NVIDIA RTXA8000 and L40S. We utilize 6 CPU cores, 24G of RAM for non-visual environments, and 64G for visual experiments. Experiments take 2 to 4 hours for non-visual and 6 to 12 hours for visual environments.

B Ablations.

B.1 Action-conditioning

In this section, we ablate the component of performing action-conditioning for the predictor $\psi(s_t)$ vs $\psi(s_t, a_t)$ for TRA and TD-SR. We consider a similar comparison for BYOL-γ in Table 3. For this comparison, when we perform action-conditioning, we utilize a policy representation $\pi(s = \phi(s), g = \phi(g))$ as we have predictor $\psi(s, a)$, and otherwise $\pi(s = \psi(s), g = \phi(g))$ as in the original TRA implementation. We find that results can be environment specific. On average, results are not improved for TRA, but we find an improvement for TD-SR, hence in our main Table 2 we include the action-conditioned results for TD-SR and the action-free results for TRA to match the original implementation.

Dataset	**TRA**	**TRA^a**	**TD-SR**	**TD-SR^a**
antmaze-medium-stitch	54 ± 6	57 ± 12	$\mathbf{64} \pm 10$	$\mathbf{64} \pm 6$
antmaze-large-stitch	11 ± 8	7 ± 7	17 ± 6	$\mathbf{23} \pm 4$
humanoidmaze-medium-stitch	$\mathbf{45} \pm 8$	$\mathbf{45} \pm 5$	36 ± 3	$\mathbf{42} \pm 4$
humanoidmaze-large-stitch	5 ± 4	9 ± 4	6 ± 2	$\mathbf{11} \pm 3$
antsoccer-arena-stitch	14 ± 4	$\mathbf{25} \pm 8$	17 ± 5	22 ± 10
visual-antmaze-medium-stitch	$\mathbf{52} \pm 3$	33 ± 4	47 ± 5	$\mathbf{49} \pm 2$
visual-antmaze-large-stitch	17 ± 1	22 ± 5	$\mathbf{28} \pm 3$	$\mathbf{29} \pm 2$
visual-scene-play	16 ± 3	$\mathbf{18} \pm 2$	12 ± 2	14 ± 1
average	27	27	28	32

Table 5: **Action-conditioning ablations.** We ablate the choice to condition on the first action for predictor ψ for TRA and FB over 10 seeds for non-visual and 4 seeds for visual environments.

Dataset	**BYOL-γ^a**	$-\psi_b$	$\gamma = 0$	$-\psi_b, \gamma = 0$	**BYOL$^a_{n=1}$**	**BYOL$^a_{n=3}$**	**BYOL$^a_{n=5}$**
antmaze-medium-stitch	61 ± 6	$\mathbf{67} \pm 2$	59 ± 5	60 ± 5	60 ± 8	60 ± 7	58 ± 7
antmaze-large-stitch	$\mathbf{21} \pm 5$	19 ± 7	8 ± 4	13 ± 5	19 ± 4	8 ± 4	3 ± 3
humanoidmaze-medium-stitch	$\mathbf{54} \pm 5$	$\mathbf{52} \pm 5$	18 ± 2	27 ± 7	33 ± 4	32 ± 2	20 ± 1
humanoidmaze-large-stitch	$\mathbf{14} \pm 2$	$\mathbf{13} \pm 2$	3 ± 1	5 ± 2	3 ± 2	3 ± 1	5 ± 2
antsoccer-arena-stitch	21 ± 4	$\mathbf{27} \pm 7$	25 ± 7	23 ± 6	25 ± 12	11 ± 7	13 ± 9
average	34	36	23	26	28	23	20

Table 6: **N-step BYOL ablations**. We ablate the BYOL-γ to an n-step BYOL baseline, where we report results over 4 seeds.

B.2 N-STEP BYOL

We provide an additional BYOL-based baseline that utilizes n-step next-representation recurrent prediction, while BYOL-γ uses non-recurrent prediction. We utilize the same BYOL-γ architecture with forward prediction, but with the following objective, computing loss with n terms, where ψ_f^n is shorthand for n recurrent calls:

$$\mathcal{L} = f(\psi_f(\phi(s_t, a_t), \bar{\phi}(s_{t+1})) + f(\psi_f(\psi_f(\phi(s_t, a_t), a_{t+1}), \bar{\phi}(s_{t+2})) + \cdots + f(\psi_f^n(\cdot), \bar{\phi}(s_{t+n}))$$

Theoretically, in a finite MDP, we can interpret this objective of capturing information up to n-step transitions (Tang et al., 2023), i.e. information related to $\{P_a, P_a^2, \cdots, P_a^n\}$ is captured by loss terms $\{\psi_f, \psi_f^2, \cdots, \psi_f^n\}$ respectively. As BYOL-γ captures information related to $\tilde{M}^\pi = (1-\gamma)\sum_{t\geq 0}\gamma^t P_\pi^t$, these two objectives match in theory at $\gamma = 0$. In practice, as n-step operates recurrently, we are constrained to a shorter n which limits the ability for learning long-horizon information.

Empirically, we report comparison of n-step with $n = \{1, 3, 5\}$ to BYOL-γ in Table 6. We validate our n-step implementation in the base case ($n = 1$) with the ablation $\{-\psi_b, \gamma = 0\}$ to BYOL-γ making them equivalent. With increased multi-step prediction (as we increase n), we find worse performance on average.

B.3 FORWARD-BACKWARD ALGORITHM

As an alternative to GCBC, we could instead consider the full Forward-Backward (FB) algorithm for zero-shot goal-reaching, as proposed in Touati et al. (2023). Here, instead of conditioning the policy on a goal representation $\phi(g)$, we instead condition ψ on a vector z such that jointing learning ϕ and ψ produces a *policy-dependent* successor representation where

$$M^{\pi_z}(s, a, s_+) = \psi(s, a, z)^\top \phi(s_+) \cdot p(s_+), \quad \text{and} \quad \pi_z(a \mid s) := \operatorname*{argmax}_a F(s, a, z)^\top z, \tag{13}$$

ψ and ϕ can be learned through a TD relationship analogous to Equation 2, additionally sampling vectors z according to some distribution. In the discrete setting, the policy can be derived directly from Equation 13. In the continuous setting, Touati and Ollivier (2021) additionally learn a policy network $\pi(s, z)$, trained to maximize $F(s, a, z)^\top z$, in a DDPG-style (Lillicrap et al., 2015) algorithm.

At inference time, a policy for a goal state g can be obtained by first encoding the goal state to the z-representation space using the relationship $z = \mathbb{E}_{s\sim\beta}\left[r(s)\phi(s)\right]$, which implies $z = \phi(g)$ for goal-reaching tasks.

For these experiments, we follow the z sampling method from Touati and Ollivier (2021) by using a 50-50 mixture of states s sampled from β and encoded to $z = \phi(s)$ and vectors sampled uniformly on a sphere of radius $\sqrt{d}$ where d is the latent dimension. We use network architectures for ϕ and ψ matching those used in the implementation of FB provided in Tirinzoni et al. (2025), however we keep the number and size of the hidden layers as well as the latent dimension consistent with our implementations of other methods. Additionally, we add a BC-loss to the policy loss as a regularization, with coefficient 1. We sweep the learning rate over three values 10^{-4}, 10^{-5} and 10^{-6} and selected the best performing, averaged over four seeds, for each environment.

In Table 7 we compare our proposed BC with auxiliary loss methods (**TD-SR**^{a} and **BYOL-**γ^a), which use successor measure learning as an auxiliary loss for BC, to value-based methods which instead use a goal-conditioned value function (**GCIQL**) or successor measure (**FB**) to learn a policy through RL. We find that auxiliary loss methods significantly outperform across almost all environments.

Dataset	**BYOL-**γ^a	**TD-SR**^{a}	**FB**	**GCIQL**
`antmaze-medium-stitch`	61 ± 6	**64** $\pm$ **6**	36 ± 5	29 ± 6
`antmaze-large-stitch`	21 ± 5	22 ± 3	5 ± 4	7 ± 2
`humanoidmaze-medium-stitch`	**54** $\pm$ **5**	41 ± 5	26 ± 5	12 ± 3
`humanoidmaze-large-stitch`	**14** $\pm$ **2**	12 ± 3	2 ± 1	0 ± 0
`antsoccer-arena-stitch`	**21** $\pm$ **4**	18 ± 12	19 ± 4	2 ± 0
`average`	**34**	31	17	10

Table 7: **Forward-Backward Algorithm**. We compare our proposed GCBC with auxiliary loss methods (**TD-SR**^{a} and **BYOL-**γ^a) to an implementation of the Forward-Backward algorithm (**FB**) and an offline RL method **GCIQL**, both of which learn an actor which maximizes a goal-conditioned value function. We report best results from a hyperparameter sweep, averaged over four seeds

B.4 Constant Encoder Output Dimension

Our main experimental setup utilized in Table 2 and other experiments utilize an encoder output dimension of size equal to the state dimension, corresponding to `ant` $|s| = 29$, and `humanoid` $|s| = 69$ for non-visual environments. We perform an additional comparison in Table 8 using a fixed size latent dimension $= 64$, matching the latent dimension used for visual environments. We can see that the larger latent dimension helps performance for each method on `antmaze`. Generally, we see similar trends to our prior experiments, such as BYOL-γ performing stronger in `humanoidmaze` experiments, while FB performs stronger on `antmaze`.

Dataset	**BYOL-**γ^a	**TD-SR**^{a}	**TRA**	**TRA (Myers et al., 2025b)**
`antmaze-medium-stitch`	64 ± 7	**73** $\pm$ **8**	67 ± 6	61 ± 3
`antmaze-large-stitch`	18 ± 7	24 ± 9	15 ± 10	13 ± 2
`humanoidmaze-medium-stitch`	**48** $\pm$ **7**	41 ± 3	41 ± 5	**46** $\pm$ **2**
`humanoidmaze-large-stitch`	**12** $\pm$ **5**	10 ± 2	4 ± 2	9 ± 1
`antsoccer-arena-stitch`	**21** $\pm$ **10**	12 ± 4	18 ± 5	17 ± 1
`average`	**33**	32	29	29

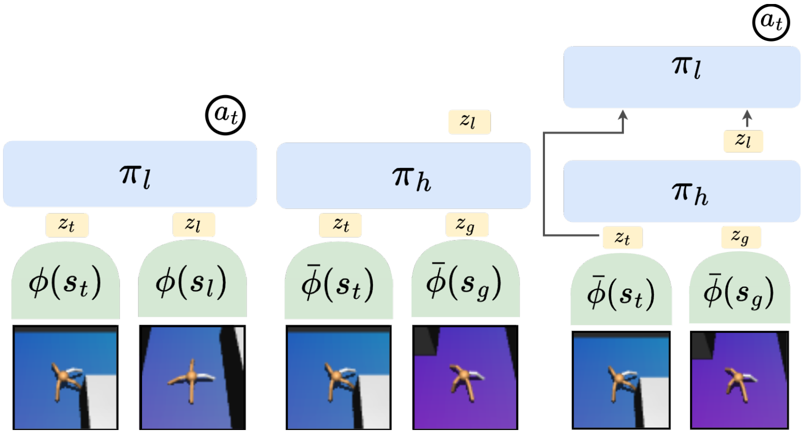
Table 8: **Constant Encoder Output Dimension**. We conduct an ablation repeating our experimental setup for representation learning methods with a constant encoder output dimension at 64. For reference, we also report results from Myers et al. (2025b).

B.5 GCBC Encoder Ablation

We perform an ablation where we use the same architecture as representation learning methods for GCBC. Standard GCBC learns a shared state-goal encoder, $\phi(s, g)$, while representation learning methods pass inputs through an encoder separately, $\phi(s)$, $\phi(g)$ and representations are concatenated and fed to an action head. In our main results, we report OGBench GCBC results with shared state-goal encoder, as this is a stronger baseline. However, to better illustrate the impact that auxiliary loss learning has on GCBC performance, in Table 9, we report GCBC results for an architecture matching representation learning methods (GCBC-ϕ). We especially see a difference in visual environments, where state, goal are stacked ($64 \times 64 \times 6$) before going through the CNN.

Dataset	GCBC	GCBC-ϕ
`antmaze-medium-stitch`	45 ± 11	33 ± 5
`antmaze-large-stitch`	3 ± 3	5 ± 4
`humanoidmaze-medium-stitch`	29 ± 5	32 ± 6
`humanoidmaze-large-stitch`	6 ± 3	4 ± 3
`visual-antmaze-medium-stitch`	67 ± 4	37 ± 6
`visual-antmaze-large-stitch`	24 ± 3	4 ± 3
`visual-scene-play`	12 ± 2	10 ± 1
`average non-visual`	**21**	19
`average visaul`	**34**	17
`average`	**27**	18

Table 9: **GCBC Encoder Ablation.**



(a) Training

(b) Evaluation

Figure 5: **Architecture for HBYOL-γ.** During training (a), we first train a low-level policy with BYOL-γ. Then, we train a high-level policy with a similar procedure to π_h in HGCBC, but using the representation space ϕ. To train π_h, we freeze ϕ, labeled $\bar{\phi}$, and use it for encoding inputs, and the output space, where the π_h predicts the representation of the sub-goal: $z_l = \bar{\phi}(s_l)$. During evaluation (b), π_h first predicts a sub-goal representation, which is then passed to the π_h, where both policies utilize a common state representation.

Dataset	**GCBC**	**HGCBC**	**HGCBC-ϕ**	**BYOL-γ^a**	**HBYOL-γ^a**	**HIQL**
`antmaze-medium-stitch`	45 ± 11	60 ± 4	·	61 ± 6	**76** ± 12	94 ± 1
`antmaze-large-stitch`	3 ± 3	11 ± 8	·	21 ± 5	**29** ± 9	67 ± 5
`humanoidmaze-medium-stitch`	29 ± 5	35 ± 4	·	54 ± 5	**61** ± 2	88 ± 2
`humanoidmaze-large-stitch`	6 ± 3	4 ± 0	·	14 ± 2	**21** ± 3	28 ± 3
`visual-antmaze-medium-stitch`	67 ± 4	·	74 ± 6	68 ± 4	**84** ± 8	87 ± 2
`visual-antmaze-large-stitch`	24 ± 3	·	19 ± 1	26 ± 5	**31** ± 3	28 ± 2
`visual-scene`	12 ± 2	·	8 ± 3	**17** ± 1	14 ± 2	49 ± 4
`average-nonvisual`	21	28	·	38	**47**	69
`average-visual`	34	·	34	37	**43**	55
`average`	27	·	·	37	**45**	63

Table 10: **Hierarchical BC with BYOL-γ.** We report performance averaged over 4 seeds, for HGCBC, HGCBC$-\phi$, and (H)BYOL-γ. We report GCBC and HIQL results from OGBench. We **highlight** the the best performing BC methods, **bold** for methods within 95% of the BC, and **darker highlight** for BC methods which are within 95% or better than HIQL.

C Hierarchical Policies

Although we focus on the impact of representation learning in "flat" learning methods, hierarchical policies are also an effective orthogonal direction for improving generalization to longer horizon tasks. We demonstrate that BYOL-γ also improves on a hierarchical GCBC setup (HGCBC) used in Frans et al. (2025). With HGCBC, we train a high-level policy $\pi^h(l \ |s, g)$ that predicts sub-goals l, and a low-level policy conditioned on sub-goals $\pi^h(a \mid s, l)$, which are both trained with BC. Using BYOL-γ, we implement its hierarchical version, **HBYOL-γ**, as follows: (1) perform our standard BYOL-γ setup, which produces $\pi^l(a \mid \phi(s), \phi(l))$ and (2) train a hierarchical policy in the existing latent space of the low-level policy $\pi^h(\bar{\phi}(l) \mid \bar{\phi}(s), \bar{\phi}(g))$, where $\bar{\phi}$ denotes that the representation is fixed for the high-level policy. **HBYOL-γ^a** allows for both policies to operate in a shared representation space, and avoids state reconstruction performed by standard HGCBC. Prior work does not implement HGCBC in visual settings which would require predicting in pixel space. As a baseline, we implement HGCBC-ϕ, which avoids pixel prediction by using GCBC-ϕ (Appendix B.5). This matches our HBYOL-γ architecture, and two-stage setup but without representation learning on ϕ. For HGCBC-ϕ, we also found it was better to only use representations ϕ for the output space of π_h, and to train a shared input encoder from scratch.

In Table 10, we compare between these BC setups and also report results of hierarchical implicit Q-learning (HIQL) (Park et al., 2023). For HGCBC methods, we use a sub-goal (l) step of 25, listing other hyperparamters in Table 11. We see that HBYOL-γ is the strongest BC setup, outperforming non-hierarchical BYOL-γ and HGCBC. HBYOL-γ is also competitive with HIQL, especially on `visual-antmaze` environments.

Table 11: Additional hyperparameters for HGCBC, HGCBC-ϕ, HBYOL-γ. For other hyperparameters we match those in Table 4. For high-level policies π_h that predict in representation space (HGCBC-ϕ, HBYOL-γ), we find it is better to use a smaller learning rate.

Hyperparameter	**Value**
Hierarchical head	MLP (512, 512, 512, 512)
Low-level head	MLP (512, 512, 512)
Sub-goal steps	25
Learning rate	3×10^{-4} (HGCBC), 10^{-4} (HGCBC-ϕ, HBYOL-γ)

D CL TO TD-SR

Here, illustrate that connection between CL and TD-SR, showing that in the limit an n-step version of TD-SR becomes similar to CL.

We can rewrite Equation (1) to see the connection between TD-SR and CL (MC). Under assumptions that f is the dot product between ϕ and ψ, and ϕ, ψ are centered, if we apply a second-order Taylor expansion to the denominator of the CL loss (Touati et al., 2023) we have:

$$\text{CL}_{\text{InfoNCE}} \approx \mathbb{E}_{s\sim p, s'\sim p}\left[(\psi(s)^T\phi(s'))^2\right] - 2\mathbb{E}_{\substack{k\sim\text{geom}(1-\gamma)\\ s_t\sim p, s_{t+k}\sim p^\pi(s_{t+k}|s_t)}}\left[\psi(s_t)^T\phi(s_{t+k})\right] \tag{14}$$

Next, we can consider an n-step variant of the TD-SR loss (Blier et al., 2021) which we refer to as TD-SR(n):

$$\min_{\phi,\psi} \mathbb{E}_{\substack{s_t\sim p\\ s'\sim p}}\left[(\psi(s_t)^T\phi(s') \;-\; \gamma^n\bar{\psi}(s_{t+n})^T\bar{\phi}(s'))^2\right] - 2\sum_{i=1}^{n}\mathbb{E}_{s_t\sim p, s_{t+i}\sim p^\pi}\left[\gamma^i\psi(s_t)^T\phi(s_{t+i})\right] \tag{15}$$

We can make the full connection to CL with infinite horizon n:

$$\begin{aligned}
\underset{n\to\infty}{\text{TD-SR}(n)} &= \mathbb{E}_{\substack{s_t\sim p\\ s'\sim p}}\left[(\psi(s_t)^T\phi(s'))^2\right] - 2\sum_{i=1}^{n}\mathbb{E}_{\substack{s_t\sim p\\ s_{t+i}\sim p^\pi(s_{t+i}s_0)}}\left[\gamma^i\psi(s_t)^T\phi(s_{t+i})\right] && (16)\\
&= \mathbb{E}_{\substack{s_t\sim p\\ s'\sim p_0}}\left[(\psi(s_t)^T\phi(s'))^2\right] - \frac{2\gamma}{(1-\gamma)}\sum_{i=1}^{n}\mathbb{E}_{\substack{s_t\sim p\\ s_{t+i}\sim p^\pi(s_{t+i}s_0)}}\left[(1-\gamma)\gamma^{i-1}\psi(s_t)^T\phi(s_{t+i})\right] && (17)\\
&= \mathbb{E}_{\substack{s_t\sim p\\ s'\sim p}}\left[(\psi(s_t)^T\phi(s'))^2\right] - \frac{2\gamma}{(1-\gamma)}\mathbb{E}_{\substack{k\sim\text{geom}(1-\gamma)\\ s_t\sim p, s_{t+k}\sim p^\pi(s_{t+k}|s_t)}}\left[\psi(s_t)^T\phi(s_{t+k})\right] && (18)
\end{aligned}$$

Thus, we can see that in the infinite horizon form of TD-SR(n), it is related to the form of $\text{CL}_{\text{InfoNCE}}$ in (1), but with the positive contrastive term weighted by factor $\frac{\gamma}{1-\gamma}$.

E FINITE MDP

E.1 BYOL

BYOL as an Ordinary Differential Equation (ODE) In finite MDPs, we can characterize the BYOL objective which gives intuition about what information is captured in ϕ, ψ, and conditions that may be useful for stability (Tang et al., 2023; Khetarpal et al., 2025). Consider a finite MDP with transition P^π, linear d-dimensional encoder $\Phi \in \mathbb{R}^{|S|\times d}$, and linear action-free latent-dynamics $\Psi \in \mathbb{R}^{d\times d}$. In a finite MDP, Equation (3) becomes:

$$\min_{\Phi,\Psi} \text{BYOL}(\Phi,\Psi) := \min_{\Phi,\Psi} \mathbb{E}_{s_t\sim\, p_0(s), s_{t+1}\sim P^\pi,}\left[\|\psi^T\Phi^Ts_t - \bar{\Phi}^Ts_{t+1}\|_2^2\right] \tag{19}$$

A property to prevent this objective from collapsing is that Ψ is updated more quickly than Φ. In practice, this is commonly realized as the dynamics are generally a smaller network than the encoder. This system can be analyzed in an ideal setup, where we first find the optimal Ψ, each time before taking a gradient step for Φ, which leads to the ODE for representations Φ (Tang et al., 2023):

$$\Psi^* \in \arg\min_{\Psi} \mathrm{BYOL}(\Phi, \Psi), \quad \dot{\Phi} = -\nabla_{\Phi} \mathrm{BYOL}(\Phi, \Psi)|_{\Psi=\Psi^*} \tag{20}$$

We are able to analyze this ODE with the following assumptions (Tang et al., 2023):

Assumption E.1 (Orthogonal initialization)**.** $\Phi^\top \Phi = I$

Assumption E.2 (Uniform state distribution)**.** $p_0(s) = \frac{1}{|\mathcal{S}|}$

Assumption E.3 (Symmetric dynamics)**.** $P^\pi = (P^\pi)^\top$

Under these three assumptions, Khetarpal et al. (2025) prove that the BYOL ODE is equivalent to monotonically minimizing the surrogate objective:

$$\min_{\Psi} \|P^\pi - \Phi\Psi\Phi^T\|_F + C \tag{21}$$

Where $\|\cdot\|_F$ is the Frobenius matrix norm. Thus, we can understand that the BYOL objective as learning a d-rank decomposition of the underlying dynamics P^π. Additionally, the top d eigenvectors of P^π match those of $(I - \gamma P^\pi)^{-1} = M^\pi$ (Chandak et al., 2023). However, we will highlight that there are key differences when *learning* a low-rank decomposition between P^π and M^π. This is described by Touati et al. (2023), where we can consider that in a real-world problem with underlying continuous-time dynamics, actions may have little effect, and P^π is close to the identity, i.e. close to full-rank. However, M^π, which takes powers of $(P^\pi)^t$, has a "sharpening effect" on the difference between eigenvalues, which gives a clearer learning signal. This is intuitive on a real-world problem like robotics, even with discrete-time dynamics, where $s_{t+1} \approx s_t$, but we have larger differences between s_t and s_{t+k}.

E.2 BYOL-γ

In the finite MDP, we now verify theorem 4.1 , where BYOL-γ approximates the successor representation with matrix decomposition $\tilde{M}^\pi \approx \Phi\Psi\Phi^T$.

We consider the same objective (19), where we need to update the expectation of the sampling distribution:

$$\min_{\Phi,\Psi} \text{BYOL-}\gamma(\Phi, \Psi) := \min_{\Phi,\Psi} \mathbb{E}_{s_t\sim\ p_0(s), s_+ \sim \tilde{M}^\pi,} \left[\|\psi^T \Phi^T s_t - \bar{\Phi}^T s_+\|_2^2\right] \tag{22}$$

Assuming that this objective is optimized under the ODE (20). We have that our objective monotonically minimizes:

$$\min_{\Psi} \|\tilde{M}^\pi - \Phi\Psi\Phi^T\|_F + C \tag{23}$$

This directly translates as we can consider $\tilde{M}^\pi = P^\pi$ as simply a valid transition matrix for a new, temporally abstract, version of the original MDP. We maintain the original assumptions E.1, E.2, and E.3. We do not need an additional assumption for $\tilde{M}^\pi$, as assumption E.3 for symmetric P^π implies a symmetric $\tilde{M}^\pi = (1-\gamma)\sum_{t\geq 0} \gamma^t P_\pi^t$,

Under this setup, we also have that $\Psi\Phi \in \mathbb{R}^{n\times d}$ relates to the successor feature matrix, where each row $(\Psi\Phi)_i$ contains the vector $(1-\gamma)\psi^\pi(s_i)$:

$$(1-\gamma)\psi^\pi(s_i) = \sum_j \tilde{M}^\pi(s_i, s_j)\phi(s_j) \tag{24}$$

$$= (\tilde{M}^\pi\Phi)_i \tag{25}$$

$$\approx (\Phi\Psi\Phi^T\Phi)_i \tag{26}$$

$$= (\Phi\Psi)_i \tag{27}$$

In other words, in the restricted finite MDP, where we minimize (23), we are simultaneously learning successor features $\psi^\pi \approx \Psi\Phi$ and basis features Φ.

F Mixture Datasets

In Section 3.2, we describe a practical setting where we have an offline dataset generated by a set of policies $\{\beta_j\}$. While we previously describe that BYOL-γ approximates $\tilde{M}^{\pi}$ when we have MC samples directly an arbitrary π, we now describe the behavior of BYOL-γ when trained jointly on MC samples from multiple $\{\beta_j\}$, first in the finite MDP, and how this relates to approximating the SR of the unknown mixture policy $\tilde{M}^{\beta}$.

SR of Mixture Policy. We begin by obtaining the SR for the mixture policy $\beta(a|s) := \sum_j \beta_j(a|s)p(\beta_j|s)$, first defining 1-step transitions:

$$P^{\beta_j}_{i,l} = p^{\beta_j}(s_{t+1} = l | s_t = i) = \sum_a \beta_j(a|s = i)p(s_{t+1} = l \mid s_t = i, a) \tag{28}$$

$$P^{\beta}_{i,l} = \sum_a \sum_j \beta_j(a|s)p(\beta_j|s)p(s_{t+1} = l|s_t = i, a) = \sum_j p(\beta_j|s = i)P^{\beta_j}_{i,l} \tag{29}$$

Using $w_j(i) = p(\beta_j|s = i)$, $W_j = \text{diag}(w_j(1), \cdots, w_j(|S|))$, we can see the transitions of the mixture policy β as simply a (state-dependent) weighted average of the transitions of $\{\beta_j\}$.

$$P^{\beta} = \sum_j W_j P^{\beta_j} \tag{30}$$

$$\tilde{M}^{\beta} = (1-\gamma)\sum_{t\geq 0}\gamma^{t+1}(\sum_j W_j P^{\beta_j})^{t+1} \tag{31}$$

Approximated SR of BYOL-γ. Using samples from a set of unknown policies $\{\beta_j\}$ the BYOL-γ objective corresponds to:

$$\min_{\Phi,\Psi} \mathbb{E}_{s_t\sim\ p(s),\beta_j\sim p(\beta_j|s_t),s_+\sim\tilde{M}^{\beta_j}} \left[\|\psi^T\Phi^T s_t - \bar{\Phi}^T s_+\|_2^2\right] \tag{32}$$

$$= \min_{\Phi,\Psi} \mathbb{E}_{s_t\sim\ p(s),s_+\sim\widehat{M}} \left[\|\psi^T\Phi^T s_t - \bar{\Phi}^T s_+\|_2^2\right] \tag{33}$$

i.e. by theorem 4.1 we are approximating $\widehat{M} = \sum_j p(\beta_j|s_t)\tilde{M}^{\beta_j}$, which we can compare to Equation (31) via:

$$\widehat{M} = (1-\gamma)\sum_{t\geq 0}\gamma^{t+1}\sum_j W_j(P^{\beta_j})^{t+1} \tag{34}$$

Intuitively, while $\tilde{M}^{\beta}$ corresponds to the *SR of the average policy*, BYOL-γ approximates $\widehat{M}$, *an average of policy SRs*.

General Case. We rewrite the inequality from Equation (9) but with mixture data:

$$\begin{aligned}\mathcal{L}_{\text{BYOL-}\gamma}(\phi, \psi) &= \mathbb{E}_{\beta_j\sim p(\beta_j),s_t\sim p^{\beta_j}(s),s_+\sim\tilde{M}^{\beta_j}(s_t,s_+)}\left[f(\psi(\phi(s_t)), \bar{\phi}(s_+))\right] \\ &\geq \mathbb{E}_{\beta_j\sim p(\beta_j),s_t\sim p^{\beta_j}(s)}, \left[f(\psi(\phi(s_t)), \mathbb{E}_{s_+\sim\tilde{M}^{\beta_j}(s_t,s_+)}\bar{\phi}(s_+))\right] \\ &= \mathbb{E}_{\beta_j\sim p(\beta_j),s_t\sim p^{\beta_j}(s)}\left[f(\psi(\phi(s_t)), (1-\gamma)\psi^{\beta_j}_{\bar{\phi}}(s_t)\right]\end{aligned} \tag{35}$$

F.1 CL on Mixture Data

We discuss the behavior of CL on mixture datasets. First, we write Equation (36), we rewrite Equation (1) when practically applied to mixture data as implemented in TRA:

$$\mathcal{L}_{\text{TRA}} \approx \mathbb{E}_{\substack{\beta_j\sim p(\beta_j),s_t\sim p^{\beta}_j(s)\\ s_+\sim\tilde{M}^{\beta_j}(s_t,s_+)}}\left[f(\psi(s_t), \phi(s_+))\right] - \mathbb{E}_{s^{1:N}\sim p^{\beta}_j(s)}\left[\log\sum_{i=2}^{N} e^{f(\psi(s^1),\phi(s^i))}\right] \tag{36}$$

We note that a mismatch occurs between the numerator (attractive), and the denominator (repulsive) terms. Namely, we attract two representations only when they are sampled from the same policy β_j, but minimize similarly for states sampled under the occupancy of the mixture policy β.

We could consider other forms where both terms sample from the same distributions. Namely, in the ideal case if we could get MC samples from $s_+ \sim M^\beta(s, s_+)$, the loss has the form:

$$\mathcal{L}_{\mathrm{CL}^\beta} \approx \mathbb{E}_{\substack{s_t \sim p^\beta(s) \\ s_+ \sim \tilde{M}^\beta(s_t, s_+)}} \left[f(\psi(s_t), \phi(s_+))\right] - \mathbb{E}_{s^{1:N} \sim p^\beta(s)} \left[\log \sum_{i=2}^{N} e^{f(\psi(s^1), \phi(s^i))}\right] \tag{37}$$

In practice, we only can take MC samples from, β_j's, so we could view the loss as an expectation over these policies:

$$\begin{aligned}\mathcal{L}_{\mathrm{CL}^{\beta_j}} &\approx \mathbb{E}_{\substack{\beta_j \sim p(\beta_j), s_t \sim p^{\beta_j}(s) \\ s_+ \sim \tilde{M}^{\beta_j}(s_t, s_+)}} \left[f(\psi(s_t), \phi(s_+))\right] - \mathbb{E}_{\beta_j \sim p(\beta_j), s^{1:N} \sim p^{\beta_j}(s)} \left[\log \sum_{i=2}^{N} e^{f(\psi(s^1), \phi(s^i))}\right] \\ &= \mathbb{E}_{\beta_j} \left[\mathbb{E}_{\substack{s_t \sim p^{\beta_j}(s) \\ s_+ \sim \tilde{M}^{\beta_j}(s_t, s_+)}} \left[f(\psi(s_t), \phi(s_+))\right] - \mathbb{E}_{s^{1:N} \sim p^{\beta_j}(s)} [\log \sum_{i=2}^{N} e^{f(\psi(s^1), \phi(s^i))}]\right]\end{aligned} \tag{38}$$

This objective similarly corresponds to capturing information related to a mixture of different policies, similarly to the BYOL-γ objective. We can see the positive term of $\mathcal{L}_{\mathrm{TRA}}$ matches $\mathcal{L}_{\mathrm{CL}^{\beta_j}}$ while the negative term matches $\mathcal{L}_{\mathrm{CL}^\beta}$. In other words, $\mathcal{L}_{\mathrm{TRA}}$ is an under-optimistic compared to $\mathcal{L}_{\mathrm{CL}^\beta}$ and over-pessimistic compared to $\mathcal{L}_{\mathrm{CL}^{\beta_j}}$. We can see how this may discourage stitching. For example, if we have a trajectory $a \to b$, $b \to c$. Although we want relation from a, c, $\psi(a)\phi(c)$ is only sampled as a negative term.

SVD approximation of TRA. In the single policy case, Touati et al. (2023) demonstrates that CL with a single policy ($\mathcal{L}_{\mathrm{CL}}\beta$) relates to an SVD of $\frac{\tilde{M}^\beta(s,s')}{p^\beta(s')}$:

$$\mathcal{L}_{\mathrm{CL}}\beta \approx \mathbb{E}_{s \sim p^\beta, s' \sim p^\beta} \left[\left(\frac{\tilde{M}^\beta(s, s')}{p^\beta(s')} - \psi(s)^T \phi(s')\right)^2\right] + C$$

We now show that the mixture policy case corresponds to an SVD of $\frac{\sum_j p(\beta_j|s)\tilde{M}^{\beta_j}(s,s_+)}{p^\beta(s_+)}$:

$$\begin{aligned}&\mathcal{L}_{\mathrm{TRA}} \approx \mathbb{E}_{\substack{\beta_j \sim p(\beta_j), s_t \sim p^{\beta_j} \\ s' \sim \tilde{M}^{\beta_j}(s_t, s')}} \left[f(\psi(s_t), \phi(s'))\right] - \mathbb{E}_{s \sim p^\beta} \left[\log \mathbb{E}_{s' \sim p^\beta}[e^{f(\psi(s), \phi(s'))}]\right] \\ &= \mathbb{E}_{s \sim p^\beta, s' \sim p^\beta} \left[\frac{\sum_j p(\beta_j|s)\tilde{M}^{\beta_j}(s, s')}{p^\beta(s')} f(\psi(s), \phi(s))\right] - \mathbb{E}_{s \sim p^\beta} \left[\log \mathbb{E}_{s' \sim p^\beta}[e^{f(\psi(s), \phi(s'))}]\right]\end{aligned} \tag{39}$$

Under assumptions that f is the dot product between ϕ and ψ, and ϕ, ψ are centered, if we apply a second-order Taylor expansion to second term:

$$= \mathbb{E}_{s \sim p^\beta, s' \sim p^\beta} \left[\frac{\sum_j p(\beta_j|s)\tilde{M}^{\beta_j}(s, s')}{p^\beta(s')} \psi(s)^T \phi(s')\right] - \frac{1}{2} \mathbb{E}_{s \sim p^\beta, s' \sim p^\beta} \left[\psi(s)^T \phi(s'))\right] \tag{40}$$

$$= \mathbb{E}_{s \sim p^\beta, s' \sim p^\beta} \left[\left(\frac{\sum_j p(\beta_j|s)\tilde{M}^{\beta_j}(s, s')}{p^\beta(s')} - \psi(s)^T \phi(s')\right)^2\right] \tag{41}$$

G Additional Results for Horizon Generalization

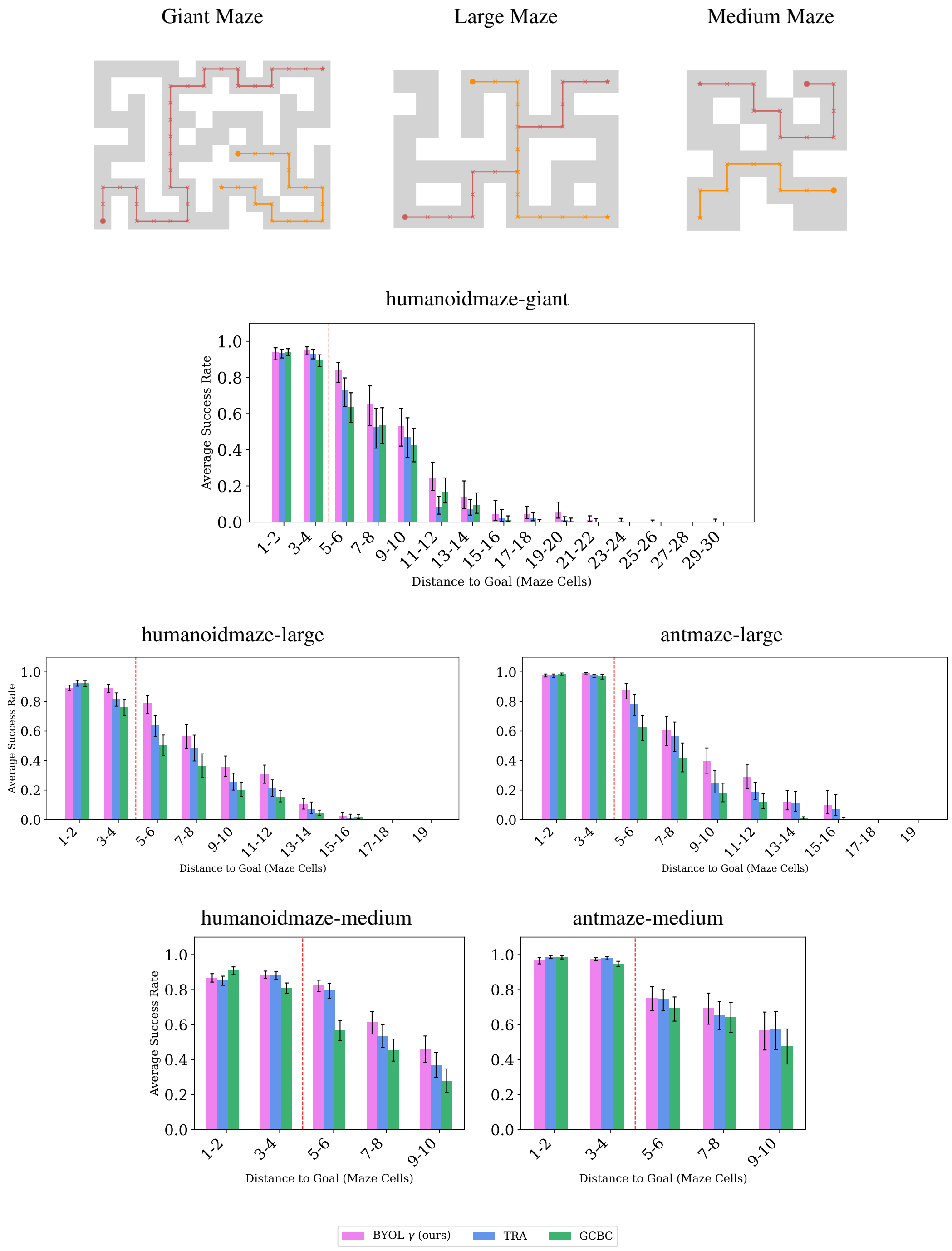


Figure 6: **Evaluating Generalization with Increasing Horizons:** The distances to the right of the red dotted line require combinatorial generalization. The maze maps show examples of how intermediate goals are selected along the optimal path.

We include additional results matching the setup in Section 5.3, for `antmaze-medium`, and `{humanoidmaze}-{medium,large,giant}` in Figure 6. We can observe that BYOL-γ leads in performance as the distance between the start and goal grows when compared to other methods.

H Representations

H.1 Additional Visualizations

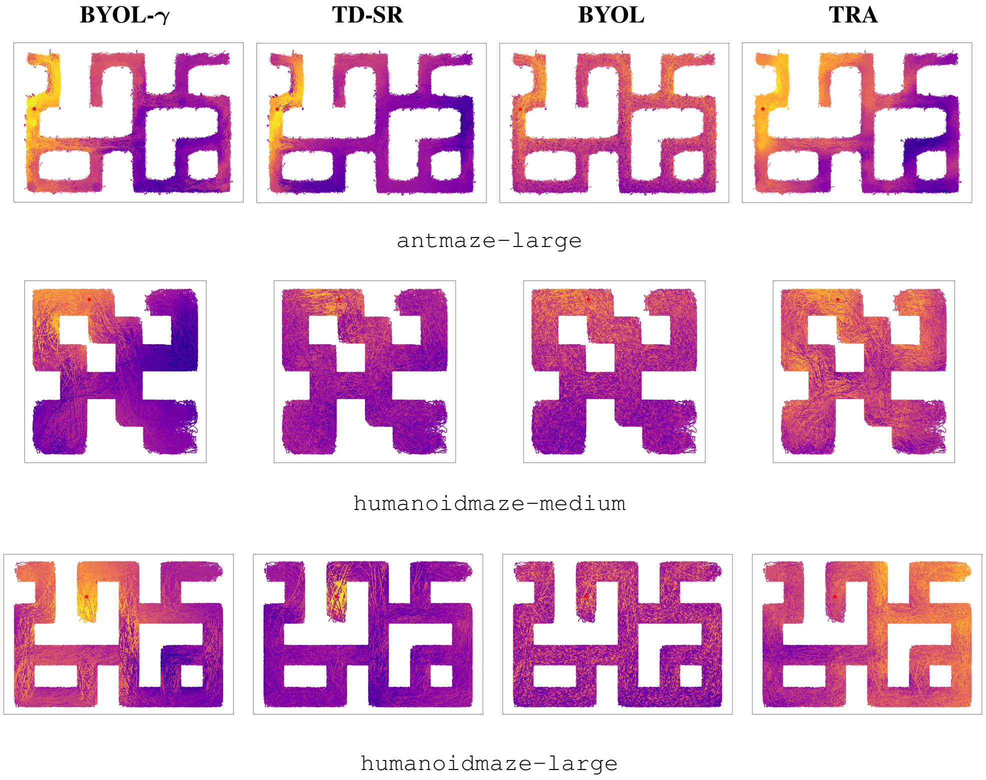


Figure 7: **Additional Visualization of the Learned Representation**: depicts the similarity between the prediction of the current state representation to the goal representation. Brighter color indicates higher similarity.

H.2 Correlation to Shortest Path

We conduct a quantitative comparison between representations through alignment with shortest-path distance in the environment. Namely, we compute the correlation between similarity in the representation space, $\frac{\psi(s,\cdot)^T\phi(g)}{\|\psi(s,\cdot)\|\|\phi(g)\|}$, to the shortest path distance in the maze between sampled start and goal cells (xy space). While the shortest path distance does not measure ground truth temporal distance, as it does not account for robot dynamics, it still provides a simple reference for the general structure we expect to see in representations. We can see that in Table 12, on average BYOL-γ's representations seem to most strongly correlate with shortest path distance. We also compute the success rate of the same checkpoints used for correlation, and notice relationships between these correlations and empirical success rate. We see that the ranking of methods in terms of average correlation in representation space matches the ordering of methods in terms of average empirical policy success.

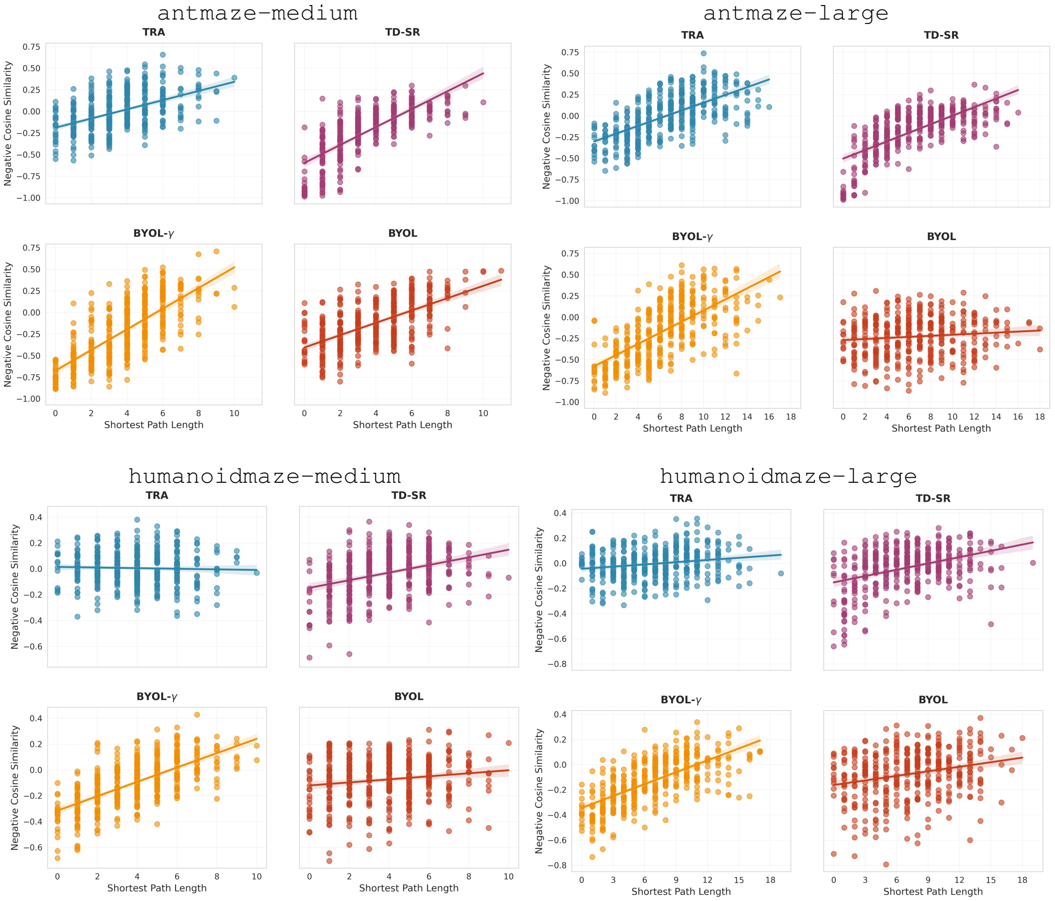


Figure 8: **Scatter plot of negative cosine similarity between randomly sampled (state,goal)l pairs in representation spaces and true shortest path, aggregated over 4 model seeds, each sampled at** 100 **(state,goal) pairs.**

Dataset	**BYOL-γ^a**	**BYOL**	**TRA**	**TDSR^a**
`antmaze-medium-stitch`	0.71 ± 0.01	0.59 ± 0.05	0.49 ± 0.05	0.72 ± 0.03
`antmaze-large-stitch`	0.66 ± 0.02	0.10 ± 0.02	0.62 ± 0.03	0.67 ± 0.02
`humanoidmaze-medium-stitch`	0.64 ± 0.02	0.18 ± 0.04	0.02 ± 0.02	0.36 ± 0.03
`humanoidmaze-large-stitch`	0.62 ± 0.03	0.20 ± 0.03	0.15 ± 0.04	0.38 ± 0.03
`average maze correlation`	**0.66**	0.27	0.32	0.53
`average maze success`	**39**	26	31	36

Table 12: **Correlation of representation space with shortest path distance**. For each method, we use $10,000$ (state,goal) pairs to compute correlation, and then compute the average and standard deviation of the correlation over 4 model seeds, and the success rate over these same checkpoints.