ANGULAR REACHABILITY REPRESENTATIONS FOR OFFLINE GOAL-CONDITIONED REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline goal-conditioned reinforcement learning (GCRL) provides a promising framework for learning generalizable goal-reaching policies from large-scale offline trajectory datasets. However, existing approaches often suffer from inadequate modeling of state-goal relationships, leading to suboptimal policy learning and limited deployment accuracy. In this paper, we propose a novel contrastive representation learning framework for offline GCRL based on unit hyperspherical state-goal embeddings. Our approach learns structured representations of states and goals on two unit hyperspheres, where the reachability between a state and a goal is characterized by their angular relationship. Specifically, this reachability measure can be interpreted as a goal-conditioned value function defined by the future-state occupancy measure, and is equivalently represented by the cosine similarity between the corresponding unit embeddings. We refer to these embeddings as **Angular Reachability Representations (ARR)**. Building upon the learned representations, we propose a novel offline reinforcement learning algorithm that performs policy optimization based on ARR, improving data utilization and mitigating distribution shift. Moreover, we introduce a test-time planning strategy that leverages ARR to refine action selection during deployment without additional training. Many experimental results on OGBench show the effectiveness of the proposed representation, policy-learning method, and test-time planner across offline goal-conditioned control tasks.

1 INTRODUCTION

Large-scale self-supervised pretraining has become a dominant paradigm in natural language processing and computer vision (Devlin et al., 2019; Brown et al., 2020; Chen et al., 2020; He et al., 2020; Grill et al., 2020). This success suggests a promising direction toward building general-purpose intelligent agents by learning reusable knowledge from large-scale experience. However, extending this paradigm to reinforcement learning remains challenging because useful representations for decision making must capture not only semantic information but also temporal dynamics, controllability, and future consequences. Offline goal-conditioned reinforcement learning (GCRL) offers a promising framework for learning generalizable goal-reaching policies from lager-scale offline trajectories. It seeks to learn generalizable behaviors from unlabeled trajectories, but requires representations that capture dynamics-dependent state–goal reachability rather than raw observation similarity.

Contrastive methods address this challenge by learning from future states: C-Learning (Eysenbach et al., 2021) predicts future-state distributions, while Contrastive Learning as Goal-Conditioned Reinforcement Learning (Eysenbach et al., 2022) connects contrastive state–goal compatibility to goal-conditioned values. Although these methods exploit long-horizon trajectory information, they represent reachability largely as an implicit similarity score and provide limited geometric structure for composition and planning. Temporal-distance and quasimetric approaches model asymmetric goal-reaching relations more explicitly. QRL (Wang et al., 2023) introduces an asymmetric quasimetric representation, while subsequent works connect contrastive successor representations to temporal distance (Myers et al., 2024) and combine Monte Carlo learning with quasimetric modeling for offline GCRL (Myers et al., 2025; Zheng et al., 2026). However, these methods primarily focus on

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

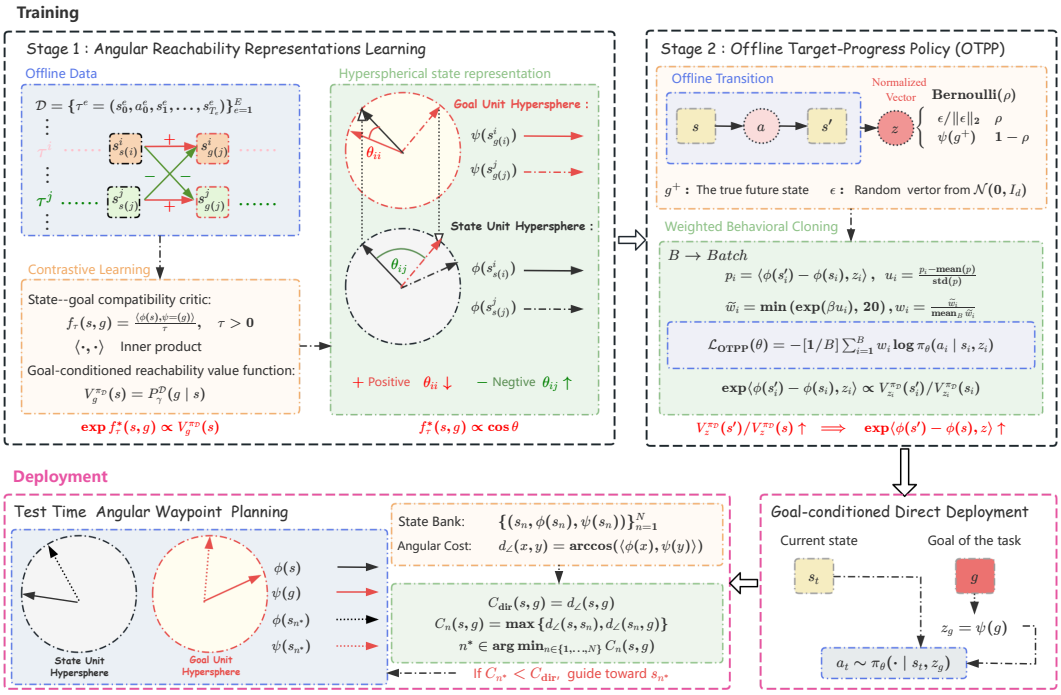

Figure 1: **Overview of Angular Reachability Representations (ARR)**. The black dashed box at the top illustrates the two-stage training procedure, while the pink dashed box below depicts the model's test-time deployment process.

distance-based representations, leaving alternative geometric structures for state–goal compatibility and planning less explored.

These developments motivate a fundamental question: can we learn an explicit representation geometry that preserves the self-supervised learning advantages of contrastive methods while providing an explicit structure for planning and deployment? In this work, we introduce Angular Reachability Representation (ARR), a new representation framework for Offline GCRL. ARR embeds states and goals onto two unit hyperspheres and models goal reachability through their angular relationship. ARR gives contrastive state–goal embeddings an explicit geometric interpretation. Rather than estimating shortest-path distances as quasimetric methods do, it learns a reusable reachability representation for offline policy learning and test-time planning.

Based on ARR, we develop a unified training and deployment framework, as shown in Figure 1, with three components that constitute our contributions. **First**, we introduce ARR, which models goal reachability through hyperspherical state–goal geometry, and learn it from offline trajectories using contrastive learning. **Second**, we develop an ARR-based offline GCRL algorithm that directly exploits the learned reachability structure for policy optimization. **Third**, we propose an ARR-based test-time planner that leverages the learned angular geometry to refine deployment decisions without additional training or environment interaction.

2 Related Work

Goal-conditioned reinforcement learning aims to learn policies capable of achieving arbitrary goals specified by desired future states. Early approaches such as hindsight experience replay (Andrychowicz et al., 2017) improve sample efficiency by relabeling achieved states as goals. However, these methods require online interaction, limiting their applicability in expensive or unsafe environments. Offline GCRL addresses this limitation by learning solely from previously collected trajectories. However, the absence of task-specific rewards and the limited coverage of arbitrary goals make it difficult to identify which actions are useful for a queried target. Existing approaches address this challenge through hindsight goal relabeling, goal-conditioned value learning, advantage-weighted

policy extraction (Andrychowicz et al., 2017; Kostrikov et al., 2022). These studies highlight that effective Offline GCRL requires representations that capture reachability beyond raw observation similarity.

Contrastive learning provides a natural approach for extracting goal-reaching information from unlabeled trajectories. CRL (Eysenbach et al., 2022) shows that contrastive state-goal compatibility can be interpreted as a goal-conditioned value function, establishing a direct connection between representation learning and control. Further work improves the stability and practical performance of contrastive RL in offline robotic tasks (Zheng et al., 2024). However, contrastive representations primarily encode reachability through implicit compatibility. Although effective at capturing future-state relationships, they do not explicitly define geometric structures for trajectory composition and planning. Our work addresses this limitation by introducing an explicit hyperspherical geometry for contrastive reachability representations.

Temporal-distance and quasimetric methods explicitly model the directional relationship between states and goals. QRL (Wang et al., 2023) introduces quasimetric representations for goal-reaching value functions and exploits geometric constraints for compositional reasoning. Later work connects contrastive successor representations with temporal distances (Myers et al., 2024) and develops quasimetric approaches for offline GCRL with improved stitching and long-horizon planning capabilities (Myers et al., 2025; Zheng et al., 2026). Unlike these approaches, which primarily learn explicit distance functions, ARR uses contrastive learning to learn angular reachability geometry on two unit hyperspheres, thereby linking the goal-conditioned value function induced by the offline dataset's future-state occupancy measure to the learned representation. These representations provide a unified interface for self-supervised learning, offline policy optimization, and test-time planning.

3 PRELIMINARIES

3.1 DISCOUNTED FUTURE-STATE OCCUPANCY

Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \gamma)$ be a Markov decision process with discount $\gamma \in [0, 1)$. The offline dataset $\mathcal{D} = \{\tau^e\}_{e=1}^{E}$ consists of trajectories collected before training, where $\tau^e = (s_0^e, a_0^e, s_1^e, \ldots, a_{T_e-1}^e, s_{T_e}^e)$. We write $d_{\mathcal{D}}(s)$ for the distribution used to sample anchor states. When the dataset is collected under multiple policies, $\pi_{\mathcal{D}}(a \mid s)$ denotes the state-conditional mixture policy induced by their aggregate action distribution (Levine et al., 2020).

For a goal g, define the one-step goal-density reward

$$r_g(s, a) \coloneqq (1 - \gamma) P(S_{t+1} = g \mid S_t = s, A_t = a). \tag{1}$$

The behavior-policy action and state values are

$$Q_g^{\pi_{\mathcal{D}}}(s, a) \coloneqq \mathbb{E}_{\pi_{\mathcal{D}}, P}\left[\sum_{k=0}^{\infty} \gamma^k r_g(S_{t+k}, A_{t+k}) \;\middle|\; S_t = s, A_t = a\right], \tag{2}$$

$$V_g^{\pi_{\mathcal{D}}}(s) \coloneqq \mathbb{E}_{A_t \sim \pi_{\mathcal{D}}(\cdot|s)}[Q_g^{\pi_{\mathcal{D}}}(s, A_t)]. \tag{3}$$

We also define the discounted future-state occupancy after the current state:

$$P_\gamma^{\pi_{\mathcal{D}}}(g \mid s) \coloneqq (1 - \gamma) \sum_{k=0}^{\infty} \gamma^k P_{\pi_{\mathcal{D}}}(S_{t+k+1} = g \mid S_t = s). \tag{4}$$

It can be sampled by drawing $K \sim \mathrm{Geom}(1 - \gamma)$ on $\{0, 1, \ldots\}$ and returning S_{t+K+1}. Importantly, the occupancy starts at the next state and therefore aligns its first term with the reward in Eq. (1) (Borsa et al., 2019; Pardo et al., 2018). The motivation for this strict-future convention and its relation to a current-inclusive alternative are given in Appendix A.

3.2 INFONCE

InfoNCE (Information Noise-Contrastive Estimation) is a contrastive objective that learns a compatibility score by distinguishing positive pairs from independently sampled negative pairs (Gutmann

& Hyvarinen, 2010; van den Oord et al., 2018; Chen et al., 2020). We use the standard row-wise softmax form of InfoNCE. Let $q(g)$ be a user-chosen goal-noise density satisfying $q(g) > 0$ whenever $P_\gamma^{\pi_\mathcal{D}}(g \mid s) > 0$ for any s in the support of $d_\mathcal{D}$. Positive state–goal pairs share a trajectory, whereas negative goals are sampled independently of the anchor. Their joint densities are $p_+(s, g) = d_\mathcal{D}(s)P_\gamma^{\pi_\mathcal{D}}(g \mid s)$ and $p_-(s, g) = d_\mathcal{D}(s)q(g)$.

Given an anchor s, we draw one positive goal $g^+ \sim P_\gamma^{\pi_\mathcal{D}}(\cdot \mid s)$ and $K \geq 1$ independent negative goals $g_1^-, \ldots, g_K^- \overset{\text{i.i.d.}}{\sim} q$. The resulting InfoNCE objective is

$$\mathcal{J}_{\text{reach}}^{(K)}(f) = \mathbb{E}_{\substack{s\sim d_\mathcal{D},\ g^+\sim P_\gamma^{\pi_\mathcal{D}}(\cdot|s) \\ g_1^-,\ldots,g_K^- \overset{\text{i.i.d.}}{\sim} q}} \left[\log \frac{\exp f(s, g^+)}{\exp f(s, g^+) + \sum_{j=1}^K \exp f(s, g_j^-)} \right]. \tag{5}$$

Here, $f : \mathcal{S} \times \mathcal{G} \to \mathbb{R}$ denotes a state–goal compatibility critic[1] that assigns a scalar score to each pair, with larger values indicating greater relative compatibility. In practice, the noise density need not be modeled separately. In the in-batch construction, positive pairs are sampled independently, and the future state from another pair serves as a negative goal. The corresponding noise density is therefore the positive goal marginal $q_{\text{batch}}(g) \coloneqq \int d_\mathcal{D}(s)P_\gamma^{\pi_\mathcal{D}}(g \mid s)$. Consequently, under i.i.d. or with-replacement sampling of positive pairs, each off-diagonal anchor–goal pair has population density $d_\mathcal{D}(s)q_{\text{batch}}(g)$ (He et al., 2020; Tian et al., 2020).

4 METHOD

Our method consists of three components, as shown in Figure 1. The first two constitute the training procedure: Stage 1 learns the Angular Reachability Representation (ARR), and Stage 2 trains the policy using the learned ARR. The third component is test-time deployment, which supports both direct goal-conditioned execution and ARR-based waypoint planning.

4.1 STAGE 1: ANGULAR REACHABILITY REPRESENTATIONS LEARNING

We first establish a connection between the goal-conditioned state-value function derived from offline data in Eq. (3) and the discounted future-state occupancy in Eq. (4) as follows.

Proposition 1 (Discounted reachability is a state value)**.** *For the reward in Eq. (1), the behavior-policy state value equals the discounted future-state occupancy:*

$$V_g^{\pi_\mathcal{D}}(s) = P_\gamma^{\pi_\mathcal{D}}(g \mid s). \tag{6}$$

The proof is provided in Appendix A.1. Eq. (6) provides the key link between goal-conditioned value learning and the contrastive estimation of future-state occupancy developed below.

Then, we construct a state–goal compatibility critic. Let $\bar{\phi}$ and $\bar{\psi}$ be arbitrarily deep neural networks whose nonzero final outputs are normalized as $\phi(s) = \frac{\bar{\phi}(s)}{\|\bar{\phi}(s)\|_2}$, $\psi(g) = \frac{\bar{\psi}(g)}{\|\bar{\psi}(g)\|_2}$. Thus $\phi(s), \psi(g) \in \mathbb{S}^{d-1} \coloneqq \{z \in \mathbb{R}^d : \|z\|_2 = 1\}$. The state–goal compatibility critic is

$$f_\tau(s, g) \coloneqq \frac{\langle \phi(s), \psi(g) \rangle}{\tau}, \qquad \tau > 0. \tag{7}$$

The inner product $\langle \cdot, \cdot \rangle$ is a cosine similarity. A larger value corresponds to a smaller cross-encoder angle, while τ controls the attainable logit range. By combining InfoNCE with the state–goal compatibility critic, we can establish a connection among state reachability (or the state-value function from Propositon 1) and the cosine similarity between hyperspherical vectors as follows.

Theorem 2 (Bayes-optimal InfoNCE critic)**.** *Let $p_s(g) \coloneqq P_\gamma^{\pi_\mathcal{D}}(g \mid s)$ and $r_s(g) \coloneqq p_s(g)/q(g)$. Assume $q(g) > 0$ wherever $p_s(g) > 0$ and let f range over all measurable finite-valued critics. For the objective in Eq. (5), the population maximizer is*

$$f^*(s, g) = \log r_s(g) = \log \frac{V_g^{\pi_\mathcal{D}}(s)}{q(g)}. \tag{8}$$

[1] In contrastive learning, the critic function indicates the similarity between a pair of inputs.

Algorithm 1 Stage 1: Angular Reachability Representations Learning

Require: Offline trajectories $\mathcal{D}$; empirical goal vocabulary $\mathcal{G}_{\mathcal{D}}$; encoders ϕ, ψ; discount γ; temperature τ; batch size $B \geq 2$; number of training iterations N

1: **for** $n = 1, \ldots, N$ **do**
2: Sample B valid anchor indices (e_i, t_i) independently with replacement and set $s_i \leftarrow s_{t_i}^{e_i}$
3: **for** $i = 1, \ldots, B$ **do**
4: Set $g_i^+ \leftarrow \mathrm{FutureSample}_\gamma(e_i, t_i)$
5: Compute $f_\tau(s_i, g_j^+)$ for all $i, j \in \{1, \ldots, B\}$
6: Update $(\phi, \psi$ by maximizing the in-batch InfoNCE objective in Eq. (5)

The proof is deferred to Appendix B.1. Together, Theorem 2 and Proposition 1 establish a connection between contrastive learning of the critic f and goal reachability under the behavior policy represented by the offline data (Eysenbach et al., 2022; Myers et al., 2025). With separate normalized state and goal encoders, this connection admits a geometric interpretation through the angle between their hyperspherical embeddings. The distinct roles of the two encoders allow the representation to capture asymmetric reachability relations. A smaller angle between the two unit vector (equivalently, a larger cosine similarity) indicates greater reachability from the starting state to the goal state. Accordingly, we refer to the learned representations ϕ and ψ as Angular Reachability Representations (ARR). Algorithm 1 presents the training procedure for Stage 1. In Algorithm 1, $\mathrm{FutureSample}_\gamma(e, t)$ samples a future goal from trajectory e after time t using geometric temporal weights; truncation and absorbing states are handled as described in Appendix B.3.

4.2 Stage 2: Offline Target-Progress Policy Training

After Stage 1, we freeze the normalized encoders ϕ and ψ and train a policy $\pi_\theta(a \mid s, z)$ conditioned on a target embedding $z \in \mathbb{S}^{d-1}$. Our offline target-progress policy (OTPP) training uses weighted behavior cloning, where the weight of each dataset action is determined by the progress of its observed transition toward the assigned target embedding (Peng et al., 2019; Kostrikov et al., 2022).

For each minibatch of offline transitions$\{(s_i, a_i, s_i')\}_{i=1}^B$, we construct target embedding by mixing encoded dataset goals with random unit vectors. Let g_i be a goal observation drawn using an offline goal sampler, and independently draw $\zeta_i \sim \mathcal{N}(0, I_d)$. We set

$$z_i = \begin{cases} \psi(g_i), & \text{with probability } \rho, \\ \zeta_i / \|\zeta_i\|_2, & \text{with probability } 1 - \rho, \end{cases} \qquad \rho \in [0, 1]. \tag{9}$$

Here $\psi(g_i)$ is already unit-normalized, and the second branch samples uniformly from $\mathbb{S}^{d-1}$. The mixture exposes the policy to both encoded goals and arbitrary latent target. The parameter ρ controls their sampling proportions. These arbitrary latent targets broaden policy conditioning beyond observed goal embeddings and introduces stochastic variation that may improve robustness to unseen goals (Nair et al., 2020a; Pong et al., 2020; Tarbouriech et al., 2022; Pinto et al., 2017). For each transition, we compute the target-progress score $p_i \coloneqq \mathrm{sg}[\langle \phi(s_i') - \phi(s_i), z_i \rangle]$, where sg denotes the stop-gradient operator. Since $\phi(s_i)$, $\phi(s_i')$, and z_i are unit vectors, $p_i \in [-2, 2]$. A positive score indicates that the observed next-state embedding has greater cosine similarity to the target embedding and therefore a smaller angle to it.

For an encoded goal command $z_i = \psi(g_i)$, the target-progress score satisfies $p_i = \tau\big(f_\tau(s_i', g_i) - f_\tau(s_i, g_i)\big)$, which follows directly from Eq. (7). Thus, OTPP uses changes in the Stage 1 discounted reachability (or state value) to weight dataset actions. When the critic exactly recovers the log density ratio with a state-independent additive offset, this identity further gives

$$\frac{p_i}{\tau} = \log V_{g_i}^{\pi_{\mathcal{D}}}(s_i') - \log V_{g_i}^{\pi_{\mathcal{D}}}(s_i), \tag{10}$$

for positive values. Although computed from a single transition, the target-progress is evaluated in a representation learned from discounted future-state occupancy and can therefore reflect longer-horizon reachability under the behavior policy (Mazoure et al., 2023).

In practice, we standardize the progress scores within each minibatch:

$$\mu_p \coloneqq \frac{1}{B}\sum_{j=1}^{B} p_j, \qquad \sigma_p \coloneqq \left[\frac{1}{B}\sum_{j=1}^{B}(p_j-\mu_p)^2\right]^{1/2}, u_i \coloneqq \frac{p_i-\mu_p}{\sigma_p+\epsilon}, \tag{11}$$

where $\epsilon > 0$ ensures a nonzero denominator. We then exponentiate and clip the standardized scores, followed by normalization using the batch mean:

$$\bar{w}_i \coloneqq \min\{\exp(\beta u_i), w_{\max}\}, \quad w_i \coloneqq \mathrm{sg}\left[\frac{\bar{w}_i}{\frac{1}{B}\sum_{j=1}^{B}\bar{w}_j+\epsilon}\right], \tag{12}$$

where $\beta \geq 0$ controls the sensitivity to standardized progress and $w_{\max} > 0$ caps the unnormalized weights. The final weights have approximately unit batch mean; the cap applies before this normalization. The policy is trained by minimizing

$$\mathcal{L}_{\mathrm{OTPP}}(\theta) \coloneqq -\frac{1}{B}\sum_{i=1}^{B} w_i \log \pi_\theta(a_i \mid s_i, z_i). \tag{13}$$

Actions with greater progress relative to the batch mean receive larger imitation weights. All weights remain positive, so actions with lower relative progress retain an imitation contribution with reduced weight. Algorithm 2 summarizes the training procedure.

Algorithm 2 Stage 2: Offline Target-Progress Policy Training

Require: Offline trajectories $\mathcal{D}$; frozen normalized encoders ϕ, ψ; policy π_θ; offline goal sampler; batch size $B \geq 2$; number of training iterations N; mixture probability ρ; weight parameters $\beta, w_{\max}, \epsilon$

for $n = 1, \ldots, N$ **do**
  Sample a minibatch $\{(s_i, a_i, s_i')\}_{i=1}^{B}$ from $\mathcal{D}$
  for $i = 1, \ldots, B$ **do**
    Sample $b_i \sim \mathrm{Bernoulli}(\rho)$
    if $b_i = 1$ **then**
      Draw g_i using the offline goal sampler and set $z_i \leftarrow \psi(g_i)$
    else
      Draw $\zeta_i \sim \mathcal{N}(0, I_d)$ and set $z_i \leftarrow \zeta_i / \|\zeta_i\|_2$
    Compute $p_i \leftarrow \mathrm{sg}[\langle \phi(s_i') - \phi(s_i), z_i \rangle]$
  Standardize $\{p_i\}_{i=1}^{B}$ using Eq. (11)
  Compute the clipped and normalized weights $\{w_i\}_{i=1}^{B}$ using Eq. (12)
  Update θ by minimizing $\mathcal{L}_{\mathrm{OTPP}}$ in Eq. (13)
return $\pi_\theta(a \mid s, z)$

4.3 Goal-Conditioned Deployment and Angular Waypoint Planning

Goal-Conditioned Deployment. For a downstream goal observation g, encode it once and condition the policy on its target embedding:

$$z_g = \psi(g), \qquad a_t \sim \pi_\theta(\cdot \mid s_t, z_g). \tag{14}$$

No task-specific value function or policy is trained.

Angular Waypoint Planning. To provide intermediate guidance at test time, we use a state bank containing N observations and their precomputed ARR, $\{(s_n, \phi(s_n), \psi(s_n))\}_{n=1}^{N}$. Define the cross-encoder angular cost

$$d_\angle(x, y) \coloneqq \arccos\big(\mathrm{clip}\big(\langle \phi(x), \psi(y) \rangle, -1, 1\big)\big). \tag{15}$$

A smaller angular cost corresponds to a larger learned critic score $f_\tau(x, y)$. Since the start and goal encoders differ, $d_\angle$ can be asymmetric and is not assumed to satisfy the triangle inequality.

Algorithm 3 Goal-conditioned deployment with angular waypoint planning

Require: Initial state s; final goal g; state bank $\{(s_n, \phi(s_n), \psi(s_n))\}_{n=1}^{N}$; frozen encoders ϕ, ψ; policy π_θ; replanning interval $H \geq 1$; interaction budget T

Set $t \leftarrow 0$ and cache $z_g \leftarrow \psi(g)$
while the final goal condition is unmet and $t < T$ **do**
  Set $w \leftarrow g$ and $z_w \leftarrow z_g$
  Compute $C_{\text{dir}} \leftarrow d_\angle(s, g)$
  Form $\mathcal{I}(s)$ by excluding near or recently failed waypoints
  if $\mathcal{I}(s) \neq \varnothing$ **then**
    Compute $C_n \leftarrow \max\{d_\angle(s, s_n), d_\angle(s_n, g)\}$ for all $n \in \mathcal{I}(s)$
    Select $n^* \in \arg\min_{n \in \mathcal{I}(s)} C_n$
    if $C_{n^*} < C_{\text{dir}}$ **then**
      Set $w \leftarrow s_{n^*}$ and $z_w \leftarrow \psi(s_{n^*})$
  for $h = 1, \ldots, H$ **do**
    if w is reached, the final goal is reached, or $t \geq T$ **then**
      break
    Sample or select $a \sim \pi_\theta(\cdot \mid s, z_w)$
    Execute a and observe the next state s'
    Set $s \leftarrow s'$ and $t \leftarrow t + 1$

At the current state s and final goal g, let $\mathcal{I}(s) \subseteq \{1, \ldots, N\}$ denote the admissible waypoint indices, excluding candidates near the current state or recently identified as failed waypoints. We score each two-leg route by the larger of its two angular costs:

$$\begin{aligned} C_{\text{dir}}(s, g) &\coloneqq d_\angle(s, g), \\ C_n(s, g) &\coloneqq \max\{d_\angle(s, s_n),\, d_\angle(s_n, g)\}, \\ n^* &\in \arg\min_{n \in \mathcal{I}(s)} C_n(s, g), \quad \mathcal{I}(s) \neq \varnothing. \end{aligned} \tag{16}$$

If $C_{n^*}(s, g) < C_{\text{dir}}(s, g)$, we use s_{n^*} as the immediate guidance goal; otherwise, or if $\mathcal{I}(s)$ is empty,we guide directly toward g. Thus, an accepted waypoint has both angular costs smaller than the current direct angular cost. The policy executes for at most H steps before replanning from the reached state. Algorithm 3 summarizes this procedure. In Appendix C, we establish a theoretical decrease guarantee for the angular cost computed by Algorithm 3. And in our experiments, angular test-time planning substantially improves the policy's goal-reaching success rate.

5 Experiments

5.1 Hyperparameter Sensitivity Studies

We first study the hyperparameter sensitivity of our method to representation and policy-training choices on the state-based benchmark `pointmaze-medium-navigate-v0` from OGBench (Park et al., 2025). In Figure 2,these experiments isolate one factor at a time. Unless otherwise stated, the reference configuration uses row-wise InfoNCE with temperature $\tau = 0.2$, a Stage-1 declared encoder depth of 64, latent dimension 64, batch size 2048, Stage-2 target-mixing probability $\rho = 0.5$, and progress weight $\beta = 1.0$; both stages use the recorded 100K-update budget. Each entry in this record is a single-run estimate with the remaining settings fixed. We evaluate both direct goal-conditioned deployment and the optional angular waypoint planner; these are two deployment modes. The plotted values are aggregate task success rates computed over 50 episodes per task, for a total of 250 episodes across the five tasks.

Temperature τ. Panel (a) illustrates the trade-off in choosing τ. A smaller temperature increases the representable range of critic f and can reduce approximation error, but sharpens the softmax and may produce small gradients. A larger temperature improves smoothness and numerical stability, but weakens discrimination and may overly constrain the logit range in Theorem 2. The proof of how τ affects the expressive capacity of f is provided in Proposition 7 in Appendix B.4.

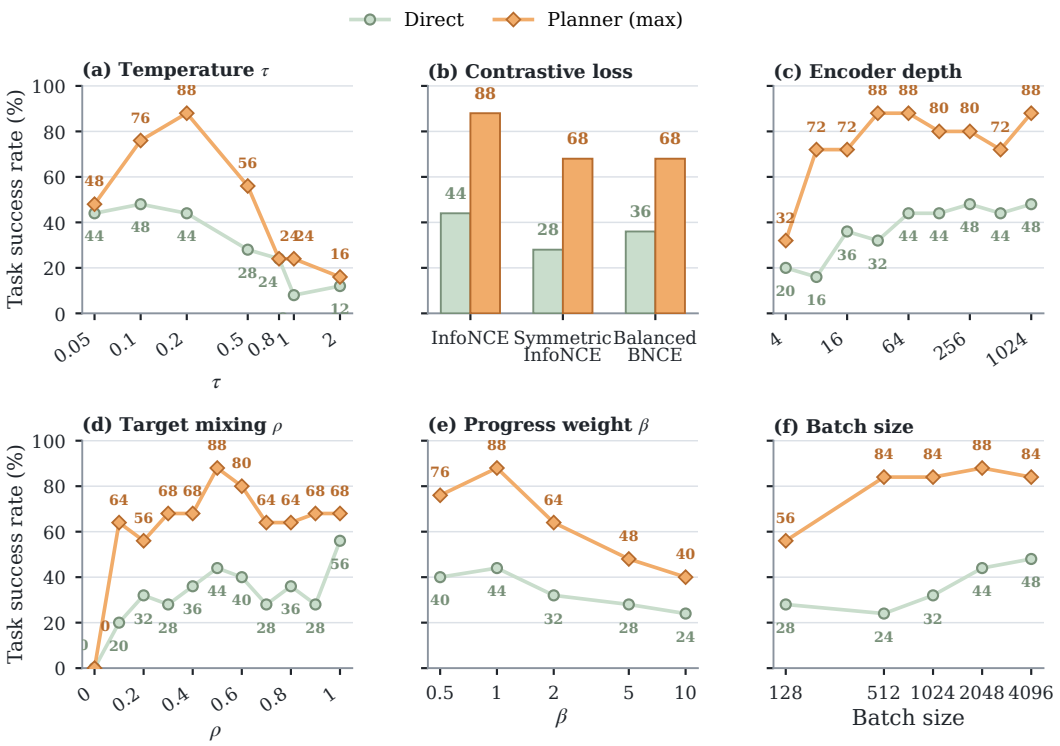

Figure 2: **Hyperparameter sensitivity studies.** Each panel varies one factor while the remaining settings follow the reference configuration. Both direct deployment and the angular waypoint planner are evaluated, denoted respectively by **Direct** and **Planner (max)**.

Contrastive loss. Panel (b) shows that standard InfoNCE achieves better performance than symmetric InfoNCE and balanced BNCE in our experiments. A detailed discussion of symmetric InfoNCE and balanced BNCE is provided in Appendix B.2.

Encoder depth. Panel (c) shows that increasing the Stage 1 encoder depth improves ARR learning, consistent with the scaling behavior reported by Scaling CRL (Wang et al., 2026). However, the gains exhibit diminishing returns: a depth of 1024 performs best, while 64 layers already approaches saturation.

Target mixing ρ. Panel (d) shows that purely random targets lead to ineffective learning, whereas using only trajectory-derived future states can limit waypoint coverage at test time. A moderate ρ improves performance by exposing the policy to a broader range of target conditions.

Progress weight β. Panel (e) shows that β controls the sharpness of the progress-based weighting. Excessively large values amplify small progress differences and concentrate imitation on only the highest-progress transitions.

Batch size. Panel (f) shows that larger training batches generally improve performance. Larger batches provide more informative in-batch negatives for contrastive learning and yield more stable progress statistics and behavior-cloning weights in Stage 2. The improvement eventually becomes less pronounced as the batch size increases, while memory and computational costs continue to grow.

5.2 Comparison on OGBench Tasks

We next compare our method with offline goal-conditioned baselines on four OGBench tasks (Park et al., 2025) in Table 1. The navigation environments are `pointmaze-medium-navigate-v0`, `antmaze-medium-navigate-v0` and `antmaze-large-navigate-v0`; the manipulation environment is `scene-play-v0`. For each method, we train eight independent random seeds. Each seed is evaluated for 50 episodes per task, and we report the mean and standard deviation of

Table 1: **OGBench results.** Values are task success rates in percent (mean ± standard deviation), rounded to the nearest percentage point.

	Baselines									Our Method	
Task / split	GCBC	GCIVL	GCIQL	QRL	HIQL	CGCIVL	MQE	CRL	Scaling CRL	Direct	Planner
Navigation: `pointmaze-medium-navigate-v0`											
Task 1	47 ± 5	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	67 ± 4	73 ± 4	46 ± 8	100 ± 0
Task 2	0 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	100 ± 0	53 ± 9	93 ± 3	52 ± 3	93 ± 9
Task 3	0 ± 0	0 ± 0	0 ± 0	55 ± 5	53 ± 9	70 ± 8	60 ± 7	33 ± 5	42 ± 1	40 ± 8	100 ± 0
Task 4	0 ± 0	0 ± 0	0 ± 0	100 ± 0	100 ± 0	100 ± 0	93 ± 9	46 ± 7	63 ± 8	32 ± 11	100 ± 0
Task 5	7 ± 9	100 ± 0	100 ± 0	100 ± 0	82 ± 8	100 ± 0	100 ± 0	47 ± 9	56 ± 11	44 ± 9	100 ± 0
All Tasks	11 ± 1	60 ± 0	60 ± 0	91 ± 1	87 ± 2	92 ± 1	91 ± 2	49 ± 4	65 ± 5	43 ± 7	**99** ± 1
Navigation: `antmaze-medium-navigate-v0`											
Task 1	37 ± 5	50 ± 3	67 ± 14	80 ± 2	73 ± 9	100 ± 0	100 ± 0	94 ± 3	97 ± 1	82 ± 4	100 ± 0
Task 2	36 ± 2	13 ± 9	75 ± 4	93 ± 3	100 ± 0	93 ± 9	93 ± 9	99 ± 1	96 ± 3	53 ± 9	100 ± 0
Task 3	13 ± 9	20 ± 16	77 ± 9	87 ± 9	80 ± 16	100 ± 0	53 ± 9	83 ± 11	95 ± 4	65 ± 7	85 ± 4
Task 4	40 ± 9	23 ± 7	57 ± 13	81 ± 12	80 ± 6	87 ± 9	7 ± 9	88 ± 6	86 ± 1	32 ± 6	100 ± 0
Task 5	30 ± 1	88 ± 8	81 ± 9	93 ± 5	100 ± 0	93 ± 9	93 ± 9	97 ± 2	95 ± 2	64 ± 9	100 ± 0
All Tasks	31 ± 3	39 ± 7	71 ± 8	87 ± 5	87 ± 4	95 ± 3	69 ± 6	92 ± 3	94 ± 2	59 ± 4	**97** ± 1
Navigation: `antmaze-large-navigate-v0`											
Task 1	7 ± 9	16 ± 3	39 ± 15	76 ± 10	87 ± 9	80 ± 8	80 ± 16	91 ± 1	87 ± 7	65 ± 6	100 ± 0
Task 2	23 ± 1	29 ± 7	34 ± 14	73 ± 4	13 ± 9	33 ± 1	83 ± 8	77 ± 9	44 ± 2	54 ± 7	80 ± 5
Task 3	48 ± 2	22 ± 2	66 ± 6	96 ± 2	100 ± 0	100 ± 0	80 ± 7	85 ± 15	82 ± 8	55 ± 4	100 ± 0
Task 4	27 ± 9	35 ± 4	15 ± 3	75 ± 11	87 ± 9	93 ± 9	33 ± 9	90 ± 3	79 ± 7	61 ± 8	100 ± 0
Task 5	27 ± 2	16 ± 3	22 ± 6	79 ± 2	93 ± 9	93 ± 6	81 ± 9	78 ± 12	85 ± 7	59 ± 2	87 ± 9
All Tasks	26 ± 3	24 ± 2	35 ± 5	80 ± 5	76 ± 3	80 ± 3	71 ± 7	84 ± 6	75 ± 4	59 ± 3	**93** ± 2
Manipulation: `scene-play-v0`											
`open`	33 ± 4	100 ± 0	85 ± 9	18 ± 6	77 ± 5	79 ± 1	100 ± 0	55 ± 6	27 ± 7	37 ± 4	42 ± 3
`unlock_and_lock`	7 ± 9	20 ± 0	79 ± 10	0 ± 0	70 ± 8	87 ± 1	93 ± 4	18 ± 5	3 ± 2	4 ± 2	10 ± 1
`rearrange_medium`	0 ± 0	73 ± 9	72 ± 6	3 ± 2	67 ± 9	66 ± 2	89 ± 7	17 ± 8	1 ± 1	2 ± 2	7 ± 2
`put_in_drawer`	0 ± 0	7 ± 9	3 ± 3	6 ± 4	57 ± 5	73 ± 1	92 ± 5	9 ± 1	1 ± 1	4 ± 2	5 ± 1
`rearrange_hard`	0 ± 0	0 ± 0	0 ± 0	0 ± 0	7 ± 9	36 ± 5	63 ± 2	0 ± 0	0 ± 0	1 ± 1	2 ± 2
All Tasks	8 ± 2	40 ± 4	48 ± 3	5 ± 2	56 ± 1	68 ± 5	**87** ± 3	20 ± 4	6 ± 3	10 ± 2	13 ± 1

the resulting success rates across the eight seeds. For the manipulation environments, we report each named task separately to expose which skills are learned. Direct and Planner denote the two deployment modes of our method: direct goal-conditioned execution and ARR-based angular waypoint planning. Additional experimental details and descriptions of the comparison methods are provided in Appendix D. Table 1 shows that our Direct method substantially outperforms GCBC (Ghosh et al., 2021), indicating that ARR-based transition weighting improves the effectiveness of offline policy learning. Our Planner mehthod achieves state-of-the-art performance on the navigation tasks, suggesting that the learned ARR reachability geometry can obviously improve success rate on long-horizon tasks. These results further demonstrate that ARR effectively captures state–goal relationships in offline data.

6 Conclusion

We introduced Angular Reachability Representations (ARR), which use contrastive learning on two unit hyperspheres to encode asymmetric state–goal reachability. The learned geometry connects discounted future-state occupancy with behavior-policy values and supports ARR-weighted offline behavior cloning without task-specific rewards. The same representation enables direct deployment and angular waypoint planning. The experimental results on OGBench showed the effectiveness of our method, especially on long-horizon navigation tasks.

Limitations and future work. Although the ARR-weighted Direct variant generally outperforms GCBC, its performance remains modest on several tasks. Future work could convert ARR into goal-conditioned rewards or progress signals and integrate them with established offline RL algorithms, such as IQL, CQL, AWAC, and TD3+BC (Kostrikov et al., 2022; Kumar et al., 2020; Nair et al., 2020b; Fujimoto & Gu, 2021). Such combinations may improve direct deployment in long-horizon navigation, multi-stage manipulation, and trajectory-stitching tasks. Regardless of the policy optimizer, ARR can still support test-time angular waypoint planning following Algorithm 3.

AI USE STATEMENT

In this work, we did not use generative AI tools to generate synthetic datasets, implement or run the proposed algorithms, conduct experiments, clean datasets, perform qualitative analysis, or interpret experimental results. The remaining required-disclosure categories were not applicable to this work.

Additionally, we used generative AI tools for recommended tasks, including drafting and editing portions of the manuscript, improving readability. All AI-assisted work was reviewed by the authors. No AI-generated content was accepted without author review and verification. We take responsibility for the final content of this work, including all text, claims, mathematical arguments, citations, and artifacts produced with the aid of generative AI.

REFERENCES

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Andre Barreto, Will Dabney, Remi Munos, Jonathan J. Hunt, Tom Schaul, Hado van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Peter J. Bickel, Michael Bruckner, and Tobias Scheffer. Discriminative learning for differing data distributions. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 81–88, 2007.

Diana Borsa, Andre Barreto, John Quan, Daniel J. Mankowitz, Remi Munos, Hado van Hasselt, David Silver, and Tom Schaul. Universal successor features approximators. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1812.07626.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranab Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 1597–1607. PMLR, 2020.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2007.00224.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019. doi: 10.18653/v1/N19-1423.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. In *International Conference on Learning Representations*, 2021.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 35603–35620, 2022. URL https://arxiv.org/abs/2206.07568.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. URL https://arxiv.org/abs/2004.07219.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20132–20145, 2021.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/1912.06088.

Jean-Bastien Grill, Florian Strub, Florent Altche, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2006.07733.

Michael Gutmann and Aapo Hyvarinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pp. 297–304. PMLR, 2010.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2010.01028.

Kaiqiang Ke, Qian Lin, Zongkai Liu, Shenghong He, and Chao Yu. Conservative offline goal-conditioned implicit v-learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 29591–29607. PMLR, 2025.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2004.11362.

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.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

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

Bogdan Mazoure, Benjamin Eysenbach, Ofir Nachum, and Jonathan Tompson. Contrastive value learning: Implicit models for simple offline reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning*, volume 229, pp. 1257–1267. PMLR, 2023.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 37076–37096, 2024.

Vivek Myers, Bill Chunyuan Zheng, Benjamin Eysenbach, and Sergey Levine. Offline goal-conditioned reinforcement learning with quasimetric representations. In *Advances in Neural Information Processing Systems*, volume 38, pp. 22292–22317, 2025. doi: 10.52202/085713-0661.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems*, volume 32, 2019a. URL https://arxiv.org/abs/1906.04733.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. In *Advances in Neural Information Processing Systems*, volume 32, 2019b. URL https://arxiv.org/abs/1912.02074.

Ashvin Nair, Shikhar Bahl, Alexander Khazatsky, Vitchyr Pong, Glen Berseth, and Sergey Levine. Contextual imagined goals for self-supervised robotic learning. In *Proceedings of the Conference on Robot Learning*, volume 100, pp. 530–539. PMLR, 2020a.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020b. URL https://arxiv.org/abs/2006.09359.

Fabio Pardo, Arash Tavakoli, Vitaly Levdik, and Petar Kormushev. Time limits in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 4045–4054. PMLR, 2018.

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

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.20092.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019. URL https://arxiv.org/abs/1910.00177.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 2817–2826. PMLR, 2017.

Vitchyr Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 7783–7792. PMLR, 2020.

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 *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8748–8763. PMLR, 2021.

Jean Tarbouriech, Omar Darwiche Domingues, Pierre Menard, Matteo Pirotta, Michal Valko, and Alessandro Lazaric. Adaptive multi-goal exploration. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151, pp. 7349–7383. PMLR, 2022.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, pp. 776–794, 2020. doi: 10.1007/978-3-030-58621-8_45.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. URL https://arxiv.org/abs/1807.03748.

Feng Wang, Xiang Xiang, Jian Cheng, and Alan L. Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM International Conference on Multimedia*, pp. 1041–1049, 2017. doi: 10.1145/3123266.3123359.

Kevin Wang, Ishaan Javali, Michał Bortkiewicz, Tomasz Trzcinski, and Benjamin Eysenbach. 1000 layer networks for self-supervised rl: Scaling depth can enable new goal-reaching capabilities. *Advances in Neural Information Processing Systems*, 38:157643–157670, 2026.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9929–9939. PMLR, 2020.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 36411–36430, 2023.

Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. Gendice: Generalized offline estimation of stationary values. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1811.04744.

Bill Chunyuan Zheng, Vivek Myers, Benjamin Eysenbach, and Sergey Levine. Scaling goal-conditioned reinforcement learning with multistep quasimetric distances. In *International Conference on Learning Representations*, 2026.

Chongyi Zheng, Benjamin Eysenbach, Homer Rich Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing contrastive rl: Techniques for robotic goal reaching from offline data. In *International Conference on Learning Representations*, 2024.

APPENDIX

A THE FUTURE-OCCUPANCY CONVENTION

In this paper, we define the discounted future-state occupancy, which differs from the original CRL definition by excluding the starting state. There are two reasons for this choice. First, reinforcement learning is inherently dynamic and requires continual interaction with the environment. Although the current state may already coincide with the goal, we are more interested in future persistence—that is, whether the next and subsequent states continue to occupy the goal. Second, the original CRL formulation includes the initial state and adds a one-time initial-state bonus, making its claimed equality between the Q-function and the discounted occupancy inexact under the next-state reward definition. Excluding the initial state and its bonus aligns the occupancy with the reward's temporal indexing and yields a mathematically more rigorous formulation (Eysenbach et al., 2022; Dayan, 1993; Pardo et al., 2018).

A.1 PROOF OF THE DISCOUNTED REACHABILITY RESULT

Proposition 1 (Discounted reachability is a state value)**.** *For the reward in Eq. (1), the behavior-policy state value equals the discounted future-state occupancy:*

$$V_g^{\pi_\mathcal{D}}(s) = P_\gamma^{\pi_\mathcal{D}}(g \mid s). \tag{6}$$

Proof. We proceed in two steps. First, fix an action a and condition on $S_t = s, A_t = a$. By the definition of $Q_g^{\pi_\mathcal{D}}$ and the reward in Eq. (1),

$$\begin{aligned} Q_g^{\pi_\mathcal{D}}(s,a) &= \mathbb{E}_{\pi_\mathcal{D},P}\left[\sum_{k=0}^{\infty}\gamma^k r_g(S_{t+k}, A_{t+k}) \;\middle|\; S_t = s, A_t = a\right] \\ &= (1-\gamma)\mathbb{E}_{\pi_\mathcal{D},P}\left[\sum_{k=0}^{\infty}\gamma^k P(S_{t+k+1} = g \mid S_{t+k}, A_{t+k}) \;\middle|\; S_t = s, A_t = a\right]. \end{aligned}$$

Because every summand is nonnegative, Tonelli's theorem (equivalently, monotone convergence) allows us to exchange the expectation and the infinite sum:

$$Q_g^{\pi_\mathcal{D}}(s,a) = (1-\gamma)\sum_{k=0}^{\infty}\gamma^k \mathbb{E}_{\pi_\mathcal{D},P}[P(S_{t+k+1} = g \mid S_{t+k}, A_{t+k}) \mid S_t = s, A_t = a].$$

For each $k \geq 0$, the tower property and the Markov property imply

$$\begin{aligned} &\mathbb{E}_{\pi_\mathcal{D},P}[P(S_{t+k+1} = g \mid S_{t+k}, A_{t+k}) \mid S_t = s, A_t = a] \\ &\quad = \mathbb{E}_{\pi_\mathcal{D},P}[\mathbf{1}\{S_{t+k+1} = g\} \mid S_t = s, A_t = a] \\ &\quad = P_{\pi_\mathcal{D}}(S_{t+k+1} = g \mid S_t = s, A_t = a). \end{aligned}$$

Therefore,

$$Q_g^{\pi_\mathcal{D}}(s,a) = (1-\gamma)\sum_{k=0}^{\infty}\gamma^k P_{\pi_\mathcal{D}}(S_{t+k+1}=g \mid S_t=s, A_t=a).$$

We now average over the action $A_t \sim \pi_\mathcal{D}(\cdot \mid s)$. Using the definition of the state value and the law of total probability gives

$$\begin{aligned}
V_g^{\pi_\mathcal{D}}(s) &= \int_\mathcal{A} Q_g^{\pi_\mathcal{D}}(s,a)\,\pi_\mathcal{D}(a \mid s)\mathrm{d}a \\
&= (1-\gamma)\sum_{k=0}^{\infty}\gamma^k \int_\mathcal{A} P_{\pi_\mathcal{D}}(S_{t+k+1}=g \mid S_t=s, A_t=a)\,\pi_\mathcal{D}(a \mid s)\mathrm{d}a \\
&= (1-\gamma)\sum_{k=0}^{\infty}\gamma^k P_{\pi_\mathcal{D}}(S_{t+k+1}=g \mid S_t=s) \\
&= P_\gamma^{\pi_\mathcal{D}}(g \mid s),
\end{aligned}$$

where the last equality follows from the definition in Eq. (4). □

A.2 Why the occupancy starts after the current state

For a generic policy π, define the action-conditioned strict-future occupancy

$$P_{\gamma,+}^{\pi}(g \mid s,a) \coloneqq (1-\gamma)\sum_{k=0}^{\infty}\gamma^k P_\pi(S_{t+k+1}=g \mid S_t=s, A_t=a). \tag{17}$$

Proposition 3 (Strict-future occupancy equals the action value)**.** *For the analytical reward in Eq. (1),*

$$P_{\gamma,+}^{\pi}(g \mid s,a) = Q_g^{\pi}(s,a). \tag{18}$$

Proof. For every $k \geq 0$, the tower property and the Markov property give

$$P_\pi(S_{t+k+1}=g \mid S_t=s, A_t=a) = \mathbb{E}_{\pi,P}[P(S_{t+k+1}=g \mid S_{t+k}, A_{t+k}) \,|\, S_t=s, A_t=a]\,.$$

Substituting this identity into Eq. (17), using the reward definition, and interchanging expectation and the nonnegative sum yields

$$\begin{aligned}
P_{\gamma,+}^{\pi}(g \mid s,a) &= (1-\gamma)\sum_{k=0}^{\infty}\gamma^k \mathbb{E}_{\pi,P}[P(S_{t+k+1}=g \mid S_{t+k}, A_{t+k}) \,|\, S_t=s, A_t=a] \\
&= \mathbb{E}_{\pi,P}\left[\sum_{k=0}^{\infty}\gamma^k r_g(S_{t+k}, A_{t+k}) \,\middle|\, S_t=s, A_t=a\right] = Q_g^{\pi}(s,a).
\end{aligned}$$

□

After averaging over the action distribution, Proposition 3 gives the state-value identity used by HCFP:

$$P_{\gamma,+}^{\pi}(g \mid s) \coloneqq \int_\mathcal{A} P_{\gamma,+}^{\pi}(g \mid s,a)\,\pi(a \mid s)\mathrm{d}a = V_g^{\pi}(s). \tag{19}$$

When g is absorbing, this discounted future occupancy is also a discounted first-arrival probability. Without absorption, it is the probability or density of occupying g at a geometrically sampled future time; it does not by itself require the process to remain at g at every subsequent step (Dayan, 1993; Barreto et al., 2017).

A.3 Why the original CRL Proposition 1 and Lemma 4.1 require correction

The original CRL Proposition 1 combines two conventions: a discounted occupancy that includes the current state and a next-state reward with a one-time initial-state bonus. As written, its claimed equality between the occupancy and the Q-function is false. Consequently, the original Lemma 4.1, which substitutes that equality into the optimal contrastive critic, also requires correction. We make both corrections explicit below (Eysenbach et al., 2022).

For a fixed initial pair, write the current-inclusive occupancy as

$$\overline{P}^{\pi}_{\gamma,0}(g \mid s,a) \coloneqq (1-\gamma)\sum_{k=0}^{\infty}\gamma^k P_\pi(S_k = g \mid S_0 = s, A_0 = a), \tag{20}$$

The original formulation also adds the following one-time initial-state bonus:

$$r^{\mathrm{CRL}}_{g,0}(s,a) \coloneqq (1-\gamma)\left[P(S_1 = g \mid S_0 = s, A_0 = a) + p_0(g)\right], \tag{21}$$

where $p_0(g) \coloneqq P(S_0 = g)$ under the initial-state distribution. The ordinary next-state reward r_g in Eq. (1) is used after the initial step. Let

$$\widetilde{Q}^{\pi}_{g,0}(s,a) \coloneqq \mathbb{E}_{\pi,P}\left[r^{\mathrm{CRL}}_{g,0}(S_0,A_0) + \sum_{k=1}^{\infty}\gamma^k r_g(S_k,A_k) \;\middle|\; S_0 = s, A_0 = a\right] \tag{22}$$

denote the return under this original definition. Because the bonus is applied only at the initial step, it is not a stationary state–action reward unless the state is augmented with an initial-step flag. More importantly, it does not repair the temporal mismatch with the current-inclusive occupancy.

Proposition 4 (Correction to the original CRL Proposition 1)**.** *For a discrete state space and* $\gamma \in [0,1)$*,*

$$\begin{aligned}
\widetilde{Q}^{\pi}_{g,0}(s,a) &= Q^{\pi}_g(s,a) + (1-\gamma)p_0(g),\\
\overline{P}^{\pi}_{\gamma,0}(g \mid s,a) &= (1-\gamma)\mathbf{1}\{s=g\} + \gamma Q^{\pi}_g(s,a)\\
&= (1-\gamma)\mathbf{1}\{s=g\} + \gamma\widetilde{Q}^{\pi}_{g,0}(s,a) - \gamma(1-\gamma)p_0(g).
\end{aligned} \tag{23}$$

Therefore neither $Q^{\pi}_g(s,a) = \overline{P}^{\pi}_{\gamma,0}(g \mid s,a)$ *nor* $\widetilde{Q}^{\pi}_{g,0}(s,a) = \overline{P}^{\pi}_{\gamma,0}(g \mid s,a)$ *holds in general.*

Proof. The bonus is constant under conditioning on (S_0, A_0), so the first line of Eq. (23) follows immediately. For the ordinary next-state reward,

$$Q^{\pi}_g(s,a) = (1-\gamma)\sum_{j=0}^{\infty}\gamma^j P_\pi(S_{j+1} = g \mid S_0 = s, A_0 = a).$$

The current-inclusive occupancy therefore satisfies

$$\begin{aligned}
\overline{P}^{\pi}_{\gamma,0}(g \mid s,a) &= (1-\gamma)\mathbf{1}\{s=g\} + (1-\gamma)\sum_{k=1}^{\infty}\gamma^k P_\pi(S_k = g \mid S_0 = s, A_0 = a)\\
&= (1-\gamma)\mathbf{1}\{s=g\} + \gamma Q^{\pi}_g(s,a),
\end{aligned}$$

where the second equality uses $\sum_{k=1}^{\infty}\gamma^k u_k = \gamma\sum_{j=0}^{\infty}\gamma^j u_{j+1}$. Substituting the first line of Eq. (23) gives the final line. □

The error in Proposition 1 propagates directly into the original Lemma 4.1. The next result gives the corrected statement while retaining the original current-inclusive positive sampling distribution and binary contrastive objective.

Lemma 5 (Corrected form of the original Lemma 4.1)**.** *Let* $p_{\mathrm{neg}}(g)$ *be the marginal goal density used for negative samples in the original equal-prior binary NCE objective. If positive goals are*

sampled from $\overline{P}^{\pi}_{\gamma,0}(\cdot \mid s,a)$, then the Bayes-optimal binary critic satisfies

$$\begin{aligned}\exp f^*_{\mathrm{orig}}(s,a,g) &= \frac{\overline{P}^{\pi}_{\gamma,0}(g \mid s,a)}{p_{\mathrm{neg}}(g)} \\ &= \frac{(1-\gamma)\mathbf{1}\{s=g\} + \gamma Q^{\pi}_{g}(s,a)}{p_{\mathrm{neg}}(g)} \\ &= \frac{(1-\gamma)\mathbf{1}\{s=g\} + \gamma \widetilde{Q}^{\pi}_{g,0}(s,a) - \gamma(1-\gamma)p_0(g)}{p_{\mathrm{neg}}(g)}.\end{aligned} \tag{24}$$

Thus the original critic is proportional to the current-inclusive occupancy, not directly to either version of the Q-function.

Proof. For the original binary contrastive problem, the positive and negative joint densities, with anchor density $d(s,a)$, are

$$\begin{aligned}p_{\mathrm{CRL},+}(s,a,g) &= d(s,a)\overline{P}^{\pi}_{\gamma,0}(g \mid s,a), \\ p_{\mathrm{CRL},-}(s,a,g) &= d(s,a)p_{\mathrm{neg}}(g).\end{aligned}$$

With equal class priors, the Bayes-optimal logit obeys $\exp f^*_{\mathrm{orig}} = p_{\mathrm{CRL},+}/p_{\mathrm{CRL},-}$. Cancelling $d(s,a)$ gives the first line of Eq. (24); Proposition 4 gives the remaining two lines. □

If the current-state term and initial bonus are discarded, positive goals are instead sampled from the strict-future occupancy in Eq. (17). Proposition 3 then restores the exact binary-NCE relationship intended by the original Proposition 1 and Lemma 4.1:

$$\exp f^*_{\mathrm{strict}}(s,a,g) = \frac{P^{\pi}_{\gamma,+}(g \mid s,a)}{p_{\mathrm{neg},+}(g)} = \frac{Q^{\pi}_{g}(s,a)}{p_{\mathrm{neg},+}(g)}, \tag{25}$$

where $p_{\mathrm{neg},+}$ is the marginal induced by strict-future positive sampling. This correction is about the exact numerical and log-value interpretation (Bickel et al., 2007; Nachum et al., 2019a; Zhang et al., 2020; Nachum et al., 2019b). For fixed (s,g) and $\gamma > 0$, the additional terms in Eq. (24) do not depend on a, so the corrected current-inclusive score still has the same action maximizers as $Q^{\pi}_{g}(s,a)$. For continuous states, the indicator is replaced by the corresponding Dirac density or by membership in a measurable goal region.

B Proofs and Details for Stage 1

B.1 Proof of the Bayes-optimal InfoNCE critic

Theorem 2 (Bayes-optimal InfoNCE critic)**.** *Let $p_s(g) \coloneqq P^{\pi_{\mathcal{D}}}_{\gamma}(g \mid s)$ and $r_s(g) \coloneqq p_s(g)/q(g)$. Assume $q(g) > 0$ wherever $p_s(g) > 0$ and let f range over all measurable finite-valued critics. For the objective in Eq. (5), the population maximizer is*

$$f^*(s,g) = \log r_s(g) = \log \frac{V^{\pi_{\mathcal{D}}}_{g}(s)}{q(g)}. \tag{8}$$

Proof. It is useful to place the positive at a uniformly random index. Let $I \sim \mathrm{Unif}\{0,\dots,K\}$; conditionally on $S = s$ and $I = i$, draw $G_i \sim p_s$ and draw every G_j with $j \neq i$ independently from q. For a candidate tuple $X = (s, g_0, \dots, g_K)$, Bayes' rule gives

$$\Pr(I = i \mid X) = \frac{p_s(g_i)\prod_{j\neq i} q(g_j)}{\sum_{\ell=0}^{K} p_s(g_\ell)\prod_{j\neq \ell} q(g_j)} = \frac{r_s(g_i)}{\sum_{\ell=0}^{K} r_s(g_\ell)}. \tag{26}$$

The model posterior induced by a critic is

$$\widehat{\Pr}_f(I = i \mid X) = \frac{\exp f(s,g_i)}{\sum_{\ell=0}^{K} \exp f(s,g_\ell)}. \tag{27}$$

The InfoNCE objective is the expected categorical log-likelihood, and hence the standard cross-entropy decomposition yields

$$\mathcal{J}^{(K)}_{\text{reach}}(f) = -H(I \mid X) - \mathbb{E}_X \,\mathrm{KL}\Big(\Pr(I \mid X) \,\Big\|\, \widehat{\Pr}_f(I \mid X)\Big). \tag{28}$$

It is maximized exactly when the two posteriors agree almost everywhere. Taking the ratio of two posterior coordinates gives, for any g, g' in the support,

$$f(s,g) - f(s,g') = \log r_s(g) - \log r_s(g'). \tag{29}$$

Therefore $f(s,g) - \log r_s(g)$ is independent of g and equals some $c(s)$. Conversely, substituting Eq. (8) into the softmax cancels $c(s)$ and recovers the Bayes posterior. Proposition 1 gives $p_s(g) = V_g^{\pi_\mathcal{D}}(s)$. More generally, all population maximizers are of the form $f^*(s,g) + c(s)$, where $c(s)$ is an arbitrary measurable finite-valued row-wise shift. For the remainder of the paper, we use the representative selected by the convention $c(s) \equiv 0$; equivalently, its row partition function is one. □

B.2 In-batch Balanced BNCE and Symmetric InfoNCE

Building on row-wise InfoNCE, we consider two in-batch variants: balanced BNCE, which equally weights diagonal positive and off-diagonal negative pairs, and symmetric InfoNCE, which jointly optimizes state-to-goal and goal-to-state softmax objectives (Chen et al., 2020; Khosla et al., 2020; Radford et al., 2021; Tian et al., 2020).

Let $B \geq 2$ and draw B independent positive pairs $(s_i, g_i^+) \overset{\text{i.i.d.}}{\sim} p_+(s,g) = d_\mathcal{D}(s) P_\gamma^{\pi_\mathcal{D}}(g \mid s), \quad i = 1, \ldots, B$. For $\ell_{ij} \coloneqq f(s_i, g_j^+)$, the in-batch balanced BNCE maximization objective is

$$\widehat{\mathcal{J}}^{(B)}_{\text{IB-BNCE}}(f) \coloneqq \frac{1}{2B} \sum_{i=1}^{B} \log \sigma(\ell_{ii}) + \frac{1}{2B(B-1)} \sum_{\substack{1 \leq i,j \leq B \\ i \neq j}} \log \sigma(-\ell_{ij}), \tag{30}$$

where $\sigma(u) \coloneqq (1 + \exp(-u))^{-1}$.

Since the pairs are sampled independently, every off-diagonal pair (s_i, g_j^+) with $i \neq j$ has product marginal $d_\mathcal{D}(s) q_{\text{batch}}(g)$. Therefore, the population objective is

$$\begin{aligned} \mathcal{J}_{\text{IB-BNCE}}(f) &\coloneqq \mathbb{E}\Big[\widehat{\mathcal{J}}^{(B)}_{\text{IB-BNCE}}(f)\Big] \\ &= \frac{1}{2}\,\mathbb{E}_{\substack{s \sim d_\mathcal{D} \\ g \sim P_\gamma^{\pi_\mathcal{D}}(\cdot \mid s)}} \big[\log \sigma\big(f(s,g)\big)\big] + \frac{1}{2}\,\mathbb{E}_{\substack{s \sim d_\mathcal{D} \\ g \sim q_{\text{batch}}}} \big[\log \sigma\big(-f(s,g)\big)\big]. \end{aligned} \tag{31}$$

Theorem 6 (Bayes-optimal in-batch balanced BNCE critic)**.** *Assume that $P_\gamma^{\pi_\mathcal{D}}(\cdot \mid s)$ and q_{batch} are mutually absolutely continuous for $d_\mathcal{D}$-almost every s, with finite density ratio. Then the unique population maximizer of Eq. (31), up to null sets, is*

$$f^*_{\text{IB-BNCE}}(s,g) = \log \frac{P_\gamma^{\pi_\mathcal{D}}(g \mid s)}{q_{\text{batch}}(g)} = \log \frac{V_g^{\pi_\mathcal{D}}(s)}{q_{\text{batch}}(g)}. \tag{32}$$

Unlike row-wise softmax InfoNCE, the sigmoid likelihood fixes the absolute logit and leaves no row-wise additive gauge.

Proof. Draw a batch of independent positive pairs as above. To express the class-balanced objective as a binary classification experiment, first draw $Y \sim \text{Bernoulli}(1/2)$. If $Y = 1$, draw $I \sim \text{Unif}\{1, \ldots, B\}$ and return $X = (S_I, G_I^+)$.If $Y = 0$, draw an ordered pair (I, J) uniformly from $\{(i,j) : 1 \leq i, j \leq B,\ i \neq j\}$ and return $X = (S_I, G_J^+)$.

The expected log-likelihood of this experiment is exactly Eq. (31): the first branch averages the diagonal terms and the second branch averages the off-diagonal terms.

The conditional density of $X = (s,g)$ given $Y = 1$ is $d_\mathcal{D}(s) P_\gamma^{\pi_\mathcal{D}}(g \mid s)$, whereas the conditional density given $Y = 0$ is $d_\mathcal{D}(s) q_{\text{batch}}(g)$. Hence Bayes' rule gives

$$\eta(s,g) \coloneqq \Pr(Y = 1 \mid X = (s,g)) = \frac{P_\gamma^{\pi_\mathcal{D}}(g \mid s)}{P_\gamma^{\pi_\mathcal{D}}(g \mid s) + q_{\text{batch}}(g)}.$$

A critic f induces the posterior $\widehat{\eta}_f(s,g) \coloneqq \sigma\big(f(s,g)\big)$. The conditional Bernoulli cross-entropy decomposition therefore yields

$$\mathcal{J}_{\text{IB-BNCE}}(f) = -H(Y \mid X) \quad - \mathbb{E}_X\big[\mathrm{KL}\big(\mathrm{Bern}\big(\eta(X)\big) \,\big\|\, \mathrm{Bern}\big(\widehat{\eta}_f(X)\big)\big)\big] \le -H(Y \mid X).$$

For $f_0(s,g) \coloneqq \log \frac{P_\gamma^{\pi_\mathcal{D}}(g|s)}{q_{\text{batch}}(g)}$, we have $\sigma\big(f_0(s,g)\big) = \eta(s,g)$, so f_0 attains the upper bound. Conversely, equality requires $\widehat{\eta}_f(s,g) = \eta(s,g)$ almost everywhere. Since the sigmoid is strictly increasing,

$$f(s,g) = \mathrm{logit}\big(\eta(s,g)\big) = \log \frac{P_\gamma^{\pi_\mathcal{D}}(g \mid s)}{q_{\text{batch}}(g)}$$

almost everywhere. Finally, Proposition 1 gives $P_\gamma^{\pi_\mathcal{D}}(g \mid s) = V_g^{\pi_\mathcal{D}}(s)$, proving Eq. (32). □

Using the same in-batch logits, symmetric InfoNCE averages the state-to-goal row-wise and goal-to-state column-wise softmax objectives. Under the same support assumptions, its population maximizers over unrestricted finite-valued critics satisfy $f^*_{\text{sym}}(s,g) = \log[V_g^{\pi_\mathcal{D}}(s)/q_{\text{batch}}(g)] + C$, where C is a global constant. Thus, symmetric InfoNCE reduces the state-dependent additive ambiguity of row-wise InfoNCE to a global shift, whereas in-batch balanced BNCE uniquely determines the absolute logit.

B.3 Sampling and Empirical Loss

We use one boundary-aware future-goal sampler for both types of trajectory endings. First sample B anchors independently from $d_\mathcal{D}$. For an anchor s_t, if its episode reaches a true absorbing state, extend the trajectory with copies of that state and draw $K \sim \mathrm{Geom}(1-\gamma)$ on $\{0,1,\dots\}$. If the recorded endpoint is a time limit or another nonabsorbing boundary, do not treat that endpoint as absorbing; instead draw K from the geometric law conditioned on the valid offsets (Pardo et al., 2018; Fu et al., 2020):

$$P(K = k \mid t, T) = \frac{\gamma^k}{\sum_{j=0}^{T-t-1} \gamma^j}, \qquad k \in \{0, \dots, T-t-1\}, \tag{33}$$

and set $g^+ = s_{t+K+1}$ from the same trajectory. Thus, the absorbing branch exactly realizes the infinite-horizon discounted future occupancy, whereas the nonabsorbing branch defines a horizon- and time-dependent truncated future distribution. The density-ratio derivation in Theorem 2 applies to the resulting sampled positive law. For the truncated branch, an exact horizon-specific value interpretation requires the anchor to include the variables that determine the sampling law (in particular, the time or remaining horizon); with a state-only encoder, the objective instead identifies the mixture induced by the recorded sampling procedure. Only the absorbing branch has the infinite-horizon $V_g^{\pi_\mathcal{D}}(s)$ interpretation of Proposition 1; in the other branch, value and log-value statements refer to the explicitly truncated counterpart. We call this boundary-aware operation $\mathrm{FutureSample}_\gamma(e,t)$.

Our methhod intentionally omits the action from the first encoder: including a_i would change the identified quantity from the state value $V_g^{\pi_\mathcal{D}}(s_i)$ to the action value $Q_g^{\pi_\mathcal{D}}(s_i,a_i)$. For an in-batch InfoNCE implementation, form the complete pairwise logit matrix

$$\ell_{ij} \coloneqq f_\tau(s_i, g_j^+), \qquad i,j \in \{1,\dots,B\}. \tag{34}$$

The diagonal entries are positive pairs and the off-diagonal entries are in-batch negatives. Use the row-wise softmax loss

$$\widehat{\mathcal{L}}_{\text{InfoNCE}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\ell_{ii})}{\sum_{j=1}^{B}\exp(\ell_{ij})}. \tag{35}$$

This is the standard InfoNCE cross-entropy with $B-1$ in-batch negatives per anchor; there is no separate positive/negative normalization. Under independent pair sampling, an off-diagonal goal has the marginal distribution q_{batch}, not a declaration that it is unreachable from the anchor. Algorithm 1 invokes the boundary-aware sampler defined above; the termination type must be retained so that the selected positive-sampling law is explicit rather than an unintended mixture of infinite-horizon and time-dependent truncated laws (Chen et al., 2020; He et al., 2020; Chuang et al., 2020; Kalantidis et al., 2020).

B.4 Temperature and Critic Expressivity

Proposition 7 (Temperature bounds relative InfoNCE scores)**.** *Every* ARR *critic satisfies*

$$|f_\tau(s,g)| \le \frac{1}{\tau}, \qquad |f_\tau(s,g) - f_\tau(s,g')| \le \frac{2}{\tau}. \tag{36}$$

Consequently, its represented density-ratio odds between two goals are bounded by $\exp(2/\tau)$*. Let* Q *denote the goal-noise probability measure induced by the density (or mass function)* q*. In the definition below,* $\operatorname{ess\,sup}_Q$ *and* $\operatorname{ess\,inf}_Q$ *denote the essential supremum and infimum with respect to* Q*: sets of goals with zero* Q*-measure are ignored. This is the appropriate notion for the population InfoNCE objective. For a discrete goal space with positive* q*-mass on each state, these quantities reduce to the ordinary maximum and minimum over the support of* q*.*

$$\Delta(s) \coloneqq \operatorname*{ess\,sup}_Q \log r_s(g) - \operatorname*{ess\,inf}_Q \log r_s(g).$$

A bounded critic with range $[-1/\tau, 1/\tau]$ *can represent the Bayes InfoNCE posterior at state* s *only if* $\Delta(s) \le 2/\tau$*; when this condition fails, optimization is a constrained approximation to the Bayes posterior. Finite-dimensional spherical factorization can impose additional error (Wang & Isola, 2020; Wang et al., 2017; Khosla et al., 2020; Radford et al., 2021).*

Proof. Cauchy–Schwarz and unit normalization imply $|\langle \phi(s), \psi(g)\rangle| \le 1$, which gives the first inequality in Eq. (36); the second follows from the triangle inequality. Because InfoNCE depends only on within-row logit differences, the relevant Bayes logit range at state s has width $\Delta(s)$. A spherical critic has maximum within-row gap $2/\tau$, so it can represent the Bayes posterior only when $\Delta(s) \le 2/\tau$. If the width is larger, the coupled InfoNCE objective must settle for a constrained approximation. □

C Proof of Angular Waypoint Descent

Proposition 8 (Estimated angular descent under max-leg acceptance)**.** *Let* $D_g(s) \coloneqq d_\angle(s,g)$*. Suppose that the planner accepts a waypoint* s_{n^*} *at state* s *only if*

$$\max\{d_\angle(s, s_{n^*}),\, d_\angle(s_{n^*}, g)\} < D_g(s). \tag{37}$$

Then

$$d_\angle(s, s_{n^*}) < D_g(s), \qquad D_g(s_{n^*}) < D_g(s). \tag{38}$$

Consequently, if policy execution reaches the accepted waypoint s_{n^*} *exactly, the estimated direct angular cost to the final goal strictly decreases.*

Proof. Each of the two angular costs is no larger than their maximum. Therefore, Eq. (37) implies

$$\begin{aligned} d_\angle(s, s_{n^*}) &\le \max\{d_\angle(s, s_{n^*}),\, d_\angle(s_{n^*}, g)\} < D_g(s), \\ D_g(s_{n^*}) = d_\angle(s_{n^*}, g) &\le \max\{d_\angle(s, s_{n^*}),\, d_\angle(s_{n^*}, g)\} < D_g(s). \end{aligned}$$

This proves Eq. (38). If policy execution reaches the accepted waypoint exactly, the new state is $s' = s_{n^*}$, and hence

$$D_g(s') = D_g(s_{n^*}) < D_g(s).$$

Thus, the estimated direct angular cost to the final goal strictly decreases. □

D Training and Planner Configurations

Table 1 compares our method with representative offline goal-conditioned RL baselines from OGBench (Park et al., 2025). These methods can be broadly grouped into imitation-based, value-based, geometric, and contrastive approaches.

GCBC represents the simplest imitation-based baseline. It directly clones dataset actions conditioned on future goals (Ghosh et al., 2021), without learning an explicit value function or performing test-time planning. ARR shares the use of behavior cloning, but uses the learned reachability geometry

to assign larger weights to transitions that make greater progress toward the target. Thus, ARR-Direct retains the simplicity of imitation learning while incorporating dynamics-aware state–goal information.

GCIVL, GCIQL, and CGCIVL belong to the value-based family. GCIVL and GCIQL adapt implicit value and Q-learning to goal-conditioned settings using expectile regression and IQL-style policy extraction (Kostrikov et al., 2022; Park et al., 2025). CGCIVL further introduces conservative regularization and quasimetric modeling to reduce value overestimation for cross-trajectory state–goal pairs (Ke et al., 2025). HIQL extends this family with a hierarchical policy: a high-level policy predicts latent subgoals, while a low-level policy executes them (Park et al., 2023). These methods learn value functions explicitly and use them for policy extraction or subgoal selection. ARR instead learns a reward-free reachability representation from future-state contrastive pairs and uses it to weight policy learning, without requiring a task-specific value function. Nevertheless, ARR-derived rewards could be combined with these offline RL algorithms in future work.

QRL and MQE form the geometric or temporal-distance family. QRL models goal-conditioned values as asymmetric quasimetrics (Wang et al., 2023), while MQE combines quasimetric representations with multistep Monte Carlo returns to improve long-horizon goal stitching (Zheng et al., 2026). These methods explicitly model temporal distance and use geometric constraints to support compositional planning. ARR also provides an explicit geometry, but its angular compatibility is learned on two asymmetric unit hyperspheres and represents dataset-induced reachability rather than a shortest-path distance. The ARR-based waypoint planner uses this geometry for lightweight test-time planning, without requiring a hierarchical policy or a quasimetric shortest-path objective.

CRL and Scaling CRL are the closest contrastive baselines. CRL estimates goal-reaching values through contrastive learning between state–action pairs and future states (Eysenbach et al., 2022), while Scaling CRL studies how increasing critic depth affects contrastive goal-reaching performance (Wang et al., 2026). ARR builds on the same principle of learning reachability from unlabeled trajectories, but uses strict-future state–goal pairs, separate normalized state and goal encoders, and an asymmetric hyperspherical compatibility. In addition, ARR directly reuses the learned geometry for target-progress policy weighting and angular waypoint planning, thereby connecting representation learning, policy optimization, and deployment.

The following tables record the configurations used for the four state-based OGBench benchmarks in Section 5.2. "Obs./act." denotes the state and continuous-action dimensions. The notation "$w/d/L$" in the Stage 1 column gives encoderwidth, declared depth, and latent dimension, while "w/d" in the Stage 2 column gives policy width and depth. Here β is `progress_beta` and ρ is `stage2_goal_target_prob`; both are training parameters and are distinct from the test-time planner thresholds. The LR column is the common learning rate for both stages, and the batch column is the minibatch size used by both training objectives. "scaling-residual" abbreviates the `scaling_crl_residual_encoder`, and OTPP denotes the `target_progress_policy` with the `otpp` objective.

Table 2: **Training configurations for the state-based benchmarks.** "0.1M" and "1M" denote 100,000 and 1,000,000 optimizer updates, respectively.

Dataset	Obs./act.	τ	Stage 1 architecture / updates	Stage 2 architecture / updates	β	ρ	Batch	LR	γ
`pointmaze-medium-navigate-v0`	2/2	0.2	`scaling-residual` $w/d/L = 64/64/64$; 0.1M	`OTPP` $w/d = 512/3$; 0.1M	1.0	0.5	2048	3×10^{-4}	0.99
`antmaze-medium-navigate-v0`	29/8	0.2	`scaling-residual` $w/d/L = 64/64/64$; 1M	`OTPP` $w/d = 512/3$; 1M	1.0	0.8	2048	3×10^{-4}	0.99
`antmaze-large-navigate-v0`	29/8	0.2	`scaling-residual` $w/d/L = 64/64/64$; 1M	`OTPP` $w/d = 512/3$; 1M	1.0	0.9	2048	3×10^{-4}	0.99
`scene-play-v0`	40/5	0.2	`scaling-residual` $w/d/L = 64/64/64$; 0.1M	`OTPP` $w/d = 512/3$; 0.1M	1.0	1.0	2048	3×10^{-4}	0.99

OTPP uses a target offset of one transition, precomputation chunks of 4096, and no tanh squashing, state-dependent standard deviation, or value networks. The state bank stored in each checkpoint has 10,000 states. The planner does not retrain either stage; it only changes deployment-time waypoint selection. The offline datasets contain up to 1,000,000 transitions (1,000 episodes), and each checkpoint has a state-bank capacity of 10,000 states.

Table 3: **Planner configurations used for deployment.** All planners use the max route aggregation and an interaction budget of 1000 steps unless shown otherwise. "Repeat" indicates exclusion of a previously selected waypoint; "failed radius" is the locomotion XY exclusion radius after a failed progress check, and is unused for the manipulation metrics.

Dataset	State bank	Route / fallback	Replan	Persistence	Min. improvement	Progress margin	Final goal distance	Failed radius; repeat	Max steps
`pointmaze-medium-navigate-v0`	10k	max /mean	25	50	0.02	0.02	1.0	0.75; yes	1000
`antmaze-large-navigate-v0`	100k	max /none	25	100	0.05	0.02	2.0	0; no	1000
`antmaze-medium-navigate-v0`	100k	max /none	25	100	0.05	0.02	2.0	0; no	1000
`scene-play-v0`	10k	max /mean	25	50	0.10	0.02	2.0	0; yes	1000