# GCHR: GOAL-CONDITIONED HINDSIGHT REGULARIZATION VIA COMPOSITIONAL PRIORS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

In goal-conditioned reinforcement learning (GCRL) with sparse rewards, the agent receives little learning signal. Hindsight experience replay (HER) relabels trajectories with the goals they achieved, but it treats each relabeled transition independently. The actor therefore receives no direct guidance from the structure of whole trajectories. We propose GCHR (**G**oal-**C**onditioned **H**indsight **R**egularization), a backbone-agnostic framework that distills past trajectories into *hindsight compositional priors* and regularizes the actor toward them. A *hindsight behaviour prior* anchors the policy to relabeled actions. A *hindsight goal prior* aggregates the target policy's behaviours toward waypoints on the same trajectory to transfer knowledge across goals. Both priors come from the replay buffer and a target copy of the actor, so GCHR trains no additional network. Theoretically, the goal prior extends action coverage beyond self-imitation, and its training loss upper-bounds the divergence to the exact prior in expectation. Under idealized conditions, a via-goal value that measures cross-goal transfer does not degrade as the policy improves. Experiments on three benchmark suites with four backbones (DDPG, SAC, TD3, CRL) and state or visual observations show gains in sample efficiency and final performance on most tasks. Ablation, goal-corruption, and coverage studies link these gains to trajectory-aligned cross-goal transfer.

## 1 INTRODUCTION

Goal-conditioned reinforcement learning (GCRL) trains agents to reach goals specified at test time (Liu et al., 2022; Zheng et al., 2024a). It is a key step toward general-purpose robotic systems (Park et al., 2025). Under sparse rewards, the agent receives feedback only when it reaches the goal, so credit assignment over long horizons is difficult.

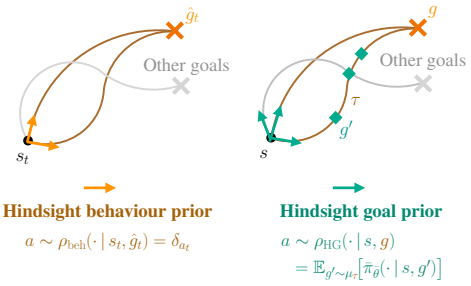



Figure 1: **GCHR overview.** GCHR regularizes the actor toward two hindsight compositional priors: a behaviour prior $\rho_{\text{beh}}$ on a relabeled action (left) and a goal prior $\rho_{\text{HG}}$ that averages the target policy over waypoints $g'$ of the same trajectory $\tau$ (right). Their mixture $\rho_{\text{mix}}$ (Equation (4)) is conceptual.

Hindsight Experience Replay (HER) (Andrychowicz et al., 2017) relabels failed trajectories with the goals they actually achieved, so every trajectory yields learning signal. Each relabeled transition, however, is used in isolation. A trajectory $s_0 \to s_1 \to \cdots \to s_T$ shows how to reach its final achieved goal $\phi(s_T)$ from $s_0$, where $\phi$ maps a state to the goal it achieves. It also shows how the intermediate states reach one another. The critic can propagate this reachability through temporal difference learning, but the *actor* receives no direct trajectory-level guidance.

We propose GCHR (**G**oal-**C**onditioned **H**indsight **R**egularization) to turn this trajectory structure into guidance for the actor. GCHR builds *hindsight compositional priors* from past trajectories and keeps the policy close to them while it maximizes return (Figure 1). The **hindsight behaviour prior** $\rho_{\text{beh}}$ anchors the policy to relabeled actions, each of which was followed by reaching its relabeled goal.



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.

The **hindsight goal prior** $\rho_{\mathrm{HG}}$ transfers knowledge across goals. If the policy knows how to reach the waypoints $g'_1, g'_2, \ldots$ visited along a trajectory, its actions toward them can serve as proposals for the target goal $g$. Both priors come from the replay buffer and a slowly updated target copy of the actor. GCHR therefore trains no additional network, dynamics model, or subgoal planner, and it can be added to any actor-critic GCRL algorithm.

**Contributions.** (i) We propose GCHR, a backbone-agnostic framework that regularizes the actor toward hindsight compositional priors built from past trajectories. (ii) We show that the goal prior extends the coverage of self-imitation and keeps a fixed share of every waypoint behaviour (Proposition 4.1). Under idealized conditions, the target policy's via-goal value does not degrade during training, up to a reachability tolerance (Theorem 4.3). (iii) We impose the two priors through a behaviour cloning term and a KL term toward the goal prior. The KL term upper-bounds the divergence to the exact goal prior in expectation and lets the policy commit to one waypoint behaviour (Proposition 5.1). (iv) On three benchmark suites and four backbones (DDPG (Lillicrap et al., 2015), SAC (Haarnoja et al., 2018), TD3 (Fujimoto et al., 2018), CRL (Eysenbach et al., 2022b)) with state and visual observations, GCHR improves sample efficiency and final performance on most tasks. Ablation, goal-corruption, and coverage studies link these gains to trajectory-aligned cross-goal transfer.

## 2 Related Work

**Hindsight relabeling.** HER (Andrychowicz et al., 2017) addresses sparse rewards in GCRL by treating achieved goals as desired goals when it trains the action value. Several studies improve its goal sampling. CHER (Fang et al., 2019) selects goals heuristically to balance diversity and proximity. MHER (Yang et al., 2021) learns a dynamics model to generate virtual relabeled goals, and model errors limit its accuracy. These methods change which relabeled goals enter training. GCHR keeps HER relabeling and adds guidance on the actor side.

**Self-imitation and goal-conditioned supervised learning.** GCSL (Ghosh et al., 2021), WGCSL (Yang et al., 2022), and related methods learn policies by unweighted or weighted behaviour cloning on hindsight data. They are biased toward past experience and can converge to suboptimal policies (Eysenbach et al., 2022a). SIL (Oh et al., 2018) reproduces past decisions that yielded high returns. GRSIL (Li et al., 2023) trains a separate self-imitated policy and greedily selects between it and the goal-conditioned policy by their estimated returns. The behaviour prior of GCHR is a form of self-imitation. The goal prior instead transfers the behaviour of the current target policy across goals. Under idealized conditions, this transfer does not degrade as the policy improves (Theorem 4.3).

**Policy bootstrapping through subgoals.** Hierarchical methods such as DHRL (Lee et al., 2022) follow waypoints from a high-level policy during execution. RIS (Chane-Sane et al., 2021) instead regularizes a flat policy toward a prior that averages the target policy over imagined subgoals. These subgoals come from a separately trained prediction network. SAW (Zhou & Kao, 2026) bootstraps a flat policy on subgoal-conditioned policies at in-trajectory waypoints with advantage-weighted importance sampling. It is an offline method and trains a separate subpolicy. The goal prior of GCHR is closest to RIS and SAW. It draws waypoints from the stored trajectory, uses only a target copy of the actor, and runs online.

**Contrastive and representation-based GCRL.** CRL (Eysenbach et al., 2022b) frames goal reaching as contrastive learning. TD-InfoNCE (Zheng et al., 2024b) replaces its Monte-Carlo positives with a temporal difference estimate. QRL (Wang et al., 2023) learns a quasimetric value function. Section 6.2 adds GCHR to CRL and two other backbones on the GPU-accelerated JaxGCRL benchmark (Bortkiewicz et al., 2025).

**Regularized actor objectives.** MPO (Abdolmaleki et al., 2018) constrains the policy by a KL term against a prior within an E-step and M-step scheme. Galashov et al. (2019) introduce information-asymmetric priors in KL-regularized RL, and Tirumala et al. (2022) learn reusable behaviour priors. Distral (Teh et al., 2017) distills knowledge shared across tasks into a common prior, which resembles our aggregation across goals. AWR (Peng et al., 2019) and AWAC (Nair et al., 2020) combine weighted behaviour cloning with RL. Offline methods add a behaviour cloning term directly to the actor loss (Fujimoto & Gu, 2021; Tarasov et al., 2023). Without its goal term, the GCHR actor

loss takes this form on hindsight data. Unlike RIS and SAW, none of these priors is built from the waypoints of a trajectory; the goal term of GCHR supplies such a prior.

# 3 PRELIMINARIES

## 3.1 GOAL-CONDITIONED MDP AND HINDSIGHT RELABELING

We consider a goal-conditioned Markov decision process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{G}, P, r, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, goal space $\mathcal{G}$, transition kernel $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward $r : \mathcal{S} \times \mathcal{G} \to \{0, 1\}$, and discount factor $\gamma \in [0, 1)$. Here $\Delta(\mathcal{S})$ is the set of probability distributions over $\mathcal{S}$. Following prior work (Andrychowicz et al., 2017; Plappert et al., 2018; Yang et al., 2021), a deterministic map $\phi : \mathcal{S} \to \mathcal{G}$ extracts the goal-relevant features of a state. The reward is $r(s, g) = \mathbb{1}\{\|\phi(s) - g\| \le \epsilon\}$ for a small threshold $\epsilon > 0$, and $S_g = \{s \in \mathcal{S} : \|\phi(s) - g\| \le \epsilon\}$ is the set of states that satisfy $g$. A goal-conditioned policy $\pi(a|s, g)$ gives the probability density of action $a$ in state $s$ under goal $g$. For a set $B \subseteq \mathcal{A}$, $\pi(B|s, g)$ denotes the probability of $B$, and other action distributions use the same notation. The Q-function and value function are

$$Q^\pi(s, a, g) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, g) \,\middle|\, s_0 = s, a_0 = a\right], \quad V^\pi(s, g) = \mathbb{E}_{a\sim\pi(\cdot|s,g)}[Q^\pi(s, a, g)]. \tag{1}$$

Given a trajectory $\tau = (s_0, a_0, s_1, \ldots, s_T)$ collected for goal $g$, HER (Andrychowicz et al., 2017) relabels transition $t$ with an achieved goal $\hat{g}_t = \phi(s_{t'})$ with $t' > t$. Following the *future* strategy of Andrychowicz et al. (2017), we draw $s_{t'}$ uniformly from $\{s_{t+1}, \ldots, s_T\}$. The relabeled transitions form $\mathcal{D}_{\text{HER}} = \{(s_t, a_t, s_{t+1}, \hat{g}_t)\}$. The replay buffer is $\mathcal{D} = \mathcal{D}_{\text{orig}} \cup \mathcal{D}_{\text{HER}}$, where $\mathcal{D}_{\text{orig}}$ keeps the original goals. Every replay element also stores a reference to its source trajectory.

## 3.2 FROM VALUES TO REACHABILITY

With rewards in $\{0, 1\}$, a value measures the discounted time the agent spends in the goal set (Eysenbach et al., 2020). We define the discounted reaching probability

$$p^\pi(g|s, a) = (1 - \gamma) \sum_{\Delta=0}^{\infty} \gamma^\Delta \Pr_\pi\left(s_\Delta \in S_g \mid s_0 = s, a_0 = a\right). \tag{2}$$

Comparing with Equation (1) gives $Q^\pi(s, a, g) = p^\pi(g|s, a)/(1 - \gamma)$. With $p^\pi(g|s) = \mathbb{E}_{a\sim\pi(\cdot|s,g)}[p^\pi(g|s, a)]$, we also have $V^\pi(s, g) = p^\pi(g|s)/(1 - \gamma)$. Let $T_g$ be the first time the state enters $S_g$, with $\gamma^{T_g} = 0$ if it never does. We call goal sets *absorbing* if, for every goal $g$, the process that follows $\pi(\cdot|\cdot, g)$ stays in $S_g$ after time $T_g$. Then $p^\pi(g|s) = \mathbb{E}_\pi[\gamma^{T_g}]$ is the discounted first-hitting probability. Section 4.2 uses this reading, and Remark 4.4 discusses when it applies.

# 4 GCHR FRAMEWORK

Past trajectories encode knowledge about goal reachability. GCHR distills this knowledge into action priors and regularizes the policy toward them. We call this regularization policy bootstrapping. Section 4.1 defines the priors and their coverage, and Section 4.2 studies when the goal prior is reliable.

## 4.1 HINDSIGHT COMPOSITIONAL PRIOR

KL-regularized RL (Haarnoja et al., 2018; Levine, 2018) keeps the policy close to a prior $\rho(a|s, g)$. It maximizes $\mathbb{E}_\pi[\sum_t \gamma^t(r(s_t, g) - \kappa D_{\text{KL}}(\pi(\cdot|s_t, g)\|\rho(\cdot|s_t, g)))]$ with temperature $\kappa > 0$, and its optimal policy is

$$\pi^*(a|s, g) \propto \rho(a|s, g) \exp\left(Q^{\pi^*}_{\text{soft}}(s, a, g)/\kappa\right), \tag{3}$$

where $Q^{\pi^*}_{\text{soft}}$ is the soft Q-function of $\pi^*$, whose return includes the KL penalties of future steps. The optimal policy reweights the prior by soft action values. A prior that already favours good actions

leaves less for the values to correct, which can speed up learning. We build two such priors from hindsight data and call them the *hindsight compositional priors*. Their conceptual mixture is

$$\rho_{\text{mix}}(a|s,g) = \lambda\rho_{\text{beh}}(a|s,g) + (1-\lambda)\rho_{\text{HG}}(a|s,g), \tag{4}$$

where $\lambda \in (0,1)$ is a conceptual weight. Section 5 imposes the two components through separate losses weighted by $\alpha$ and $\beta$, so $\lambda$ plays no operational role.

**Hindsight behaviour prior.** Actions that led to a goal in the past are good candidates for reaching it again. For a relabeled transition $(s_t, a_t, s_{t+1}, \hat{g}_t) \in \mathcal{D}_{\text{HER}}$, we set $\rho_{\text{beh}}(\cdot|s_t, \hat{g}_t) = \delta_{a_t}$, the Dirac measure at $a_t$. This prior anchors the policy to an action that was followed by reaching $\hat{g}_t$, although this action need not be optimal. Regularizing toward $\rho_{\text{beh}}$ yields behaviour cloning on hindsight data, as in self-imitation methods such as GCSL (Ghosh et al., 2021). Its limitation is coverage. It informs only the state-goal pairs that appear in $\mathcal{D}_{\text{HER}}$ and gives no guidance for other pairs.

**Hindsight goal prior.** The goal prior removes this limitation by transferring knowledge across goals. Consider a replay element with state $s$ and goal $g$. Let $\tau$ be its source trajectory, $\mathcal{G}_H(\tau) = \{\phi(s_0), \dots, \phi(s_T)\}$ the multiset of goals visited along $\tau$, $H = |\mathcal{G}_H(\tau)| = T+1$ its size, and $\mu_\tau$ the uniform distribution over its $H$ elements. The goal prior averages the target policy $\bar{\pi}_{\bar{\theta}}$ over these waypoints,

$$\rho_{\text{HG}}(a|s,g) = \mathbb{E}_{g'\sim\mu_\tau}\left[\bar{\pi}_{\bar{\theta}}(a|s,g')\right] \approx \pi_{\text{HG}}(a|s,g) := \frac{1}{K}\sum_{k=1}^{K} \bar{\pi}_{\bar{\theta}}(a|s,g'_k), \tag{5}$$

where $g'_1, \dots, g'_K$ are $K$ waypoints drawn from $\mu_\tau$. The target policy $\bar{\pi}_{\bar{\theta}}$ is a slowly updated copy of the current policy $\pi_\theta$, with $\bar{\theta} \leftarrow \tau_{\text{soft}}\theta + (1-\tau_{\text{soft}})\bar{\theta}$ and $\tau_{\text{soft}} \in (0,1]$. We call $\pi_{\text{HG}}$ the aggregated hindsight goal policy. Both $\rho_{\text{HG}}$ and $\pi_{\text{HG}}$ depend on $g$ only through $\tau$, which we omit from the notation. Even if the agent has never reached $g$ from $s$, it may know how to reach the waypoints on the same trajectory. Its behaviour toward these waypoints then offers candidate actions for $(s,g)$. We call this mechanism *cross-goal transfer*. The prior $\rho_{\text{HG}}$ averages densities, so each waypoint behaviour remains a separate mode. The behaviour prior copies a recorded action, whereas the goal prior copies the target policy's behaviour toward other goals, so the two differ even in degenerate settings (Appendix C.3). Section 6.4 isolates the contribution of the goal prior.

We now formalize the coverage gain. For a replay element $x = (s_x, a_x, s_x^+, g_x, \tau_x) \in \mathcal{D}$ with next state $s_x^+$, define the behaviour action set $\mathcal{A}_{\text{beh}}(x) = \{a : (s_x, a, \cdot, g_x) \in \mathcal{D}_{\text{HER}}\}$ and the hindsight goal action set $\mathcal{A}_{\text{HG}}(x) = \{a : \exists\, w \in \mathcal{G}_H(\tau_x),\ \bar{\pi}_{\bar{\theta}}(a|s_x, w) > 0\}$.

**Proposition 4.1** (Coverage Expansion). *(i) If $\bar{\pi}_{\bar{\theta}}(a|s,g) > 0$ for every tuple $(s,a,\cdot,g) \in \mathcal{D}_{\text{HER}}$, then $\mathcal{A}_{\text{beh}}(x) \subseteq \mathcal{A}_{\text{HG}}(x)$ for every $x \in \mathcal{D}_{\text{HER}}$. (ii) $\mathcal{A}_{\text{HG}}(x) \neq \emptyset$ for every $x \in \mathcal{D}$, including elements with $\mathcal{A}_{\text{beh}}(x) = \emptyset$. (iii) For every $x \in \mathcal{D}$, every waypoint $w \in \mathcal{G}_H(\tau_x)$, and every measurable set $B \subseteq \mathcal{A}$, we have $\rho_{\text{HG}}(B|s_x, g_x) \geq \bar{\pi}_{\bar{\theta}}(B|s_x, w)/H$ with $H = |\mathcal{G}_H(\tau_x)|$.*

The proof is in Appendix C.5. Parts (i) and (ii) show that the goal prior covers self-imitation and informs every replayed state-goal pair, including pairs without direct experience, which are common under sparse rewards. For full-support policies such as Gaussians, these set statements hold trivially. Part (iii) stays informative in that case, because the goal prior keeps at least a $1/H$ share of the mass that any waypoint behaviour places on any action set. The proposition does not state that this mass lies on high-value actions, and Section 6.6 tests this question directly.

### 4.2 Reliability of the Goal Prior

Coverage alone does not make the goal prior useful. Actions toward a waypoint help only if the agent can continue from it toward the target goal. The value of this continuation should depend mainly on the waypoint, not on where the agent reaches it. We capture this requirement with one structural condition.

**Assumption 4.2** (Uniform Reachability). *A policy $\pi$ satisfies uniform reachability with tolerance $\delta \geq 0$ if $|V^\pi(s_1,g) - V^\pi(s_2,g)| \leq \delta$ for every target goal $g$, every waypoint $g'$, and all $s_1, s_2 \in S_{g'}$.*[1]

[1] Every policy satisfies the condition with $\delta = 1/(1-\gamma)$, and Theorem 4.3 is then vacuous. The tolerance thus measures how far a policy is from the ideal case $\delta = 0$. If goals are full states and $V^\pi(\cdot, g)$ is $L$-Lipschitz, the condition holds with $\delta = 2L\epsilon$ (Appendix C.2). Analyses of directed goal selection assume that the goal space contains optimal paths (Diaz-Bone et al., 2026), which our analysis does not need.

The condition does not claim that reaching $g'$ ensures reaching $g$. It is plausible when goal-relevant features, such as end-effector or object positions, dominate reachability, as in Fetch, HandReach, and navigation. We expect it to be looser for multi-axis in-hand rotation, where finger configurations also matter (Appendix C.2). The condition depends on the policy, and our results require it only for the training iterates.

To measure how well a waypoint transfers, let $T_{g'}$ be the first time the state enters $S_{g'}$ when the agent follows $\pi(\cdot|\cdot, g')$ from $s$. At time $T_{g'}$, the agent switches to $\pi(\cdot|\cdot, g)$. The via-goal value of reaching $g$ through $g'$ is

$$V^{\pi}_{\mathrm{via}}(s, g, g') = \mathbb{E}\big[\gamma^{T_{g'}}\, V^{\pi}(s_{T_{g'}}, g)\big] = p^{\pi}(g'|s)\, \mathbb{E}_{s' \sim d^{\pi}(\cdot|s,g')}\big[V^{\pi}(s', g)\big], \tag{6}$$

where the second equality uses absorbing goal sets, under which $p^{\pi}(g'|s) = \mathbb{E}[\gamma^{T_{g'}}]$ (Section 3). When $p^{\pi}(g'|s) > 0$, the distribution $d^{\pi}(C|s, g') = \mathbb{E}[\gamma^{T_{g'}}\, \mathbb{1}\{s_{T_{g'}} \in C\}]/p^{\pi}(g'|s)$ over measurable sets $C \subseteq \mathcal{S}$ weights the hitting state by the discount. $V^{\pi}_{\mathrm{via}}$ is thus the discounted value of first reaching $g'$ and then pursuing $g$. It is our surrogate for cross-goal transfer. It lower-bounds $V^{\pi}(s, g)$ when $\pi$ is optimal but not in general (Appendix C.4). When the waypoints of a trajectory have high via-goal values, the actions that $\pi_{\mathrm{HG}}$ aggregates toward them are expected to help reach $g$. The next result shows that this surrogate does not degrade during training beyond a tolerance term.

**Theorem 4.3** (Non-Degradation of the Via-Goal Value)**.** *Let $\pi^{(k)}$ be the policy at iteration $k$, and let $\bar{\pi}^{(k)} = \pi^{(k-n_{\mathrm{lag}})}$ be the delayed target policy with lag $n_{\mathrm{lag}} \geq 1$. Assume that goal sets are absorbing under every iterate and that every iterate satisfies Assumption 4.2 with tolerance $\delta$. If $V^{\pi^{(j)}}(s, g) \geq V^{\pi^{(j-1)}}(s, g)$ for all $(s, g)$ and all $j \geq 1$, then for every $k \geq n_{\mathrm{lag}} + 1$ and all $s$, $g$, and $g'$,*

$$V^{\bar{\pi}^{(k)}}_{\mathrm{via}}(s, g, g') \geq V^{\bar{\pi}^{(k-1)}}_{\mathrm{via}}(s, g, g') - 2\delta\, p^{\bar{\pi}^{(k-1)}}(g'|s). \tag{7}$$

*In particular, the via-goal value is non-decreasing when $\delta = 0$.*

The proof in Appendix C.6 factors $V_{\mathrm{via}}$ into the hitting factor $p^{\pi}(g'|s)$ and the residual factor $\mathbb{E}_{d^{\pi}}[V^{\pi}(s', g)]$. Policy improvement cannot lower the hitting factor. Uniform reachability ties the residual factor to the value at one fixed state of $S_{g'}$, and this value cannot decrease either. The two hitting distributions may differ, which costs at most $2\delta$ in the residual factor.

*Remark* 4.4 (Scope and empirical test). Theorem 4.3 concerns the surrogate $V_{\mathrm{via}}$. It neither bounds the implemented loss $\mathcal{L}_{\mathrm{HG}}$ nor proves that the full update improves the policy. Its premises are also idealized. Uniform improvement is guaranteed only for exact policy iteration, our environments are not strictly absorbing, and the algorithm uses a Polyak-averaged target (Appendix C.4). The experiments therefore provide the primary evidence, and Section 6.5 tests directly whether trajectory-aligned waypoints drive cross-goal transfer.

## 5 Tractable Policy Bootstrapping

The KL term of GCHR keeps the policy as its first argument, as in Section 4.1. A single such term cannot impose $\rho_{\mathrm{mix}}$. Take $\lambda \in (0, 1)$ and a policy $\pi$ with a density. The Dirac atom of $\rho_{\mathrm{beh}}$ has zero $\pi$-measure, so $D_{\mathrm{KL}}(\pi\|\rho_{\mathrm{mix}}) = D_{\mathrm{KL}}(\pi\|\rho_{\mathrm{HG}}) - \log(1 - \lambda)$. This objective carries no signal from $\rho_{\mathrm{beh}}$. We therefore impose each component through its own loss.

### 5.1 Behaviour and Goal Bootstrapping Terms

We use a Gaussian policy $\pi_\theta(a|s, g) = \mathcal{N}\big(\mu_\theta(s, g), \mathrm{diag}(\sigma_\theta(s, g))^2\big)$ with mean $\mu_\theta$ and standard deviation $\sigma_\theta$. The behaviour prior becomes a behaviour cloning loss on hindsight data,

$$\mathcal{L}_{\mathrm{beh}}(\theta) = \mathbb{E}_{(s_t, a_t, \hat{g}_t) \sim \mathcal{D}_{\mathrm{HER}}}\left[\|\mu_\theta(s_t, \hat{g}_t) - a_t\|^2\right]. \tag{8}$$

When $\sigma_\theta$ is fixed to a constant $\sigma$ in every dimension, $\mathcal{L}_{\mathrm{beh}}$ equals $2\sigma^2$ times the expected negative log-likelihood of the relabeled actions, up to an additive constant. Minimizing it is thus maximum likelihood estimation of the policy mean on relabeled actions. With a learned $\sigma_\theta$, as in SAC, $\mathcal{L}_{\mathrm{beh}}$ fits only the mean. We leave it unweighted, unlike WGCSL (Yang et al., 2022), AWR (Peng et al., 2019), and AWAC (Nair et al., 2020), because the RL term in Equation (10) already selects actions by their

Q-values. The goal prior enters through the KL term of Section 4.1 with $\rho = \pi_{\mathrm{HG}}$, the aggregated policy in Equation (5),

$$\mathcal{L}_{\mathrm{HG}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, g'_{1:K}\sim\mu_{\tau_x}}\left[D_{\mathrm{KL}}\big(\pi_\theta(\cdot|s_x, g_x)\,\|\,\pi_{\mathrm{HG}}(\cdot|s_x, g_x)\big)\right]. \tag{9}$$

Here $\mathbb{E}_{g'_{1:K}}$ averages over the waypoint draws $g'_{1:K} = (g'_1, \ldots, g'_K)$. Since $\pi_{\mathrm{HG}}$ uses the target parameters $\bar{\theta}$, it is a fixed target for $\theta$. It is a $K$-component Gaussian mixture, so the KL has no closed form. We estimate it without bias from one sample $a \sim \pi_\theta(\cdot|s, g)$ as $\log \pi_\theta(a|s, g) - \log \pi_{\mathrm{HG}}(a|s, g)$ and differentiate through $a = \mu_\theta + \sigma_\theta \odot \xi$ with $\xi \sim \mathcal{N}(0, I)$. Gradients pass through $\pi_{\mathrm{HG}}$ only via $a$, so the target network runs only forward passes. For deterministic backbones (DDPG, TD3), we set $\sigma_\theta = \sigma_{\mathrm{fixed}} = 0.2$, the exploration noise scale, for both $\pi_\theta$ and the components of $\pi_{\mathrm{HG}}$. The next result characterizes what this loss optimizes.

**Proposition 5.1** (Properties of the Goal Loss). *Fix $(s, g)$ with source trajectory $\tau$, let $q$ be any action density, and write $\bar{\pi}_w = \bar{\pi}_{\bar{\theta}}(\cdot|s, w)$ for $w \in \mathcal{G}_H(\tau)$, so that $\pi_{\mathrm{HG}} = \frac{1}{K}\sum_{k=1}^{K} \bar{\pi}_{g'_k}$. (i) For every draw of the waypoints, $D_{\mathrm{KL}}(q\|\pi_{\mathrm{HG}}) \le \min_k D_{\mathrm{KL}}(q\|\bar{\pi}_{g'_k}) + \log K$. (ii) If each $g'_k$ has marginal distribution $\mu_\tau$, then $\mathbb{E}_{g'_{1:K}}[D_{\mathrm{KL}}(q\|\pi_{\mathrm{HG}})] \ge D_{\mathrm{KL}}(q\|\rho_{\mathrm{HG}})$. (iii) If the waypoints are drawn i.i.d. from $\mu_\tau$ and every $\bar{\pi}_{\bar{\theta}}(\cdot|s, w)$ with $w \in \mathcal{G}_H(\tau)$ has full support, then $\mathbb{E}_{g'_{1:K}}[D_{\mathrm{KL}}(q\|\pi_{\mathrm{HG}})]$ is non-increasing in $K$.*

The proof is in Appendix C.7. Part (i) formalizes mode seeking (Bishop & Nasrabadi, 2006, Sec. 10.1.2). A policy that follows the behaviour toward any single waypoint pays at most $\log K$, so the loss does not force $\pi_\theta$ to average incompatible waypoint behaviours. The opposite direction $D_{\mathrm{KL}}(\pi_{\mathrm{HG}}\|\pi_\theta)$ would instead match the moments of the mixture and average them. The RL term decides which mode $\pi_\theta$ follows. Part (ii) shows that, in expectation over the waypoints, $\mathcal{L}_{\mathrm{HG}}$ upper-bounds the KL toward the exact goal prior $\rho_{\mathrm{HG}}$. Part (iii) shows that a larger $K$ tightens this bound at a higher compute cost, a trade-off that Section 6.4 measures.

### 5.2 Complete GCHR Objective

Adding both terms to the actor objective gives

$$\max_\theta \quad \underbrace{\mathbb{E}_{(s,g)\sim\mathcal{D}, a\sim\pi_\theta}[Q^{\pi_\theta}(s, a, g)]}_{\text{RL objective}} - \alpha \underbrace{\mathcal{L}_{\mathrm{beh}}(\theta)}_{\text{behaviour bootstrapping}} - \beta \underbrace{\mathcal{L}_{\mathrm{HG}}(\theta)}_{\text{goal bootstrapping}}, \tag{10}$$

where $\alpha, \beta > 0$ weight the two regularizers and a learned critic $Q_\psi$ with parameters $\psi$ replaces $Q^{\pi_\theta}$ in practice (Algorithm 1). The goal term applies the regularizer of Section 4.1 with $\rho = \pi_{\mathrm{HG}}$ in the actor update only, so Equation (3) motivates GCHR but does not characterize the policy it learns. The three terms play distinct roles. The RL term drives the policy toward actions with high value for $g$. The behaviour term anchors it to recorded actions at seen state-goal pairs, and the goal term proposes actions for pairs without direct experience. The RL term also corrects the priors. If misaligned waypoints propose poor actions, the critic gradient still moves the policy toward better ones. Section 6.5 tests this corrective role.

## 6 Experiments

We test whether GCHR improves sample efficiency and final performance across backbones and benchmarks, and whether its gains come from trajectory-aligned cross-goal transfer.

**Setup.** We use three suites: (i) **OpenAI Gym robotics** (Plappert et al., 2018) with 4 Fetch and 4 Shadow Hand tasks, (ii) **JaxGCRL** (Bortkiewicz et al., 2025) with Pusher Hard, Ant U-Maze, Ant Big Maze, Cheetah, and Ant Soccer, and (iii) image-based, locomotion, and **OGBench** (Park et al., 2025) visual manipulation tasks. Figure 19 in Appendix E.3 shows representative environments. All tasks use sparse binary rewards (Appendix E.3). The backbone is DDPG+HER on Gym (SAC for the RIS comparison), SAC on the visual tasks, and SAC, TD3, or CRL on JaxGCRL. The defaults are $\alpha$=1.0, $\beta$=0.2, $K$=10, and $\tau_{\mathrm{soft}}$=0.05. All results use 5 seeds (Appendix E).

**Baselines.** We compare against DDPG (Lillicrap et al., 2015), DDPG+HER (Andrychowicz et al., 2017), MHER (Yang et al., 2021), GCSL (Ghosh et al., 2021), WGCSL (Yang et al., 2022), RIS (Chane-Sane et al., 2021), CRL (Eysenbach et al., 2022b), SAC(+HER) (Haarnoja et al., 2018), TD3(+HER) (Fujimoto et al., 2018), QRL (Wang et al., 2023), and TD-InfoNCE (Zheng et al., 2024b). Methods within a suite share the same network architecture.

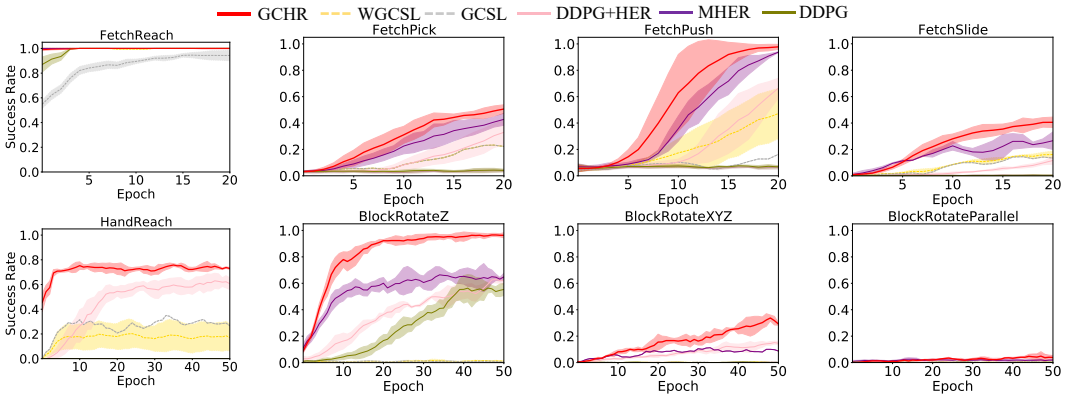



Figure 2: Learning curves on 8 goal-conditioned tasks (DDPG backbone). GCHR matches or exceeds all baselines on every task. The training protocol is in Appendix E. Shaded regions show standard deviation over 5 seeds.

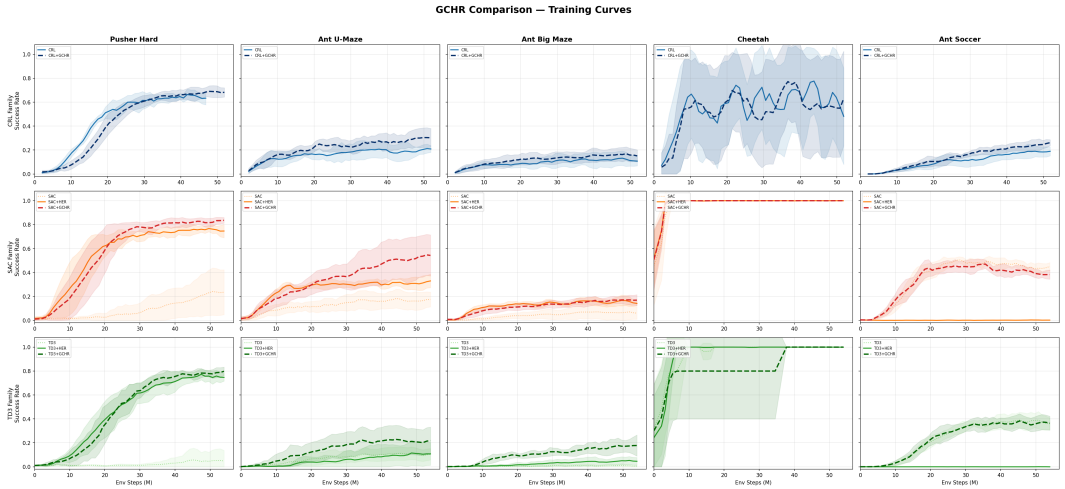



Figure 3: Training curves on JaxGCRL benchmarks. Top row: CRL family. Middle row: SAC family. Bottom row: TD3 family. Adding GCHR improves or matches the final success rate of the baseline on every backbone-task pair. Shaded regions show standard deviation over 5 seeds.

### 6.1 Main Results on OpenAI Gym Robotics

Figure 2 shows the Gym results. The gains over DDPG+HER are largest on FetchPush and FetchSlide, where object interaction makes successful goals rare. On FetchPush, GCHR needs 1.8 to 1.9$\times$ fewer samples than DDPG+HER, depending on the criterion (Figure 14 and Appendix D.10). On FetchSlide, GCHR keeps improving after the strongest baseline, MHER, plateaus, consistent with cross-goal transfer adding guidance beyond imitation. With the SAC backbone, GCHR also matches or outperforms RIS on all Fetch tasks and leads by 17 percentage points (pp) on FetchSlide, although RIS trains an extra subgoal predictor (Appendix D.3). On Hand tasks, the 20-dimensional action space makes exploration hard. GCHR leads on HandReach and BlockRotateZ and clearly outperforms the baselines on BlockRotateXYZ. Absolute success on BlockRotateXYZ and BlockRotateParallel stays low for all methods, and GCHR gains little on BlockRotateParallel (Appendix A).

### 6.2 Backbone-Agnostic Evaluation on JaxGCRL

We next add GCHR to three backbones on JaxGCRL (Bortkiewicz et al., 2025), namely CRL, SAC, and TD3. Figure 3 and Table 1 summarize the results. GCHR improves all three families. The average gain across tasks is 10pp over CRL, 14pp over SAC+HER, and 13pp over TD3+HER. The gains are significant ($p < 0.05$) in 9 of 15 task-backbone pairs. Averaged over backbones, they are largest on Ant Soccer (27.5pp) and Ant U-Maze (14.3pp). These tasks need object manipulation or

Table 1: Absolute gain of GCHR over each baseline in final success rate, in percentage points (pp). Two-sided Welch's $t$-test over 5 seeds: *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$. An orange arrow (↑) marks a positive gain.

| | Pusher Hard | Ant U-Maze | Ant Big Maze | Cheetah | Ant Soccer |
|---|---|---|---|---|---|
| CRL+GCHR vs CRL | $+4.9^{*}$ ↑ | $+9.1^{**}$ ↑ | $+5.0^{**}$ ↑ | $+23.0$ ↑ | $+8.0^{***}$ ↑ |
| SAC+GCHR vs SAC+HER | $+8.0^{***}$ ↑ | $+22.3^{**}$ ↑ | $+3.3^{*}$ ↑ | $+0.0$ | $+38.4^{***}$ ↑ |
| TD3+GCHR vs TD3+HER | $+6.1^{**}$ ↑ | $+11.6^{*}$ ↑ | $+13.2^{**}$ ↑ | $+0.1$ ↑ | $+36.0^{***}$ ↑ |

long-horizon navigation, where cross-goal transfer is expected to help most. On Ant Soccer, HER relabeling drives SAC and TD3 to about 0% success. GCHR keeps HER relabeling and avoids this collapse, and Section 6.5 shows that part of this effect comes from the regularizer itself. On Cheetah, TD3+GCHR reaches the same final success as TD3+HER but converges later.

### 6.3 Image-Based and OGBench Visual Manipulation Tasks

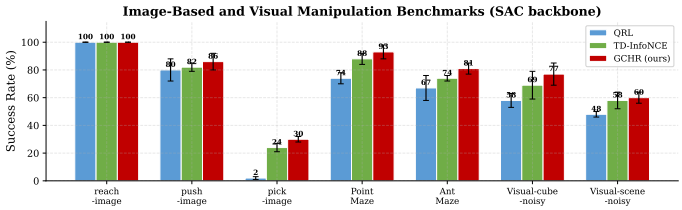



Figure 4: Success rate on image-based, locomotion, and visual manipulation benchmarks (SAC backbone). GCHR matches or outperforms QRL and TD-InfoNCE on all 7 tasks. Error bars show standard deviation over 5 seeds.

We also evaluate SAC+GCHR on image-based, locomotion, and visual manipulation tasks. It matches or outperforms QRL (Wang et al., 2023) and TD-InfoNCE (Zheng et al., 2024b) on all seven tasks and ties at 100% on reach-image (Figure 4). The largest gains over TD-InfoNCE appear on visual-cube-noisy (+8pp), AntMaze (+7pp), and pick-image (+6pp). GCHR thus extends to pixel observations without architectural changes.

### 6.4 Ablation Studies

Figure 5 removes one term at a time. Without goal bootstrapping ($\beta$=0), the behaviour prior guides the actor only at recorded state-goal pairs. Success at epoch 10 on FetchPush then drops from about 63% to 33%. On HandReach, the gap is largest early in training, from about 57% to 32% at epoch 2, and it closes by epoch 10. Without behaviour bootstrapping ($\alpha$=0), the aggregated waypoint behaviour lacks a precise anchor. Success on FetchPush then stays below 10%, and HandReach starts slower. On FetchPush, full GCHR nearly doubles the success of the better single-term variant. The two terms are therefore complementary, as Section 5.2 anticipates.

**Waypoint count and cost.** Computing $\pi_{\mathrm{HG}}$ takes $K$ forward passes of the target actor per sample. On FetchPush, the default $K$=10 adds 22% wall-clock time per step and converges with 1.8× fewer samples than DDPG+HER, a net wall-clock speedup of about 1.5× (Appendix D.10). The speedup saturates between $K$=10 and $K$=15. GCHR also matches or exceeds the baselines across the tested $\alpha$, $\beta$, $K$, and relabeling ratios (Appendix D).

### 6.5 Goal Corruption Experiment

We test whether the goal prior helps through trajectory alignment or through generic regularization. We replace a fraction $\eta_c \in \{0, 0.25, 0.5, 0.75, 1.0\}$ of the $K$=10 waypoints with random replay-buffer goals. We then train SAC+GCHR on Ant U-Maze and Ant Soccer with a 25M-step budget and 5 seeds (Figure 6). Three findings emerge. (i) Mean success decreases from 0.327 with clean waypoints to 0.243 with random ones on Ant U-Maze, and from 0.463 to 0.362 on Ant Soccer, so trajectory-aligned waypoints give the highest mean success on both tasks. (ii) On Ant U-Maze,

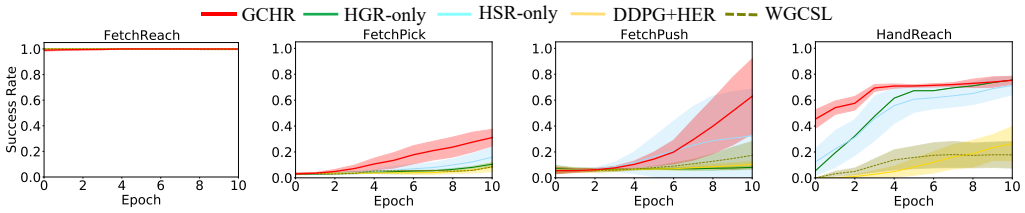



Figure 5: Ablation study. HSR-only ($\beta = 0$) keeps only behaviour bootstrapping (no $\mathcal{L}_{\mathrm{HG}}$). HGR-only ($\alpha = 0$) keeps only goal bootstrapping (no $\mathcal{L}_{\mathrm{beh}}$). Full GCHR matches or exceeds both variants on every task.

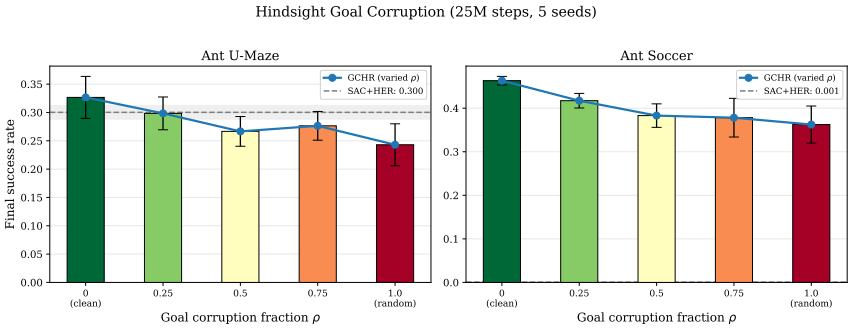



Figure 6: Goal-corruption results. Trajectory-aligned waypoints ($\eta_c = 0$, denoted $\rho$ on the figure axes) give the highest mean success. Error bars show the standard error over 5 seeds. Dashed line: SAC+HER baseline.

fully corrupted GCHR falls *below* SAC+HER (0.300) in mean, which suggests that regularization toward random goals can hurt. (iii) On Ant Soccer, even $\eta_c$=1.0 far exceeds SAC+HER (0.001), so the regularizer alone partly prevents the collapse seen in Section 6.2. The degradation is gradual, which is consistent with the corrective role of the RL term in Equation (10). Aligned waypoints add information on top of this regularization effect, and Figure 16 shows that their advantage is largest early in training.

### 6.6 Direct Evidence for Coverage Expansion

Proposition 4.1 guarantees that the goal prior keeps mass on every waypoint behaviour. It does not guarantee that this mass lies on high-value actions. We therefore test whether $\pi_{\mathrm{HG}}$ puts mass on useful actions outside the behavioural support. For 300 $(s, g)$ pairs, we sample 200 actions from $\pi_{\mathrm{HG}}$ and label each *novel* or *covered* by its $\ell_2$ distance to the nearest $\rho_{\mathrm{beh}}$ action. We then compare the Q-advantage of each group over random actions (Appendices D.2 and D.11). On Reacher (2D actions), novel actions form 18% of the samples and attain a $+0.25$ advantage. Trajectory-aligned waypoints thus propose new actions that the critic values. Covered actions from $\pi_{\mathrm{HG}}$ attain $+1.05$, above the recorded $\rho_{\mathrm{beh}}$ actions ($+0.94$), likely because the goal prior uses the current target policy. On Pusher Easy (higher-dimensional actions), novel actions form 97% of the samples early in training. Their advantage peaks at $+0.91$ near 2.2M steps, but the pooled estimate is $-0.08$. Coverage alone therefore does not ensure value in higher dimensions, and the RL term must select among the proposals. On L-Antmaze (Lee et al., 2022), only GCHR reaches the target region (Appendix D.1).

## 7 Conclusion

We introduced GCHR, a backbone-agnostic framework that turns the structure of past trajectories into guidance for the actor. It regularizes the actor toward a hindsight behaviour prior and a hindsight goal prior. The goal prior extends action coverage, and its KL term upper-bounds the divergence to the exact prior in expectation. Under idealized conditions, the target policy's via-goal value does not degrade during training. Experiments on three suites and four backbones show gains on most tasks, and ablation, goal-corruption, and coverage studies link them to trajectory-aligned cross-goal transfer. The analysis relies on uniform reachability and absorbing goals, which hold only approximately in our environments (Appendix A).

### AI USE STATEMENT

In this work, we used generative AI tools (large language models) to assist in the writing of proofs, to provide feedback on the methodology and experiments, and to help interpret results. Specifically, the tools tightened the exposition of proofs and the wording of assumptions that the authors derived, flagged claims that the reported results did not support, re-computed statistical tests from reported means and standard deviations, and compared the numbers quoted in the text with the values plotted in our figures. We have not used generative AI tools to develop the theoretical framework, formulate the mathematical claims or provide critical steps of their proofs, propose hypotheses, design the method or experiments, implement the method, or translate text, and generating synthetic data, cleaning datasets, and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools to edit the paper for readability, suggest its structure, draft revised sentences, check notation, cross-references, and bibliography entries, format tables, relabel Figure 1, and search for information to verify our descriptions of related work. The authors reviewed all AI-assisted work and checked every proof, statistic, and quoted number against their own derivations, training logs, and figures. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work proposes an actor regularizer for goal-conditioned reinforcement learning and evaluates it only in simulation. All experiments use publicly available benchmarks, namely OpenAI Gym robotics, JaxGCRL, OGBench, the image-based tasks released with TD-InfoNCE, and the L-Antmaze environment from DHRL. The work involves no human subjects, personal data, or newly collected datasets. The main foreseeable risk concerns deployment: policies trained with sparse rewards can behave unpredictably when transferred to physical robots, and we have not evaluated GCHR on real hardware. Real-world use would therefore require standard safeguards such as safe exploration, workspace limits, and human supervision. We are not aware of other risks beyond those that accompany general progress in reinforcement learning.

### REPRODUCIBILITY STATEMENT

Section 4 defines the hindsight behaviour and goal priors, Section 5 derives the losses that GCHR optimizes, and Equation (10) gives the complete objective; Algorithm 1 in Appendix B presents the training loop shared by all backbones. For the theory, Assumption 4.2 and the statements of Proposition 4.1, Theorem 4.3, and Proposition 5.1 list all assumptions, Appendices C.5–C.7 contain complete proofs, and Appendix C.2 and Appendix C.4 discuss when the assumptions hold. Appendix E describes the backbones, baselines, evaluation protocol, random seeds, and environments, and Table 3 lists the hyperparameters on the Gym suite. Appendix D documents the protocols of the coverage, goal-corruption, and wall-clock analyses, and the caption of Table 1 states the statistical test. Our implementations build on public code: the metric-residual-network repository for the Gym baselines, the JaxGCRL benchmark, and the official TD-InfoNCE implementation for the image-based tasks. Upon publication, we commit to release the full source code, configuration files, and evaluation scripts.

## REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Michał Bortkiewicz, Władysław Pałucki, Vivek Myers, Tadeusz Dziarmaga, Tomasz Arczewski, Łukasz Kuciński, and Benjamin Eysenbach. Accelerating goal-conditioned reinforcement learning

algorithms and research. In *The Thirteenth International Conference on Learning Representations*, 2025.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *International conference on machine learning*, pp. 1430–1440. PMLR, 2021.

Leander Diaz-Bone, Marco Bagatella, Jonas Hübotter, and Andreas Krause. Discover: Automated curricula for sparse-reward reinforcement learning. *Advances in Neural Information Processing Systems*, 38:21863–21896, 2026.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. *arXiv preprint arXiv:2011.08909*, 2020.

Benjamin Eysenbach, Soumith Udatha, Russ R Salakhutdinov, and Sergey Levine. Imitating past successes can be very suboptimal. *Advances in Neural Information Processing Systems*, 35: 6047–6059, 2022a.

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, 2022b.

Meng Fang, Tianyi Zhou, Yali Du, Lei Han, and Zhengyou Zhang. Curriculum-guided hindsight experience replay. *Advances in neural information processing systems*, 32, 2019.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Alexandre Galashov, Siddhant M Jayakumar, Leonard Hasenclever, Dhruva Tirumala, Jonathan Schwarz, Guillaume Desjardins, Wojciech M Czarnecki, Yee Whye Teh, Razvan Pascanu, and Nicolas Heess. Information asymmetry in kl-regularized rl. *arXiv preprint arXiv:1905.01240*, 2019.

Dibya Ghosh, Abhishek Gupta, Ashwin Reddy, Justin Fu, Coline Manon Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals via iterated supervised learning. In *International Conference on Learning Representations*, 2021.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Seungjae Lee, Jigang Kim, Inkyu Jang, and H Jin Kim. Dhrl: a graph-based approach for long-horizon and sparse hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 35:13668–13678, 2022.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Yao Li, Yuhui Wang, and XiaoYang Tan. Self-imitation guided goal-conditioned reinforcement learning. *Pattern Recognition*, 144:109845, 2023.

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, pp. 8799–8806, 2023.

Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. *arXiv preprint arXiv:2201.08299*, 2022.

Jason Yecheng Ma, Jason Yan, Dinesh Jayaraman, and Osbert Bastani. Offline goal-conditioned reinforcement learning via $f$-advantage regression. *Advances in Neural Information Processing Systems*, 35:310–323, 2022.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. In *International conference on machine learning*, pp. 3878–3887. PMLR, 2018.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. In *The Thirteenth International Conference on Learning Representations*, 2025.

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.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023.

Yee Teh, Victor Bapst, Wojciech M Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Dhruva Tirumala, Alexandre Galashov, Hyeonwoo Noh, Leonard Hasenclever, Razvan Pascanu, Jonathan Schwarz, Guillaume Desjardins, Wojciech Marian Czarnecki, Arun Ahuja, Yee Whye Teh, et al. Behavior priors for efficient reinforcement learning. *Journal of Machine Learning Research*, 23(221):1–68, 2022.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*, pp. 36411–36430. PMLR, 2023.

Rui Yang, Meng Fang, Lei Han, Yali Du, Feng Luo, and Xiu Li. Mher: Model-based hindsight experience replay. *arXiv preprint arXiv:2107.00306*, 2021.

Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking goal-conditioned supervised learning and its connection to offline rl. *arXiv preprint arXiv:2202.04478*, 2022.

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 *The Twelfth International Conference on Learning Representations*, 2024a.

Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive difference predictive coding. In *The Twelfth International Conference on Learning Representations*, 2024b.

John Zhou and Jonathan Kao. Flattening hierarchies with policy bootstrapping. *Advances in Neural Information Processing Systems*, 38:82975–83007, 2026.

# CONTENTS OF APPENDIX

**A Limitations** **14**

**B Algorithm** **14**

**C Theoretical Analysis** **15**

C.1 Notation Summary . . . . . 15

C.2 Extended Discussion of Assumption 4.2 . . . . . 15

C.3 Extended Discussion of Proposition 4.1 . . . . . 16

C.4 Extended Discussion of Theorem 4.3 . . . . . 16

C.5 Proof of Proposition 4.1 . . . . . 17

C.6 Proof of Theorem 4.3 . . . . . 17

C.7 Proof of Proposition 5.1 . . . . . 17

**D Additional Results** **18**

D.1 Trajectory-Level Coverage Visualization . . . . . 18

D.2 Coverage Expansion Protocol . . . . . 19

D.3 Comparison with RIS . . . . . 19

D.4 Robustness to Environment Stochasticity . . . . . 20

D.5 The Impact of Hyperparameter $\beta$ and $\alpha$ . . . . . 20

D.6 The Impact of the Waypoint Pool . . . . . 21

D.7 Sample Efficiency . . . . . 21

D.8 Relabeling Ratio . . . . . 21

D.9 Learning Curves Under Goal Corruption . . . . . 22

D.10 Wall-Clock Overhead Analysis . . . . . 23

D.11 Q-Landscape Visualization . . . . . 23

**E Implementation Details** **24**

E.1 Algorithm and Architecture . . . . . 24

E.2 Evaluation Setup and Hyperparameters . . . . . 25

E.3 Environment Details . . . . . 26

## A LIMITATIONS

The uniform reachability condition (Assumption 4.2) may fail under strongly irreversible dynamics, where $\pi_{\mathrm{HG}}$ could steer the agent toward dead-end waypoints. The RL term in Equation (10) offers a partial safeguard. We expect the condition to be loose for multi-axis in-hand rotation, where all methods reach low absolute success and GCHR gains little on BlockRotateParallel. We have not estimated $\delta$ on our tasks, so we have not verified that the condition causes this small gain, although the learned critic could provide such an estimate. Theorem 4.3 identifies $p^{\pi}$ with the discounted first-hitting probability, which requires absorbing goals. Our environments do not realize absorbing goals exactly, so the experiments are the primary evidence for the quality of cross-goal transfer (Remark 4.4). The theorem also concerns the surrogate $V_{\mathrm{via}}$ and does not bound the implemented loss $\mathcal{L}_{\mathrm{HG}}$. Proposition 5.1 relates $\mathcal{L}_{\mathrm{HG}}$ to the exact goal prior, but it does not connect the loss to $V_{\mathrm{via}}$. Bridging this gap is an important direction for future work. Other directions include reachability-weighted waypoint sampling and deployment on real robots. Our evaluation focuses on sparse-reward tasks with state or image goals, and open-ended domains with language goals remain untested.

## B ALGORITHM

Algorithm 1 presents the shared skeleton of GCHR. Backbone-specific actor and critic targets, such as the SAC entropy term and TD3 clipped double-Q learning, are instantiated in the standard manner.

**Algorithm 1** GCHR: Goal-Conditioned Hindsight Regularization

**Initialize:** replay buffers $\mathcal{D}_{\mathrm{orig}}$ and $\mathcal{D}_{\mathrm{HER}}$, policy $\pi_\theta$, Q-function $Q_\psi$
**Initialize:** target networks $\bar{\theta} \leftarrow \theta$, $\bar{\psi} \leftarrow \psi$
**Hyperparameters:** $\alpha$ (behaviour weight), $\beta$ (goal weight), $K$ (waypoints), $\tau_{\mathrm{soft}}$ (soft update rate), $\eta_Q$ and $\eta_\pi$ (learning rates)
**while** not converged **do**
  Sample a goal $g$ from the task goal distribution and an initial state $s_0$
  Collect trajectory $\tau = (s_0, a_0, \ldots, s_T)$ using $\pi_\theta(\cdot|s_t, g)$
  Store $\tau$ in $\mathcal{D}_{\mathrm{orig}}$ and add its relabeled transitions to $\mathcal{D}_{\mathrm{HER}}$, with $\mathcal{D} = \mathcal{D}_{\mathrm{orig}} \cup \mathcal{D}_{\mathrm{HER}}$
  **for** $i = 1, \ldots, I$ gradient steps **do**
    **// Critic Update** (backbone-specific. SAC adds an entropy term, and TD3 uses clipped double-Q)
    Sample $(s_t, a_t, s_{t+1}, g) \sim \mathcal{D}$
    $y_t = r(s_t, g) + \gamma Q_{\bar{\psi}}(s_{t+1}, \bar{\mu}_{\bar{\theta}}(s_{t+1}, g), g)$, where $\bar{\mu}_{\bar{\theta}}$ is the mean of $\bar{\pi}_{\bar{\theta}}$
    $\psi \leftarrow \psi - \eta_Q \nabla_\psi (Q_\psi(s_t, a_t, g) - y_t)^2$
    **// Behaviour Bootstrapping**
    Sample $(s_t, a_t, \hat{g}_t) \sim \mathcal{D}_{\mathrm{HER}}$
    $\mathcal{L}_{\mathrm{beh}} = \|\mu_\theta(s_t, \hat{g}_t) - a_t\|^2$
    **// Goal Bootstrapping**
    Sample $(s, g) \sim \mathcal{D}$ with its source trajectory $\tau$, and sample $K$ waypoints $\{g'_k\}_{k=1}^K$ uniformly from $\mathcal{G}_H(\tau)$
    Compute $\pi_{\mathrm{HG}}(a|s, g) = \frac{1}{K} \sum_{k=1}^K \bar{\pi}_{\bar{\theta}}(a|s, g'_k)$ via the target network
    Sample $a \sim \pi_\theta(\cdot|s, g)$  // For deterministic backbones: $a = \mu_\theta(s, g) + \sigma_{\mathrm{fixed}}\xi$, $\xi \sim \mathcal{N}(0, I)$
    $\mathcal{L}_{\mathrm{HG}} = \log \pi_\theta(a|s, g) - \log \pi_{\mathrm{HG}}(a|s, g)$
    **// Policy Update**
    $\theta \leftarrow \theta + \eta_\pi \nabla_\theta [\mathbb{E}_{a \sim \pi_\theta}[Q_\psi(s, a, g)] - \alpha \mathcal{L}_{\mathrm{beh}} - \beta \mathcal{L}_{\mathrm{HG}}]$
  **end for**
  **// Target Network Update**
  $\bar{\theta} \leftarrow \tau_{\mathrm{soft}}\theta + (1 - \tau_{\mathrm{soft}})\bar{\theta}$,  $\bar{\psi} \leftarrow \tau_{\mathrm{soft}}\psi + (1 - \tau_{\mathrm{soft}})\bar{\psi}$
**end while**

# C Theoretical Analysis

## C.1 Notation Summary

Table 2 consolidates the symbols used in the analysis. Symbols introduced only within a single proof are not listed.

Table 2: Notation used in the theoretical analysis.

| Symbol | Meaning |
|---|---|
| $s,\ a,\ g$ | state, action, and goal |
| $\gamma$ | discount factor, $\gamma \in [0, 1)$ |
| $\phi$ | map from a state to the goal it achieves |
| $S_g$ | states that satisfy $g$, i.e., $\{s : \|\phi(s) - g\| \le \epsilon\}$ (absorbing in Section 4.2) |
| $T_g$ | first time the state enters $S_g$, with $\gamma^{T_g} = 0$ if it never does |
| $\pi(\cdot \mid s, g)$ | goal-conditioned policy |
| $\pi^{(k)}$ | online policy at training iteration $k$ |
| $\bar{\pi}^{(k)}$ | target policy at iteration $k$ (hard copy $\pi^{(k-n_{\text{lag}})}$ in the analysis) |
| $\bar{\pi}_{\bar{\theta}}$ | target policy with Polyak-averaged parameters $\bar{\theta}$ |
| $n_{\text{lag}},\ \tau_{\text{soft}}$ | hard target-update lag and Polyak coefficient |
| $\rho_{\text{beh}}$ | hindsight behaviour prior, a Dirac measure on a relabeled action |
| $\rho_{\text{HG}},\ \pi_{\text{HG}}$ | hindsight goal prior and its $K$-sample estimate (Equation (5)) |
| $\rho_{\text{mix}},\ \lambda$ | conceptual mixture of $\rho_{\text{beh}}$ and $\rho_{\text{HG}}$, and its weight |
| $\mathcal{G}_H(\tau),\ H$ | multiset $\{\phi(s_0), \ldots, \phi(s_T)\}$ of goals visited along $\tau$ and its size $T + 1$ |
| $\mu_\tau,\ K$ | uniform distribution over $\mathcal{G}_H(\tau)$ and number of sampled waypoints |
| $V^\pi(s, g)$ | goal-conditioned value |
| $p^\pi(g \mid s)$ | discounted reaching probability, equal to $\mathbb{E}[\gamma^{T_g}]$ for absorbing $S_g$ |
| $d^\pi(\cdot \mid s, g')$ | discount-weighted distribution of the hitting state $s_{T_{g'}}$ |
| $V^\pi_{\text{via}}(s, g, g')$ | via-goal value of reaching $g$ through waypoint $g'$ |
| $\delta$ | tolerance in Assumption 4.2 |
| $\mathcal{D}_{\text{HER}},\ \mathcal{D}_{\text{orig}},\ \mathcal{D}$ | relabeled transitions, original transitions, and their union |
| $x = (s_x, a_x, s_x^+, g_x, \tau_x)$ | replay element with state, action, next state, goal, and source trajectory |
| $\mathcal{A}_{\text{beh}}(x),\ \mathcal{A}_{\text{HG}}(x)$ | behaviour and hindsight goal action sets (Section 4.1) |
| $\mathcal{L}_{\text{beh}},\ \mathcal{L}_{\text{HG}}$ | behaviour and goal bootstrapping losses (Equation (10)) |
| $\alpha,\ \beta$ | weights of $\mathcal{L}_{\text{beh}}$ and $\mathcal{L}_{\text{HG}}$ |

## C.2 Extended Discussion of Assumption 4.2

**Policy dependence.** The condition depends on the policy through $V^\pi$. Our results require it only for the iterates $\{\pi^{(k)}\}_{k \ge 0}$ to which they apply. This requirement is strong and rarely verifiable exactly. It is most natural when goal-relevant features, such as end-effector or object positions, determine reachability more strongly than goal-irrelevant features, such as joint configurations or velocities.

**Size of the tolerance.** Rewards lie in $\{0, 1\}$, so $0 \le V^\pi \le 1/(1-\gamma)$ for every policy. Every policy thus satisfies Assumption 4.2 with $\delta = 1/(1-\gamma)$. By Equation (6), we also have $V^\pi_{\text{via}}(s, g, g') \le p^\pi(g'|s)/(1-\gamma)$. With $\delta = 1/(1-\gamma)$, the right-hand side of Equation (7) is therefore at most $-p^{\bar{\pi}^{(k-1)}}(g'|s)/(1-\gamma) \le 0$, and the bound holds trivially. Theorem 4.3 becomes informative only as $\delta$ decreases. When goals are full states, the condition reduces to a continuity property. In that case $\phi$ is the identity, $\mathcal{G} = \mathcal{S}$, and $S_{g'}$ is the ball of radius $\epsilon$ around $g'$. Any two states in this ball lie within distance $2\epsilon$ of each other. If $V^\pi(\cdot, g)$ is $L$-Lipschitz for every $g$, the condition thus holds with $\delta = 2L\epsilon$. The condition becomes restrictive mainly when $\phi$ discards features that affect reachability.

**Scope of the assumption.** Assumption 4.2 does *not* claim that reaching $g'$ ensures reaching $g$. It states that the residual value $V^\pi(s', g)$ at an arrival state $s' \in S_{g'}$ is nearly independent of the arrival state. This property stabilizes the via-goal decomposition in Theorem 4.3, and the tolerance $\delta$ enters its bound linearly. The practical acceleration comes from coverage (Proposition 4.1). The prior proposes diverse candidate actions, and the RL objective steers the policy toward the useful

ones. Empirically, GCHR performs well on Fetch, HandReach, and navigation, where the condition is plausible. On multi-axis in-hand rotation, where finger configurations also affect reachability, all methods reach low absolute success; GCHR still leads on BlockRotateXYZ but gains little on BlockRotateParallel. Since we have not estimated $\delta$ (Appendix A), this link to the condition remains a conjecture.

### C.3 Extended Discussion of Proposition 4.1

For policies with full support, such as Gaussians, $\mathcal{A}_{\mathrm{HG}}(x)$ is the whole action space because their densities are positive everywhere. Parts (i) and (ii) are then uninformative, and the meaningful notion of coverage concerns probability mass. Part (iii) addresses mass. It guarantees that $\rho_{\mathrm{HG}}$ keeps at least a $1/H$ share of the mass that each waypoint behaviour places on any action set. The same argument gives $\pi_{\mathrm{HG}}(B|s,g) \geq \bar{\pi}_{\bar{\theta}}(B|s,g'_k)/K$ for each sampled waypoint $g'_k$. Part (iii) does not say whether this mass lies on high-Q actions outside the high-density region of $\rho_{\mathrm{beh}}$. Section 6.6 tests this question. On Reacher, novel actions from $\pi_{\mathrm{HG}}$ achieve a positive Q-advantage over random actions. On Pusher Easy, the pooled advantage of novel actions is slightly negative, so coverage alone does not ensure value there.

**Relation between the two priors.** Consider a relabeled pair $(s_t, \hat{g}_t)$ with $K = 1$, $\tau_{\mathrm{soft}} = 1$, and a vanishing policy variance. The behaviour prior is $\delta_{a_t}$, whereas the goal prior tends to the point mass at $\mu_\theta(s_t, g'_1)$ for one waypoint $g'_1 \sim \mu_\tau$. The two coincide only if $\mu_\theta(s_t, g'_1) = a_t$, that is, only if the policy reproduces the recorded action at the sampled waypoint. The two losses also act on different quantities. The behaviour loss fits the policy to recorded data. The goal loss fits the policy at one goal to its own behaviour at another goal, and it never evaluates a recorded action. In this setting, if the sampled waypoint equals the goal, the goal loss vanishes because $\tau_{\mathrm{soft}} = 1$ makes the target an exact copy of the policy; with $\tau_{\mathrm{soft}} < 1$, it reduces to a KL toward the lagged policy at the same goal.

**Waypoints before the current state.** The distribution $\mu_\tau$ also covers waypoints visited before $s$. Their behaviours enter $\pi_{\mathrm{HG}}$ as separate modes, and the mode-seeking goal loss of Section 5 together with the RL term lets the policy leave such modes unused.

### C.4 Extended Discussion of Theorem 4.3

This section relates $V_{\mathrm{via}}$ to the value of the target goal and then expands on the gaps that Remark 4.4 lists.

**Relation to the target value.** Let $\pi$ be optimal for every goal, and fix $s$, $g$, and $g'$. Consider the policy for goal $g$ that follows $\pi(\cdot|\cdot, g')$ until time $T_{g'}$ and $\pi(\cdot|\cdot, g)$ afterwards. Rewards are non-negative, so its value for $g$ is at least $\mathbb{E}[\gamma^{T_{g'}} V^\pi(s_{T_{g'}}, g)] = V^\pi_{\mathrm{via}}(s, g, g')$. In a discounted MDP, no policy exceeds the optimal value, including policies that depend on the history. Hence $V^\pi_{\mathrm{via}}(s, g, g') \leq V^\pi(s, g)$. For a general policy, the switching policy can outperform $\pi(\cdot|\cdot, g)$, so no such bound holds.

**Absorbing goals.** The proof identifies $p^\pi(g'|s)$ with the discounted first-hitting probability $\mathbb{E}[\gamma^{T_{g'}}]$, which requires absorbing goals. Without absorbing goals, $p^\pi(g'|s)$ is a discounted occupancy of $S_{g'}$. A policy can then raise this occupancy by staying longer in $S_{g'}$ even if it enters $S_{g'}$ later. A higher occupancy thus no longer implies a higher discounted hitting probability, and the conclusion may fail.

**Uniform improvement.** Exact policy iteration improves the value at every state-goal pair. With function approximation, an update can lower the value at some pairs. For fixed $s$, $g$, and $g'$, however, the proof uses this premise only at $(s, g')$ and at one state of $S_{g'}$ for goal $g$. Improvement at these two pairs therefore suffices for Equation (7) at $(s, g, g')$.

**Polyak averaging.** The theorem treats the target policy as a delayed hard copy $\bar{\pi}^{(k)} = \pi^{(k - n_{\mathrm{lag}})}$. The algorithm uses Polyak updates, under which $\bar{\theta}$ is an exponential moving average of past parameters. The theorem then applies only approximately, namely when consecutive iterates change slowly relative to the averaging window.

**From the surrogate to the update.** The theorem bounds the change of $V_{\mathrm{via}}$ between consecutive target policies. It does not show that the update in Equation (10) improves the policy, and it does not bound $\mathcal{L}_{\mathrm{HG}}$. Such a guarantee would require a performance difference lemma for the mixed objective, which we leave to future work.

### C.5 Proof of Proposition 4.1

*Proof.* **Part (i).** Let $x \in \mathcal{D}_{\text{HER}}$. Its goal is a relabeled goal $g_x = \phi(s_{t'})$ for some state $s_{t'}$ of the source trajectory $\tau_x$, so $g_x \in \mathcal{G}_H(\tau_x)$. Take any $a \in \mathcal{A}_{\text{beh}}(x)$. By definition, $(s_x, a, \cdot, g_x) \in \mathcal{D}_{\text{HER}}$, and the positivity condition gives $\bar{\pi}_{\bar{\theta}}(a|s_x, g_x) > 0$. The witness $w = g_x$ then shows

$$a \in \{a : \exists w \in \mathcal{G}_H(\tau_x), \bar{\pi}_{\bar{\theta}}(a|s_x, w) > 0\} = \mathcal{A}_{\text{HG}}(x). \tag{11}$$

Hence $\mathcal{A}_{\text{beh}}(x) \subseteq \mathcal{A}_{\text{HG}}(x)$.

**Part (ii).** Let $x \in \mathcal{D}$. The multiset $\mathcal{G}_H(\tau_x)$ contains $\phi(s_0)$ and is therefore non-empty. Fix any $w \in \mathcal{G}_H(\tau_x)$. Since $\bar{\pi}_{\bar{\theta}}(\cdot|s_x, w)$ is a probability density, it is positive on a set of positive measure. Every action in this set belongs to $\mathcal{A}_{\text{HG}}(x)$, so $\mathcal{A}_{\text{HG}}(x) \neq \emptyset$ regardless of $\mathcal{A}_{\text{beh}}(x)$. Note that $a_x \in \mathcal{A}_{\text{beh}}(x)$ for every $x \in \mathcal{D}_{\text{HER}}$. An empty behaviour set can therefore arise only for elements of $\mathcal{D}_{\text{orig}}$, and for these elements the goal prior is the only prior that supplies guidance.

**Part (iii).** Let $x \in \mathcal{D}$, let $w \in \mathcal{G}_H(\tau_x)$, and let $B \subseteq \mathcal{A}$ be measurable. The distribution $\mu_{\tau_x}$ is uniform over the $H$ elements of $\mathcal{G}_H(\tau_x)$, counted with multiplicity. Integrating Equation (5) over $B$ therefore gives

$$\rho_{\text{HG}}(B|s_x, g_x) = \frac{1}{H} \sum_{w' \in \mathcal{G}_H(\tau_x)} \bar{\pi}_{\bar{\theta}}(B|s_x, w') \geq \frac{1}{H}\, \bar{\pi}_{\bar{\theta}}(B|s_x, w), \tag{12}$$

where the sum runs over the $H$ elements and the inequality keeps only one term with $w' = w$. All other terms are non-negative. Applying the same argument to $\pi_{\text{HG}}$ gives the bound with $K$ in place of $H$ for each sampled waypoint. □

### C.6 Proof of Theorem 4.3

*Proof.* Fix $s$, $g$, $g'$ and $k \geq n_{\text{lag}} + 1$. Let $\pi_1 = \bar{\pi}^{(k)} = \pi^{(k-n_{\text{lag}})}$ and $\pi_0 = \bar{\pi}^{(k-1)} = \pi^{(k-n_{\text{lag}}-1)}$. Both are training iterates because $k - n_{\text{lag}} - 1 \geq 0$. The premise at iteration $j = k - n_{\text{lag}} \geq 1$ gives

$$V^{\pi_1}(\tilde{s}, \tilde{g}) \geq V^{\pi_0}(\tilde{s}, \tilde{g}) \quad \text{for all } (\tilde{s}, \tilde{g}). \tag{13}$$

We write $p_i = p^{\pi_i}(g'|s)$ for $i \in \{0, 1\}$.

**Hitting factor.** By Section 3, $V^{\pi}(s, g') = p^{\pi}(g'|s)/(1-\gamma)$ for any policy. Applying Equation (13) at $(s, g')$ gives $p_1 \geq p_0$. Under absorbing goal sets, $p_i = \mathbb{E}[\gamma^{T_{g'}}]$ under $\pi_i$.

**Degenerate case.** Rewards lie in $\{0, 1\}$, so $0 \leq V^{\pi_i} \leq 1/(1-\gamma)$ and hence $V^{\pi_i}_{\text{via}}(s, g, g') \geq 0$. If $p_0 = 0$, then $\gamma^{T_{g'}} = 0$ almost surely under $\pi_0$. Hence $V^{\pi_0}_{\text{via}}(s, g, g') = 0 \leq V^{\pi_1}_{\text{via}}(s, g, g')$, and Equation (7) holds. We assume $p_0 > 0$ from now on, which also gives $p_1 > 0$. Both hitting distributions $d^{\pi_i}(\cdot|s, g')$ are then well defined.

**Residual factor.** Let $m_i = \mathbb{E}_{s' \sim d^{\pi_i}(\cdot|s,g')}[V^{\pi_i}(s', g)]$, so that $V^{\pi_i}_{\text{via}}(s, g, g') = p_i\, m_i$ by Equation (6). Fix any $s^\star \in S_{g'}$. Each $d^{\pi_i}(\cdot|s, g')$ is supported on $S_{g'}$, and Assumption 4.2 gives $|V^{\pi_i}(s', g) - V^{\pi_i}(s^\star, g)| \leq \delta$ for every $s' \in S_{g'}$. Averaging over $s' \sim d^{\pi_i}(\cdot|s, g')$ yields $|m_i - V^{\pi_i}(s^\star, g)| \leq \delta$. The two hitting distributions may differ, but both residual values lie within $\delta$ of values at the same state $s^\star$. Applying Equation (13) at $(s^\star, g)$ gives

$$m_1 \geq V^{\pi_1}(s^\star, g) - \delta \geq V^{\pi_0}(s^\star, g) - \delta \geq m_0 - 2\delta. \tag{14}$$

**Combination.** We decompose

$$V^{\pi_1}_{\text{via}}(s, g, g') - V^{\pi_0}_{\text{via}}(s, g, g') = p_1 m_1 - p_0 m_0 = (p_1 - p_0)\, m_1 + p_0\, (m_1 - m_0). \tag{15}$$

The first term is non-negative because $p_1 \geq p_0$ and $m_1 \geq 0$. The second term is at least $-2\delta p_0$ because $p_0 \geq 0$ and $m_1 - m_0 \geq -2\delta$. Hence $V^{\pi_1}_{\text{via}}(s, g, g') \geq V^{\pi_0}_{\text{via}}(s, g, g') - 2\delta p_0$. Since $\pi_1 = \bar{\pi}^{(k)}$ and $\pi_0 = \bar{\pi}^{(k-1)}$, this inequality is Equation (7). For $\delta = 0$, its right-hand side equals $V^{\pi_0}_{\text{via}}(s, g, g')$, so the via-goal value is non-decreasing. □

### C.7 Proof of Proposition 5.1

*Proof.* The pair $(s, g)$ and the trajectory $\tau$ are fixed, and we drop them from the notation. We write $\bar{\pi}_w = \bar{\pi}_{\bar{\theta}}(\cdot|s, w)$ for $w \in \mathcal{G}_H(\tau)$ and $M_K = \frac{1}{K} \sum_{k=1}^{K} \bar{\pi}_{g'_k}$, so that $M_K = \pi_{\text{HG}}$. For densities $q$ and

$p$, we use

$$D_{\mathrm{KL}}(q\|p) = \int f_q(a;p)\,da, \qquad f_q(a;p) = q(a)\log\frac{q(a)}{p(a)} - q(a) + p(a). \tag{16}$$

The identity holds because $q$ and $p$ both integrate to one. The integrand is non-negative, since $u\log(u/v) - u + v \geq 0$ for all $u, v \geq 0$, with the conventions $0\log 0 = 0$ and $u\log(u/0) = +\infty$ for $u > 0$. All divergences below therefore take values in $[0, +\infty]$.

**Part (i).** Fix $k$. Every term of $M_K$ is non-negative, so $M_K(a) \geq \bar{\pi}_{g'_k}(a)/K$ for every $a$. Hence $q(a)\log(q(a)/M_K(a)) \leq q(a)\log(q(a)/\bar{\pi}_{g'_k}(a)) + q(a)\log K$, and

$$f_q(a; M_K) \leq f_q(a; \bar{\pi}_{g'_k}) + q(a)\log K + M_K(a) - \bar{\pi}_{g'_k}(a). \tag{17}$$

If $D_{\mathrm{KL}}(q\|\bar{\pi}_{g'_k}) = +\infty$, the bound in (i) holds trivially. Otherwise, every term on the right is integrable, and $M_K - \bar{\pi}_{g'_k}$ integrates to zero. Integrating both sides gives $D_{\mathrm{KL}}(q\|M_K) \leq D_{\mathrm{KL}}(q\|\bar{\pi}_{g'_k}) + \log K$. Taking the minimum over $k$ proves (i).

**Part (ii).** Each $g'_k$ has marginal distribution $\mu_\tau$, so $\mathbb{E}[\bar{\pi}_{g'_k}(a)] = \rho_{\mathrm{HG}}(a)$ for every $a$, and hence $\mathbb{E}[M_K(a)] = \rho_{\mathrm{HG}}(a)$. The logarithm is concave, so Jensen's inequality gives $\mathbb{E}[\log M_K(a)] \leq \log \rho_{\mathrm{HG}}(a)$. For every $a$ with $q(a) > 0$, this yields

$$\mathbb{E}\big[f_q(a; M_K)\big] = q(a)\log q(a) - q(a)\,\mathbb{E}[\log M_K(a)] - q(a) + \rho_{\mathrm{HG}}(a) \geq f_q(a; \rho_{\mathrm{HG}}). \tag{18}$$

Where $q(a) = 0$, both sides equal $\rho_{\mathrm{HG}}(a)$. The multiset $\mathcal{G}_H(\tau)$ is finite, so the expectation over $g'_{1:K}$ is a finite weighted sum of non-negative functions and commutes with the integral over actions. Therefore

$$\mathbb{E}\big[D_{\mathrm{KL}}(q\|M_K)\big] = \int \mathbb{E}\big[f_q(a; M_K)\big]\,da \geq \int f_q(a; \rho_{\mathrm{HG}})\,da = D_{\mathrm{KL}}(q\|\rho_{\mathrm{HG}}). \tag{19}$$

**Part (iii).** Let $g'_1, \ldots, g'_{K+1}$ be i.i.d. draws from $\mu_\tau$, and let $M_K^{(-j)}$ be the average of the $K$ components with index $i \neq j$. Each component appears in exactly $K$ of the $K+1$ averages $M_K^{(-1)}, \ldots, M_K^{(-(K+1))}$, so

$$M_{K+1}(a) = \frac{1}{K+1}\sum_{j=1}^{K+1} M_K^{(-j)}(a). \tag{20}$$

Full support makes every $\log M_K^{(-j)}(a)$ finite. Concavity of the logarithm gives $\log M_{K+1}(a) \geq \frac{1}{K+1}\sum_{j=1}^{K+1}\log M_K^{(-j)}(a)$. Each $M_K^{(-j)}$ averages $K$ i.i.d. components and has the same distribution as $M_K$. The expectations below are finite sums because $\mathcal{G}_H(\tau)$ is finite. Taking expectations yields $\mathbb{E}[\log M_{K+1}(a)] \geq \mathbb{E}[\log M_K(a)]$. As in part (ii), $\mathbb{E}[M_{K+1}(a)] = \mathbb{E}[M_K(a)] = \rho_{\mathrm{HG}}(a)$, so $\mathbb{E}[f_q(a; M_{K+1})] \leq \mathbb{E}[f_q(a; M_K)]$ for every $a$. Integrating over $a$ as in part (ii) shows that $\mathbb{E}[D_{\mathrm{KL}}(q\|M_K)]$ is non-increasing in $K$. The same argument holds for waypoints drawn without replacement, because such draws are exchangeable. In that case $M_H = \rho_{\mathrm{HG}}$, so the bound in (ii) is tight at $K = H$. □

# D Additional Results

This section evaluates the robustness of GCHR to the hyperparameters $\beta$ and $\alpha$, the waypoint pool, the relabeling ratio, and environment stochasticity. It also details the coverage, goal-corruption, sample-efficiency, and wall-clock analyses referenced in the main text.

## D.1 Trajectory-Level Coverage Visualization

Figure 7 visualizes achieved goals during training on L-Antmaze. DDPG+HER and WGCSL remain near the starting region. MHER expands further through model-based goal generation but stalls, likely because of compounding errors in its learned dynamics model. GCHR progressively expands coverage toward the target region in the top-right corner by bootstrapping from intermediate goals along the L-shaped corridor. Each newly reached waypoint enriches the goal prior for later updates, which is consistent with the spread of achieved goals toward the target in the rightmost panel. This state-space pattern is consistent with, though not implied by, the action coverage of Proposition 4.1.

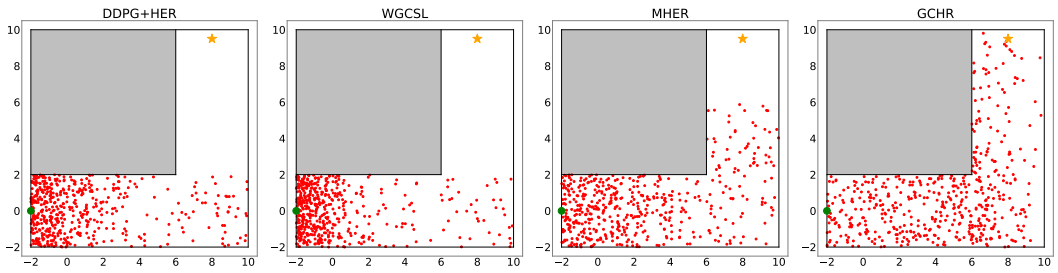



Figure 7: Goal coverage visualization in L-Antmaze (Lee et al., 2022). From left to right: DDPG+HER, WGCSL, MHER, GCHR. Only GCHR reaches the target region (top-right corner), in agreement with the coverage expansion induced by waypoint-based bootstrapping.

### D.2 Coverage Expansion Protocol

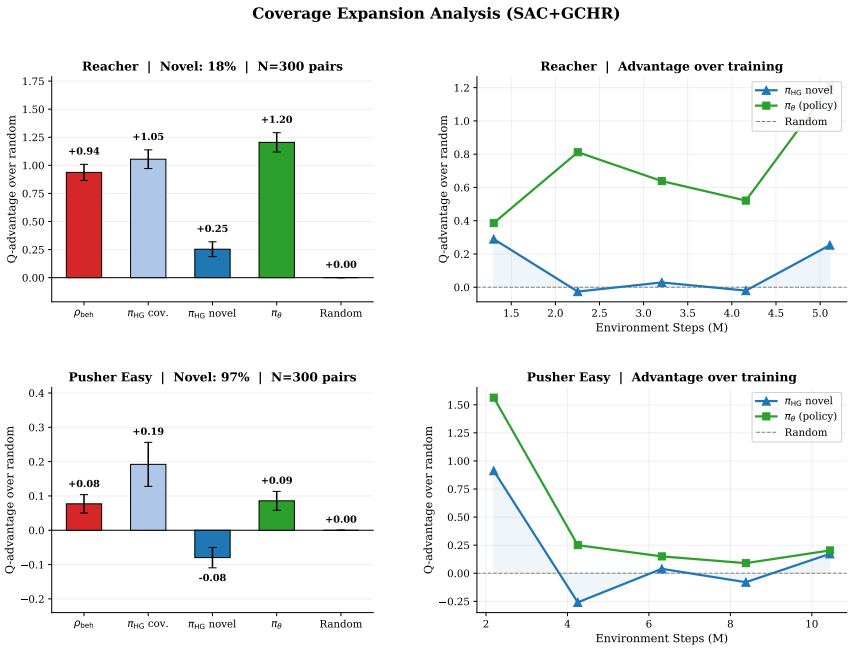



Figure 8: Coverage analysis for SAC+GCHR. **Left column**: Q-advantage over random actions by action source, pooled over 300 state-goal pairs. **Right column**: Q-advantage of novel $\pi_{\mathrm{HG}}$ actions and of the policy during training. Top: Reacher (2D actions). Bottom: Pusher Easy (higher-dimensional actions).

We evaluate SAC+GCHR on Reacher (2D actions) and Pusher Easy (higher-dimensional actions). For 300 $(s, g)$ pairs, we sample 200 actions from $\pi_{\mathrm{HG}}$. An action is *novel* if its $\ell_2$ distance to the nearest $\rho_{\mathrm{beh}}$ action exceeds a fixed threshold, and *covered* otherwise. For each action source, we report the Q-advantage over random actions. The left column of Figure 8 pools all pairs, and the right column tracks the advantage during training. On Reacher, the current policy $\pi_\theta$ attains $+1.20$, above every prior. On Pusher Easy, covered actions from $\pi_{\mathrm{HG}}$ attain $+0.19$, above both $\rho_{\mathrm{beh}}$ ($+0.08$) and $\pi_\theta$ ($+0.09$). As the behavioural support grows during training, the novel fraction on Pusher Easy decreases from 97% to 90%.

### D.3 Comparison with RIS

RIS (Chane-Sane et al., 2021) also regularizes the policy toward a subgoal-conditioned prior but requires a separately trained subgoal prediction network. We compare GCHR with RIS using the same SAC backbone on Fetch tasks (Figure 9).

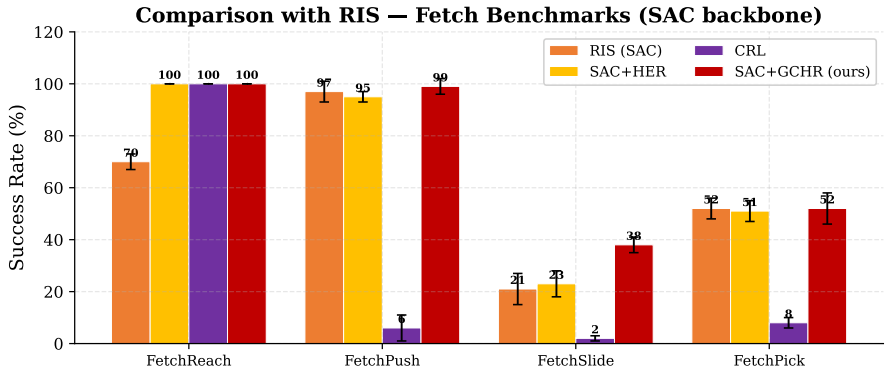



Figure 9: Success rate (%) on Fetch benchmarks. RIS, SAC+HER, and SAC+GCHR use the SAC backbone, and CRL is shown for reference. SAC+GCHR matches or outperforms RIS on all tasks without a separate subgoal prediction network. Error bars show standard deviation over 5 seeds.

SAC+GCHR matches or outperforms RIS on all tasks, with a 17pp advantage on FetchSlide, and it needs no subgoal prediction network. FetchSlide requires precise long-distance object displacement, where intermediate waypoint knowledge is most valuable. RIS relies on a learned subgoal predictor that can introduce compounding errors. GCHR aggregates the target policy's behaviour toward waypoints that were actually visited. CRL performs poorly on Fetch manipulation (6 to 8% on Push and Pick) despite strong locomotion results, possibly because its contrastive objective does not capture the object-centric goal structure of manipulation.

### D.4 Robustness to Environment Stochasticity

We examine a modified FetchPush environment with additive Gaussian action noise (standard deviation in $\{0.2, 0.5, 1.0, 1.5\}$), following Ma et al. (2022).

As shown in Figure 10, GCHR is the most robust to stochasticity and matches or outperforms the baselines at every noise level. WGCSL is barely sensitive to noise but performs worse overall. A plausible explanation is that its advantage-weighted regression is conservative and does not fully exploit the Q-function when selecting actions. DDPG+HER is more sensitive, possibly because its deterministic policy cannot adapt to the stochastic transitions induced by action perturbations. We conjecture two reasons for the robustness of GCHR. The goal prior provides diverse candidate actions that hedge against transition noise, and the RL objective in Equation (10) prefers actions that keep high value under the perturbed dynamics.

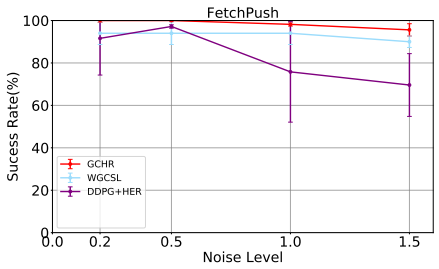



Figure 10: Mean success rate (%) for FetchPush under environment stochasticity.

### D.5 The Impact of Hyperparameter $\beta$ and $\alpha$

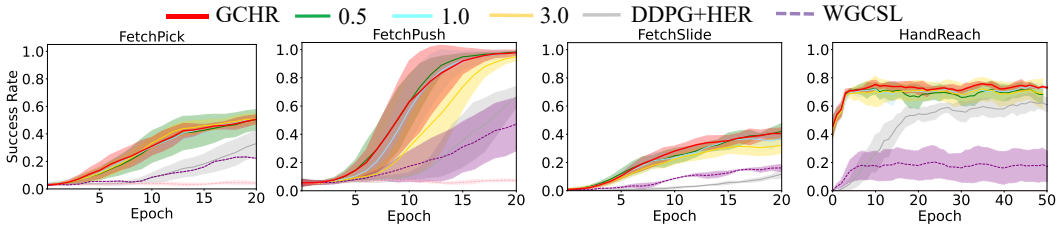



Figure 11: Hyperparameter $\beta$ ablation studies. GCHR outperforms the baselines for every value of $\beta$.

We evaluate $\beta \in \{0.2, 0.5, 1.0, 3.0\}$ (Figure 11). GCHR outperforms the baselines for every $\beta$, which indicates robustness to this choice.

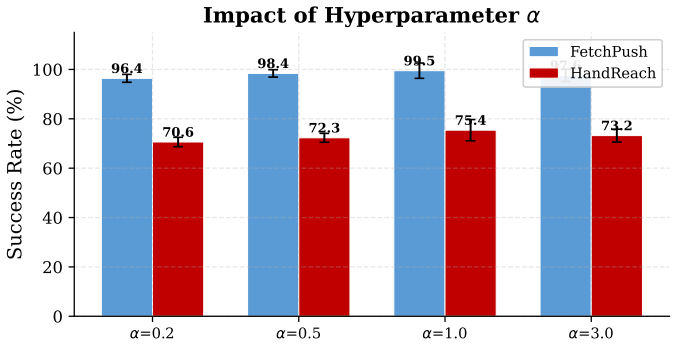



Figure 12: Impact of hyperparameter $\alpha$ on FetchPush and HandReach. Performance peaks at $\alpha = 1.0$.

Figure 12 shows that performance increases for $\alpha \leq 1$ and decreases for $\alpha > 1$, so we use $\alpha = 1.0$ by default. Across the combined $\alpha$ and $\beta$ sweeps, the ratio $\alpha/\beta$ ranges from $1/3$ to 15 without a large drop in success, which indicates robustness to the balance of the two regularizers.

### D.6 The Impact of the Waypoint Pool

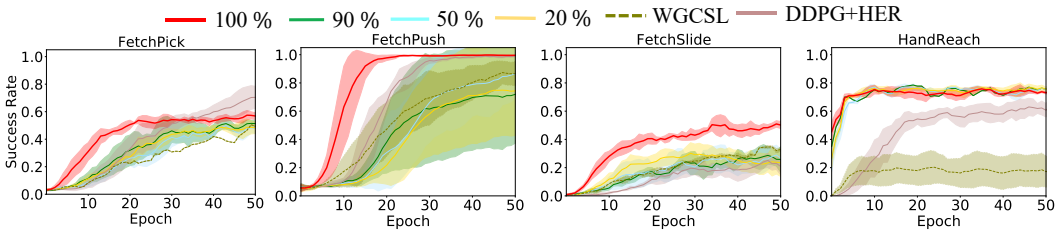



Figure 13: Waypoint pool ablation. Percentages give the fraction of the goals on the trajectory from which the $K = 10$ waypoints are drawn; 100% is the default used in Figure 2. With 20–90%, GCHR learns more slowly than DDPG+HER on FetchPush.

We draw the $K = 10$ waypoints from only a fraction $\{20\%, 50\%, 90\%, 100\%\}$ of the goals on the trajectory (Figure 13). 100% is the default used in Figure 2. Drawing from the whole trajectory learns fastest on every task. Smaller pools still match or exceed the baselines on FetchPick, FetchSlide, and HandReach, but on FetchPush they learn more slowly than DDPG+HER. Appendix D.10 studies the count $K$ itself.

### D.7 Sample Efficiency

Figure 14 shows that GCHR needs about 9,000 samples to reach a 0.45 success rate on FetchPush, whereas DDPG+HER needs about 17,000, i.e., about 1.9× fewer samples at this threshold; Appendix D.10 reports 1.8× under its convergence criterion. On BlockRotateZ, GCHR needs the fewest samples of all methods to reach a 0.5 success rate, so the advantage extends to high-dimensional manipulation where exploration is the main bottleneck. Both results are consistent with a warm start: the two priors provide informative gradients as soon as the replay buffer holds relabeled trajectories, which reduces the interactions needed to discover successful behaviours.

### D.8 Relabeling Ratio

We evaluate relabeling ratios $\{0.2, 0.5, 0.8, 1.0\}$, one ratio per task (Figure 15). GCHR matches or outperforms the baselines at every ratio. Because each ratio appears on a different task, this study compares methods at a fixed ratio but does not isolate the effect of the ratio itself. Absolute success

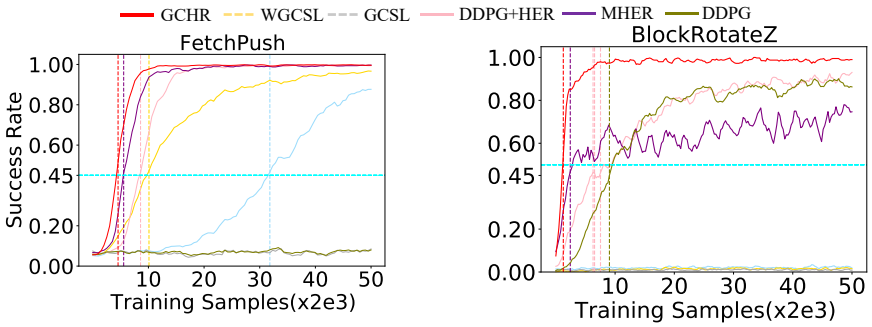



Figure 14: Mean success rate against training samples. Vertical dashed lines mark the samples each method needs to reach the threshold shown by the horizontal line; fewer samples is better.

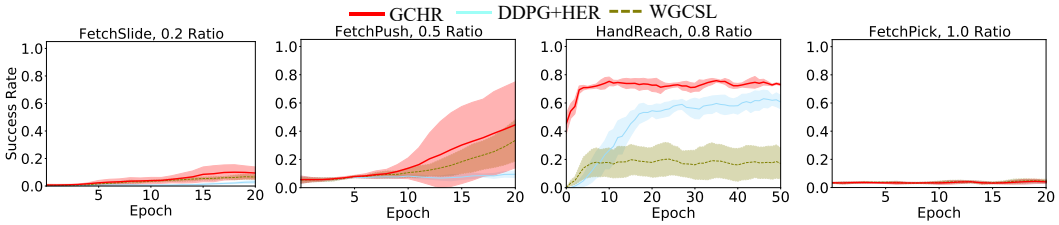



Figure 15: Relabeling ratio ablation. Each panel uses a different task and ratio. GCHR matches or outperforms the baselines at every tested ratio.

does depend on the ratio: GCHR reaches lower success on FetchPush at 0.5 than at the default 0.8 (Figure 2), and all methods fail on FetchPick at 1.0. The goal prior draws waypoints from the stored trajectory independently of HER relabeling, so it still receives trajectory structure at low ratios.

### D.9 Learning Curves Under Goal Corruption

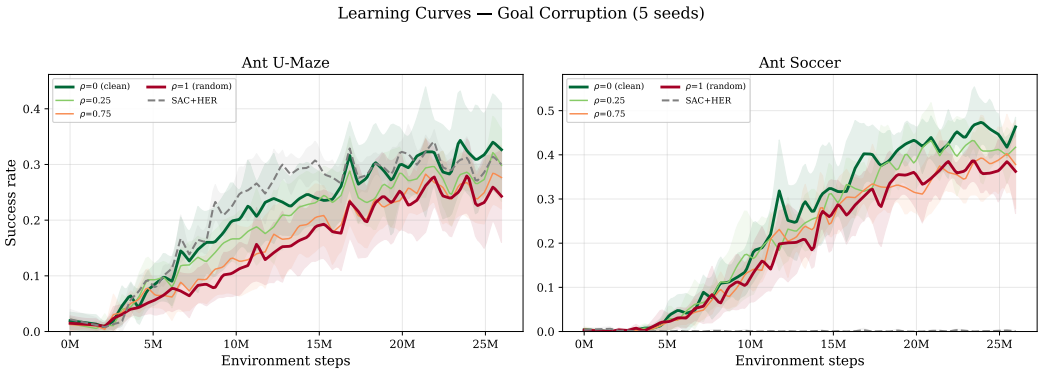



Figure 16: Training curves for the goal-corruption experiment. Clean GCHR ($\eta_c = 0$) learns faster and reaches a higher final mean success than fully corrupted GCHR ($\eta_c = 1$) in both environments. The legend uses $\rho$ for $\eta_c$.

Figure 16 complements Figure 6 with learning dynamics under different corruption levels. Clean GCHR ($\eta_c = 0$) learns faster and reaches a higher final mean success than fully corrupted GCHR ($\eta_c = 1$). The gap is largest in the first 10M steps, when trajectory-aligned waypoints give the strongest signal because the policy has not yet converged. The gap narrows later as the RL objective compensates for poor prior quality, and a smaller difference in mean success remains at 25M steps.

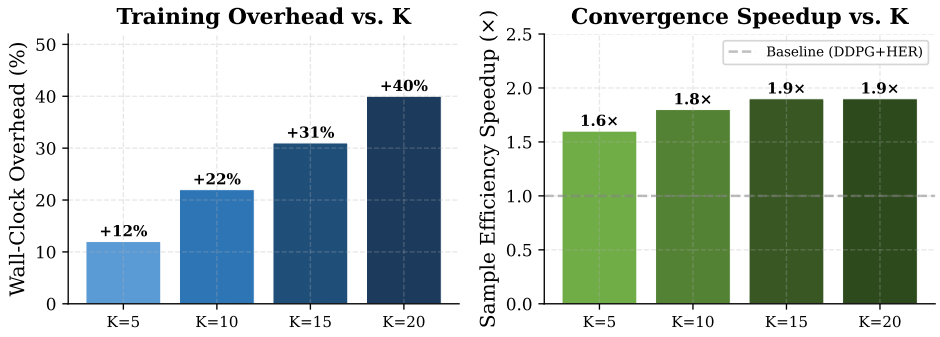



Figure 17: Wall-clock overhead and sample efficiency vs. DDPG+HER on FetchPush. **Left:** overhead scales linearly with $K$. **Right:** the convergence speedup more than compensates and saturates from $K = 15$ on.

### D.10 Wall-Clock Overhead Analysis

Figure 17 reports the computational overhead as a function of $K$. Computing $\pi_{\text{HG}}$ requires $K$ forward passes of the target actor per sample, so the overhead grows linearly with $K$. The sample-efficiency speedup is the ratio of the samples that DDPG+HER and GCHR need to converge on FetchPush. With $K = 5$, the overhead is only 12% and convergence is already 1.6× faster. The default $K = 10$ balances a 22% overhead with a 1.8× speedup. From $K = 15$ on, the speedup saturates at 1.9×, while the overhead grows to 31% for $K = 15$ and 40% for $K = 20$.

### D.11 Q-Landscape Visualization

The 2D action space of Reacher allows a direct view of coverage expansion. Figure 18 shows the Q-landscape $Q(s, \cdot, g)$ over the action space for three illustrative $(s, g)$ pairs. The heatmap encodes Q-values (blue = high, red = low). Red × markers are actions from the behavioural support $\rho_{\text{beh}}$. Blue ∘ markers are novel actions from $\pi_{\text{HG}}$ outside this support, and the green ⋆ is the current policy's action. In all three panels, the novel $\pi_{\text{HG}}$ actions lie in high-Q regions beyond the convex hull of the $\rho_{\text{beh}}$ actions. Proposition 4.1 guarantees only that $\pi_{\text{HG}}$ keeps mass outside the behavioural support; that this mass falls on high-Q actions is an empirical observation for these pairs. The pairs are chosen for illustration, and their advantages (+3.2 to +3.8) are far above the pooled advantage of novel actions on Reacher (+0.25, Section 6.6). As training progresses from left to right, the novel fraction decreases from 47% to 11% as the behavioural support expands.

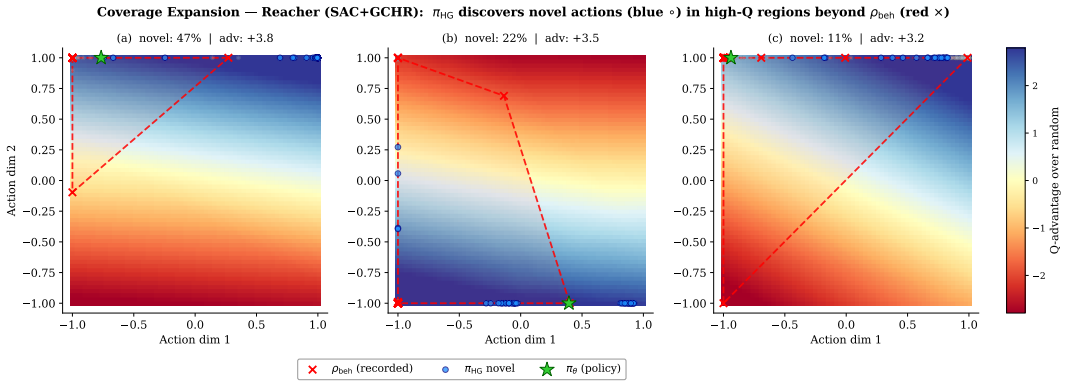



Figure 18: Q-landscape on Reacher. Background: $Q(s, \cdot, g)$ over the 2D action space (blue = high Q, red = low Q). Red ×: $\rho_{\text{beh}}$ actions. Blue ∘: novel $\pi_{\text{HG}}$ actions. Green ⋆: current policy action. For these illustrative pairs, novel actions appear in high-Q regions beyond the behavioural support, which goes beyond the mass guarantee of Proposition 4.1.

# E Implementation Details

## E.1 Algorithm and Architecture

We use an off-policy actor-critic algorithm with HER (Andrychowicz et al., 2017) as the base GCRL framework. The critic is trained by temporal difference learning, and the target networks use soft updates. DDPG and TD3 already maintain a target actor. For SAC and CRL, we add a target actor as an exponential moving average of $\pi_\theta$, which requires no gradient updates. On the Gym suite, we follow the parameter settings of Liu et al. (2023), and all baselines share the same hyperparameters. Our implementations of the baselines and GCHR build on the following repository:

- DDPG, DDPG+HER, MHER, GCSL, WGCSL: https://github.com/Cranial-XIX/metric-residual-network.

**Baseline Descriptions.** We compare GCHR with state-of-the-art goal-conditioned algorithms. Symbols in the following descriptions are local to each baseline. $\mathcal{B}$ denotes the replay buffer with original goals, and $\mathcal{B}_r$ denotes its hindsight-relabeled counterpart.

**DDPG** (Lillicrap et al., 2015): We adopt an open-source implementation pre-optimized for multi-goal tasks. The critic minimizes $\mathbb{E}_{(s_t,a_t,s_{t+1},g)\sim\mathcal{B}}[(r(s_t,g)+\gamma Q(s_{t+1},\pi(s_{t+1},g),g)-Q(s_t,a_t,g))^2]$. The policy maximizes $\mathbb{E}_{(s_t,g)\sim\mathcal{B}}[Q(s_t,\pi(s_t,g),g)]$.

**DDPG+HER** (Andrychowicz et al., 2017): Goal-conditioned DDPG augmented with Hindsight Experience Replay. For each stored transition $(s_t, a_t, s_{t+1}, g)$, the relabeled goal is drawn under the future strategy as $g' = \phi(s_{t'})$ with $t' \sim \mathrm{Uniform}(\{t+1,\dots,T\})$, and the hindsight reward $r(s_t, g')$ is recomputed under the sparse goal-reaching reward. The critic is then trained on the mixed buffer $\mathcal{B}\cup\mathcal{B}_r$ by minimizing $\mathbb{E}_{(s_t,a_t,s_{t+1},g')\sim\mathcal{B}\cup\mathcal{B}_r}\left[(r(s_t,g')+\gamma Q(s_{t+1},\pi(s_{t+1},g'),g')-Q(s_t,a_t,g'))^2\right]$, with a relabeling ratio of 0.8. Thus 80% of replay samples use $g'$, and the remaining 20% retain the original $g$. This relabeling turns every trajectory into a successful demonstration for its achieved goals.

**MHER** (Yang et al., 2021): Model-based Hindsight Experience Replay. MHER learns a dynamics model $f_\omega(s,a) \approx s'$ by minimizing $\mathbb{E}_{(s,a,s')\sim\mathcal{B}}[\|f_\omega(s,a)-s'\|^2]$ and rolls the current policy $\pi$ forward for $n$ steps under $f_\omega$ from each sampled state $s_t$. This rollout produces an imagined future state $\tilde{s}_{t+n}$. The virtual relabeled goal is $\tilde{g} = \phi(\tilde{s}_{t+n})$, with the sparse goal-reaching reward $r(s_t, \tilde{g})$ recomputed accordingly. The critic is then trained on these model-generated hindsight transitions by minimizing $\mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{B},\,\tilde{g}}\left[(r(s_t,\tilde{g})+\gamma Q(s_{t+1},\pi(s_{t+1},\tilde{g}),\tilde{g})-Q(s_t,a_t,\tilde{g}))^2\right]$. Goals from on-policy model rollouts are consistent with the current policy, at the cost of compounding errors in the learned dynamics model.

**GCSL** (Ghosh et al., 2021): Implemented by removing the critic and maximizing the likelihood of relabeled actions, that is, by minimizing $-\mathbb{E}_{(s,a,g)\sim\mathcal{B}_r}[\log\pi(a \mid s,g)]$.

**WGCSL** (Yang et al., 2022): Extension of GCSL with weighted supervised learning. WGCSL weights the GCSL log-likelihood by a discounted relabeling weight, a clipped exponential advantage weight, and a best-advantage weight. The advantage is estimated as $A(s_t,a_t,g) = r(s_t,g) + \gamma Q(s_{t+1},\pi(s_{t+1},g),g) - Q(s_t,a_t,g)$. The original paper states that WGCSL also applies to the online setting.

**RIS** (Chane-Sane et al., 2021): Goal-conditioned RL with Imagined Subgoals. RIS trains a high-level subgoal predictor $\pi_\psi^H(s_g \mid s,g)$ through advantage-weighted regression. Its subgoal cost $C_\pi(s_g \mid s,g) = \max(|V^\pi(s,s_g)|, |V^\pi(s_g,g)|)$ favours waypoints roughly halfway between $s$ and $g$. The low-level actor is trained under a SAC-style objective regularized by a KL term toward a non-parametric prior that marginalizes over imagined subgoals from the target policy, $\pi^{\mathrm{prior}}(a \mid s,g) = \mathbb{E}_{s_g\sim\pi^H(\cdot|s,g)}[\pi_{\bar{\theta}}(a \mid s,s_g)]$. The actor maximizes $\mathbb{E}_{(s,g)\sim\mathcal{B},\,a\sim\pi_\theta}\left[Q(s,a,g) - \alpha\, D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s,g)\,\|\,\pi^{\mathrm{prior}}(\cdot \mid s,g))\right]$, with the KL estimated by Monte-Carlo samples over imagined subgoals. RIS thus trains an additional subgoal prediction network and draws imagined subgoals from it. GCHR aggregates over waypoints of the stored trajectory.

**CRL** (Eysenbach et al., 2022b): Contrastive Reinforcement Learning casts goal reaching as a binary classification problem over future states. The critic is an inner-product energy $f(s,a,g) = \phi(s,a)^\top\psi(g)$. Positive goals $g^+$ are drawn from the discounted future-state occupancy along the same

trajectory, and negative goals $g^-$ from the marginal state distribution. The critic is trained by the NCE-binary objective, which maximizes $\mathbb{E}_{(s,a),\, g^+,\, g^-}[\log \sigma(f(s,a,g^+)) + \log(1-\sigma(f(s,a,g^-)))]$. Its Bayes-optimal solution satisfies $\exp(f^*(s,a,g)) \propto Q^\pi(s,a,g)/p(g)$, so the learned energy recovers the goal-conditioned Q-function up to a goal-marginal factor. The actor maximizes this energy directly, $\mathbb{E}_{(s,g)\sim\mathcal{B},\, a\sim\pi(\cdot|s,g)}[f(s,a,g)]$.

**SAC** (Haarnoja et al., 2018): Soft Actor-Critic is an off-policy maximum-entropy algorithm that trains a stochastic policy by jointly maximizing expected return and policy entropy. With a twin critic and automatic temperature tuning, the target value includes an entropy correction term, $y_t = r(s_t,g) + \gamma\, \mathbb{E}_{a_{t+1}\sim\pi_\phi(\cdot|s_{t+1},g)}\big[\min_{i=1,2} Q_{\bar\theta_i}(s_{t+1},a_{t+1},g) - \alpha \log \pi_\phi(a_{t+1} \mid s_{t+1},g)\big]$. The actor is reparameterized as $a = f_\phi(s,g,\epsilon)$ with $\epsilon \sim \mathcal{N}(0,I)$ and minimizes $\mathbb{E}_{(s,g)\sim\mathcal{B},\,\epsilon}[\alpha \log \pi_\phi(f_\phi(s,g,\epsilon) \mid s,g) - \min_{i=1,2} Q_{\theta_i}(s, f_\phi(s,g,\epsilon), g)]$. The temperature is learned by minimizing $\mathbb{E}_{a\sim\pi}[-\alpha \log \pi(a \mid s,g) - \alpha\bar{\mathcal{H}}]$ with entropy target $\bar{\mathcal{H}} = -\dim(\mathcal{A})$. We use SAC with HER relabeling (SAC+HER) as a baseline, and SAC+GCHR augments the SAC actor loss with our regularization.

**TD3** (Fujimoto et al., 2018): Twin Delayed DDPG reduces function-approximation error in actor-critic methods through clipped double-Q learning, delayed policy updates, and target policy smoothing. Both critics regress by mean squared error to the shared target $y_t = r(s_t,g) + \gamma \min_{i=1,2} Q_{\bar\theta_i}\big(s_{t+1},\, \pi_{\bar\phi}(s_{t+1},g) + \epsilon,\, g\big)$ with $\epsilon \sim \mathrm{clip}(\mathcal{N}(0,\sigma), -c, c)$, $\sigma = 0.2$, and $c = 0.5$. The actor follows the deterministic policy gradient using only $Q_{\theta_1}$, i.e. $\nabla_\phi J(\phi) \propto \mathbb{E}_{s,g}[\nabla_a Q_{\theta_1}(s,a,g)|_{a=\pi_\phi(s,g)}\, \nabla_\phi \pi_\phi(s,g)]$. It is updated once every $d = 2$ critic steps together with Polyak target updates. We use TD3 with HER relabeling (TD3+HER) as a baseline.

**QRL** (Wang et al., 2023): Optimal Goal-Reaching RL via Quasimetric Learning. QRL parameterizes the negative goal-conditioned value as a quasimetric $d_\theta(s,g)$ that satisfies the triangle inequality by construction through Interval Quasimetric Embeddings. It recovers $V^*(s,g) = -d_\theta(s,g)$ by *maximizing* $d_\theta$ subject to a local one-step consistency constraint, $\max_\theta \mathbb{E}_{s,g}[d_\theta(s,g)]$ s.t. $\mathbb{E}_{(s,a,s',r)}[\mathrm{relu}(d_\theta(s,s') + r)^2] \le \epsilon^2$, which is solved in its Lagrangian form with a learnable multiplier. A latent one-step transition $T(f(s),a)$ is trained alongside the quasimetric, and the policy is extracted by minimizing the predicted remaining distance $\mathbb{E}_{s,g,\, a\sim\pi(\cdot|s,g)}[d^z(T(f(s),a), f(g))]$. We compare against QRL on image-based tasks.

**TD-InfoNCE** (Zheng et al., 2024b): Contrastive Difference Predictive Coding replaces the Monte-Carlo InfoNCE positives of CRL with a temporal-difference bootstrap. The critic $f(s,a,g) = \phi(s,a)^\top \psi(g)$ is trained with a categorical softmax loss. Its positive is a mixture of a one-step next state (weight $1-\gamma$) and a bootstrapped future state (weight $\gamma$) reweighted by an importance ratio $w$ from the target critic, $\mathcal{L}_{\text{TD-InfoNCE}} = -\mathbb{E}\left[(1-\gamma)\log \frac{e^{f(s,a,g^+_{\text{1step}})}}{\sum_i e^{f(s,a,g^{(i)})}} + \gamma\, \lfloor w \rfloor_{\text{sg}} \log \frac{e^{f(s,a,g^+_{\text{boot}})}}{\sum_i e^{f(s,a,g^{(i)})}}\right]$. Here $w(s',a',\{g^{(i)}\}) = e^{\bar f(s',a',g^+)}/[\frac{1}{N}\sum_i e^{\bar f(s',a',g^{(i)})}]$ is evaluated at $a' \sim \pi(\cdot \mid s')$ under the target network, and $\lfloor\cdot\rfloor_{\text{sg}}$ denotes stop-gradient. This construction defines a $\gamma$-contractive InfoNCE Bellman operator, which enables fully off-policy contrastive learning and improves stability in pixel-based goal reaching.

### E.2 Evaluation Setup and Hyperparameters

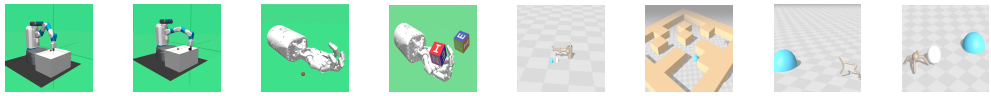
For each baseline and task, we run five random seeds ($\{100, 200, 300, 400, 500\}$). On the Gym suite, training follows Table 3, with 50 cycles per epoch and 40 gradient batches per cycle. Fetch tasks run for 20 epochs and Hand tasks for 50 epochs; Figure 13 runs all tasks for 50 epochs, and Figure 5 shows the first 10 epochs. At the end of each epoch, we compute the mean success rate over 100 independent rollouts with randomly sampled goals. The discount factor on the Gym suite is $\gamma = 0.98$. On JaxGCRL, each backbone keeps the hyperparameters of its JaxGCRL implementation, and the visual tasks keep those of the TD-InfoNCE implementation. GCHR adds only $\alpha$, $\beta$, $K$, and, for SAC and CRL, the target-actor rate $\tau_{\text{soft}}$. Results are averaged across seeds, and shaded regions show the standard deviation. We use the Adam optimizer (Kingma & Ba, 2014) in all setups.

| Hyperparameter | Value | Hyperparameter | Value |
|---|---|---|---|
| Learning rate | 1e-3 | Buffer size | $10^6$ |
| $\tau_{\text{soft}}$ (target update) | 0.05 | Batch size | 256 |
| MPI workers | 16 | Epochs | 20 (Fetch), 50 (Hand) |
| Cycles/epoch | 50 | Batches/cycle | 40 |
| Random action prob. | 0.3 | Gaussian noise | 0.2 |
| HER replay prob. | 0.8 | $\alpha$ (beh. weight) | 1.0 |
| $\beta$ (goal weight) | 0.2 | $K$ (waypoints) | 10 |
| $\sigma_{\text{fixed}}$ (det. policy) | 0.2 | $\gamma$ (discount) | 0.98 |

Table 3: Hyperparameters for GCHR and baselines on the Gym robotics suite.

### E.3 Environment Details

All goal-conditioned tasks come from OpenAI Gym robotics (Plappert et al., 2018), JaxGCRL (Bortkiewicz et al., 2025), and OGBench (Park et al., 2025). Figure 19 shows representative environments.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 19: Benchmark environments. (a) FetchReach, (b) FetchPush, (c) HandReach, (d) HandManipulateBlock, (e) Pusher Hard, (f) Ant U-Maze, (g) Cheetah, (h) Ant Soccer.

**OpenAI Gym Robotic Tasks.** The **Fetch** robotic manipulation suite (**FetchReach, FetchPush, FetchSlide, FetchPick**) evaluates 7-DoF arm control on four goal-oriented tasks. These benchmarks share high-dimensional state vectors (joint positions and velocities) and 4D action spaces (end-effector control plus gripper operation). Success requires reaching spatial targets within a tolerance of $\epsilon = 0.05$. The tasks differ in their manipulation requirements, from direct end-effector positioning (FetchReach) to indirect object interaction through pushing (FetchPush), sliding beyond workspace limits (FetchSlide), and precise aerial retrieval (FetchPick). The Gym environments adopt a $\{-1, 0\}$ reward convention, assigning $r = 0$ when $\|\phi(s) - g_{XYZ}\|_2 \leq \epsilon$ and $r = -1$ otherwise. Relative to the $\{0, 1\}$ convention of our analysis, this shifts Q-values by the constant $-1/(1-\gamma)$ and leaves the optimal policy unchanged. The **Hand** manipulation benchmarks (**HandReach, BlockRotateZ, BlockRotateXYZ, BlockRotateParallel**) require precise coordination of a 20-DoF Shadow Hand for dexterous object manipulation. These high-dimensional control tasks use multimodal state observations (joint kinematics combined with object dynamics) and sparse binary rewards. HandReach succeeds when the fingertip positions lie within 0.01 of their targets, and the block tasks succeed when the block orientation lies within 0.1 rad of the target orientation. Task complexity grows with the rotational demands, from rotation about a single axis (RotateZ) to rotation about all three axes (RotateXYZ). RotateParallel combines a random rotation about the $z$ axis with axis-aligned rotations about the $x$ and $y$ axes. These tasks test GCRL in high-DoF settings where Assumption 4.2 holds only approximately.

For image-based tasks, we build our Fetch-series experiments on the official implementation released with TD-InfoNCE (Zheng et al., 2024b). To test the method under conditions closer to real-world scenarios, we also integrate the manipulation environments from OGBench (Park et al., 2025) and adapt our pipeline to the online setting.

**JaxGCRL Tasks.** The JaxGCRL benchmark (Bortkiewicz et al., 2025) provides GPU-accelerated environments implemented in JAX, which allow millions of environment steps of training within minutes on a single GPU. We evaluate on five tasks spanning manipulation, locomotion, and navigation. **Pusher Hard** is a 3D robotic manipulation task where a robotic arm must push a circular object to a randomly sampled goal position. Both the object position and goal are randomized at each episode reset, which requires precise contact-based control. **Ant U-Maze** and **Ant Big Maze** use a

quadruped robot (Ant) that must navigate through U-shaped and large maze structures, respectively, to reach target positions sampled from designated goal regions. These tasks require long-horizon planning and exploration to find paths through the maze corridors. **Cheetah** is a 2D locomotion task where a two-legged agent must reach a goal sampled from one of two possible positions, one on each side of the starting position. This task primarily tests locomotion speed and directional control. **Ant Soccer** requires the Ant quadruped to push a spherical ball into a goal position sampled uniformly from a circle around the starting position. The ball position is randomized on the line between the agent and the goal. This task combines locomotion with object manipulation and demands both body control and contact planning. All JaxGCRL tasks use sparse binary rewards and are evaluated for 50M environment steps with 5 random seeds. The goal-corruption study in Section 6.5 uses a 25M-step budget.