Offline Goal-Conditioned Reinforcement Learning via f-Advantage Regression

Yecheng Jason Ma, Jason Yan, Dinesh Jayaraman, Osbert Bastani
University of Pennsylvania
{jasonyma, jasyan, dineshj, obastani}@seas.upenn.edu
https://jasonma2016.github.io/GoFAR/

Abstract

Offline goal-conditioned reinforcement learning (GCRL) promises general-purpose skill learning in the form of reaching diverse goals from purely offline datasets. We propose **Go**al-conditioned f-**A**dvantage **R**egression (GoFAR), a novel regression-based offline GCRL algorithm derived from a state-occupancy matching perspective; the key intuition is that the goal-reaching task can be formulated as a state-occupancy matching problem between a dynamics-abiding imitator agent and an expert agent that directly teleports to the goal. In contrast to prior approaches, GoFAR does not require any hindsight relabeling and enjoys uninterleaved optimization for its value and policy networks. These distinct features confer GoFAR with much better offline performance and stability as well as statistical performance guarantee that is unattainable for prior methods. Furthermore, we demonstrate that GoFAR's training objectives can be re-purposed to learn an agent-independent goal-conditioned planner from purely offline source-domain data, which enables zero-shot transfer to new target domains. Through extensive experiments, we validate GoFAR's effectiveness in various problem settings and tasks, significantly outperforming prior state-of-art. Notably, on a real robotic dexterous manipulation task, while no other method makes meaningful progress, GoFAR acquires complex manipulation behavior that successfully accomplishes diverse goals.

1 Introduction

Goal-conditioned reinforcement learning [18, 43, 39] (GCRL) aims to learn a repertoire of skills in the form of reaching distinct goals. *Offline* GCRL [6, 47] is particularly promising because it enables learning general goal-reaching policies from purely offline interaction datasets without any environment interaction [28, 25], which can be expensive in the real-world. As offline datasets contain diverse goals and become increasingly prevalent [9, 19, 6], policies learned this way can acquire a large set of useful primitives for downstream tasks [30]. A central challenge in GCRL is the sparsity of reward signal [3]; without any additional knowledge about the environment, an agent at a state typically only accrues positive binary reward when the state lies within the goal neighborhood. This sparse reward problem is exacerbated in the offline setting, in which the agent cannot explore the environment to discover more informative states about desired goals. Therefore, designing an effective offline GCRL algorithm is a concrete yet challenging path towards general-purpose and scalable policy learning.

In this paper, we present a novel offline GCRL algorithm, **Go**al-conditioned **f**-**A**dvantage **R**egression (GoFAR), first casting GCRL as a state-occupancy matching [27, 32] problem and then deriving a regression-based policy objective. In particular, GoFAR begins with the following goal-conditioned state-matching objective:

$$\min_{\pi} \mathrm{D}_{\mathrm{KL}}(d^{\pi}(s;g) \| p(s;g)) \tag{1}$$

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

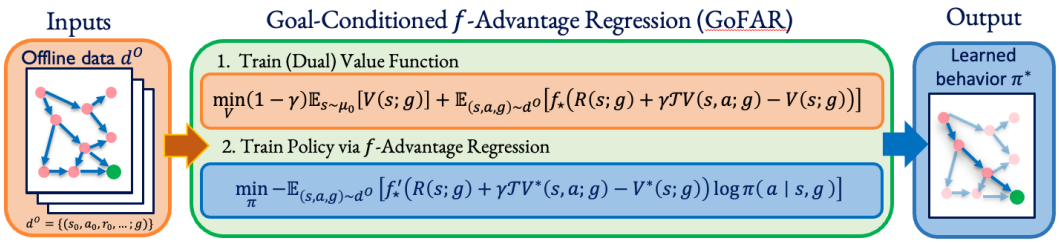


Figure 1: GoFAR schematic illustration.

where $d^\pi(s;g)$ is the goal-conditioned state-occupancy distribution of policy π and $p(s;g)$ is the distribution of states that satisfy a particular goal g. This objective casts goal-conditioned offline RL as an imitation learning problem: a dynamics-abiding agent imitates as well as possible an expert who can teleport to the goal in one step; see Figure 2. Posing GCRL this way is mathematically principled, as we show that this objective is equivalent to a probabilistic interpretation of GCRL that additionally encourages maximizing state entropy. More importantly, this objectives enables us to extend a state-of-art offline imitation learning algorithm [32] to the goal-conditioned setting and admits elegant optimization using purely offline data by considering an f-divergence regularized objective and exploiting ideas from convex duality theory [41, 4]. In particular, we obtain the dual optimal value function V^* from a single unconstrained optimization problem, using which we construct the optimal advantage function (we refer to this as *f-advantage*) that serves as importance weighting for a regression-based policy training objective; see Figure 1 for a schematic illustration.

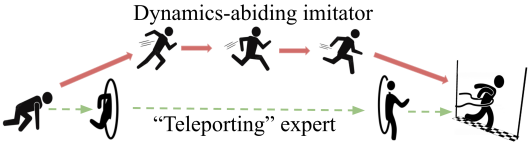


Figure 2: GCRL can be thought of imitating an expert agent that can teleport to goals.

There are several distinct features to our approach. First and foremost is GoFAR's lack of goal relabeling. Hindsight goal relabeling [3] (known as HER in the literature) relabels trajectory goals to be states that were actually achieved instead of the originally commanded goals. This heuristic is critical for alleviating the sparse-reward problem in prior GCRL methods, but is unnecessary for GoFAR. GoFAR updates its policy as if all data is already coming from the *optimal* goal-conditioned policy and therefore does not need to perform any hindsight goal relabeling. GoFAR's *relabeling-free* training is of significant practical benefits. First, it enables more stable and simpler training by avoiding sensitive hyperparameter tuning associated with HER that cannot be easily performed offline [48]. Second, hindsight relabeling suffers from hindsight bias in stochastic environments [44], as achieved goals may be accomplished due to noise; this is exacerbated in the offline setting due to inability to collect more data. By bypassing hindsight relabeling, GoFAR promises to be more robust and scalable.

Second, as suggested in Figure 1, GoFAR's value and policy training steps are completely *uninterleaved*: we do not need to update the policy until the value function has converged. This confers greater offline training stability [23, 32] compared to prior works, which mostly involve alternating updates to a critic Q-network and policy network. Furthermore, it enables an algorithmic reduction of GoFAR to weighted regression [7], which allows us to obtain strong finite-sample statistical guarantee on GoFAR's performance; prior regression-based GCRL approaches [14, 47] do not enjoy this reduction and obtain much weaker theoretical guarantees.

Finally, we show that GoFAR can also be used to learn a goal-conditioned *planner* [36, 5]. The key insight lies in observing that, under a mild assumption, GoFAR's dual value function objective does not depend on the action and can learn from state-only offline data. This enables learning a *goal-centric* value function and thereby a near-optimal goal-conditioned planner that is capable of zero-shot transferring to new domains of the same task that shares the goal space. In our experiments, we illustrate that GoFAR planner can indeed plan effective subgoals in a new target domain and enable a low-level controller to reach distant goals that it is not designed for.

We extensively evaluate GoFAR on a variety of offline GCRL environments of varying task complexity and dataset composition, and show that it outperforms all baselines in all settings. Notably, GoFAR is more robust to stochastic environments than methods that depend on hindsight relabeling. We additionally demonstrate that GoFAR learns complex manipulation behavior on a real-world high-dimensional dexterous manipulation task [2], for which most baselines fail to make any progress.

Finally, we showcase GoFAR's planning capability in a cross-robot zero-shot transfer task. To summarize, our contributions are:

1. GoFAR, a novel offline GCRL algorithm derived from goal-conditioned state-matching
2. Detailed technical derivation and analysis of GoFAR's distinct features: (1) relabeling-free, (2) uninterleaved optimization, and (3) planning capability
3. Extensive experimental evaluation of GoFAR, validating its empirical gains and capabilities

2 Related Work

Offline Goal-Conditioned Reinforcement Learning. One core challenge of (offline) GCRL [18, 43] is the sparse-reward nature of goal-reaching tasks. A popular strategy is hindsight goal relabeling [3] (or HER), which relabels off-policy transitions with future goals they have achieved instead of the original commanded goals. Existing offline GCRL algorithms [6, 47] adapt HER-based online GCRL algorithms to the offline setting by incorporating additional components conducive to offline training. [6] builds on actor-critic style GCRL algorithms [3] by adding conservative Q-learning [24] as well as goal-chaining, which expands the pool of candidate relabeling goals to the entire offline dataset.

Besides actor-critic methods, goal-conditioned behavior cloning, coupled with HER, has been shown to be a simple and effective method [14]. [47] improves upon [14] by incorporating discount-factor and advantage function weighting [37, 38], and shows improved performance in offline GCRL. Diverging from existing literature, GoFAR does not tackle offline GCRL by adapting an online algorithm; instead, it approaches offline GCRL from a novel perspective of state-occupancy matching and derives an elegant algorithm from the dual formulation that is naturally suited for the offline setting and carries many distinct properties that make it more stable, scalable, and versatile.

State-Occupancy Matching. State occupancy matching objectives have been shown to be effective for learning from observations [32, 49], exploration [27], as well as matching a hand-designed target state distribution [13]; occupancy matching, in general, has been explored in the imitation learning literature [17, 45, 13, 20]. Our strategy of estimating the state-occupancy ratio is related to DICE techniques for offline policy optimization [33, 35, 26, 21, 32]. The closest work to ours is [32], which proposed f-divergence state-occupancy matching for offline imitation learning. Our work shows that such a state-occupancy matching approach can be applied to GCRL. Algorithmically, GoFAR differs from [32] in that it does not require a separate dataset of expert demonstrations; it achieves this by modifying the discriminator, and in some settings can even be used without a discriminator, whereas a discriminator is always required by [32]. Furthermore, we derive several new algorithmic properties (Section 4.2-4.4) that are novel and particularly advantageous for offline GCRL.

3 Problem Formulation

Goal-Conditioned Reinforcement Learning. We consider an infinite-horizon Markov decision process (MDP) [40] $\mathcal{M} = (S, A, R, T, \mu_0, \gamma)$ with state space S, action space A, deterministic rewards $r(s, a)$, stochastic transitions $s' \sim T(s, a)$, initial state distribution $\mu_0(s)$, and discount factor $\gamma \in (0, 1]$. A policy $\pi : S \to \Delta(A)$ outputs a distribution over actions to use in a given state. In goal-conditioned RL, the MDP additionally assumes a goal space $G \coloneqq \{\phi(s) \mid s \in S\}$, where the state-to-goal mapping $\phi : S \to G$ is known. Now, the reward function $r(s; g)$[1] as well as the policy $\pi(a \mid s, g)$ depend on the commanded goal $g \in G$. Given a distribution over desired goals $p(g)$, the objective of goal-conditioned RL is to find a policy π that maximizes the discounted return:

$$J(\pi) \coloneqq \mathbb{E}_{g\sim p(g), s_0\sim\mu_0, a_t\sim\pi(\cdot|s_t,g), s_{t+1}\sim T(\cdot|s_t,a_t)} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t; g)\right] \tag{2}$$

The *goal-conditioned* state-action occupancy distribution $d^\pi(s, a; g) : S \times A \times G \to [0, 1]$ of π is

$$d^\pi(s, a; g) \coloneqq (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s, a_t = a \mid s_0 \sim \mu_0, a_t \sim \pi(s_t; g), s_{t+1} \sim T(s_t, a_t)) \tag{3}$$

which captures the relative frequency of state-action visitations for a policy π conditioned on goal g. The state-occupancy distribution then marginalizes over actions: $d^\pi(s; g) = \sum_a d^\pi(s, a; g)$. Then, it

[1] In GCRL, the reward function customarily does not depend on action.

follows that $\pi(a \mid s, g) = \frac{d^\pi(s,a;g)}{d^\pi(s;g)}$. A state-action occupancy distribution must satisfy the *Bellman flow constraint* in order for it to be an occupancy distribution for some stationary policy π:

$$\sum_a d(s,a;g) = (1-\gamma)\mu_0(s) + \gamma \sum_{\tilde{s},\tilde{a}} T(s \mid \tilde{s},\tilde{a}) d(\tilde{s},\tilde{a};g), \qquad \forall s \in S, g \in G \tag{4}$$

We write $d^\pi(s,g) = p(g)d^\pi(s;g)$ as the joint goal-state density induced by $p(g)$ and the policy π. Finally, given d^π, we can express the objective function (2) as $J(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{(s,g)\sim d^\pi(s,g)}[r(s;g)]$.

Offline GCRL. In offline GCRL, the agent cannot interact with the environment $\mathcal{M}$; instead, it is equipped with a static dataset of logged transitions $\mathcal{D} := \{\tau_i\}_{i=1}^N$, where each trajectory $\tau^{(i)} = (s_0^{(i)}, a_0^{(i)}, r_0^{(i)}, s_1^{(i)}, ...; g^{(i)})$ with $s_0^{(i)} \sim \mu_0$ and $g^{(i)}$ is the commanded goal of the trajectory. Note that trajectories need not to be generated from a *goal-directed* agent, in which case $g^{(i)}$ can be randomly drawn from $p(g)$. We denote the empirical goal-conditioned state-action occupancies of $\mathcal{D}^O$ as $d^O(s,a;g)$.

4 Goal-Conditioned f-Advantage Regression

In this section, we introduce Goal-conditioned f-Advantage Regression (GoFAR). We first derive the algorithm in full (Section 4.1), then delve deep into its several appealing properties (Section 4.2-4.4).

4.1 Algorithm

We first show that goal-conditioned state-occupancy matching is a mathematically principled approach for solving general GCRL problems, formalizing the teleportation intuition in the introduction.

Proposition 4.1. *Given any $r(s;g)$, for each g in the support of $p(g)$, define $p(s;g) = \frac{e^{r(s;g)}}{Z(g)}$, where $Z(g) := \int e^{r(s;g)} ds$ is the normalizing constant. Then, the following equality holds:*

$$-\mathrm{D}_{\mathrm{KL}}(d^\pi(s;g) \| p(s;g)) + C = (1-\gamma)J(\pi) + \mathcal{H}(d^\pi(s;g)) \tag{5}$$

where $J(\pi)$ is the GCRL objective (Eq. (2)) with reward $r(s;g)$ and $C := \mathbb{E}_{g\sim p(g)}[\log Z(g)]$.

See Appendix E.1 for proof. This proposition states that, for any choice of reward $r(s;g)$, solving the GCRL problem with a maximum state-entropy regularization is equivalent to optimizing for the goal-conditioned state-occupancy matching objective with target distribution $p(s;g) := \frac{e^{r(s;g)}}{Z(g)}$. Now, the key challenge with optimizing this objective offline is that we cannot sample from $d^\pi(s;g)$. To address this issue, following [32], we first derive an offline lower bound involving an f-divergence (see Appendix A for definition) regularization term, which subsequently enables solving this optimization problem via its dual using purely offline data:

Proposition 4.2. *Assume[2] for all g in support of $p(g)$, $\forall s, d^O(s;g) > 0$ if $p(s;g) > 0$. Then, for any f-divergence that upper bounds the KL-divergence,*

$$-\mathrm{D}_{\mathrm{KL}}(d^\pi(s;g) \| p(s;g)) \geq \mathbb{E}_{(s,g)\sim d^\pi(s,g)}\left[\log \frac{p(s;g)}{d^O(s;g)}\right] - \mathrm{D}_f(d^\pi(s,a;g) \| d^O(s,a;g)) \tag{6}$$

The RHS of (6) can be understood as an f-divergence regularized GCRL objective with reward function $R(s;g) = \log \frac{p(s;g)}{d^O(s;g)}$ (we use capital R to differentiate user-chosen reward R from the environment reward r). Intuitively, this reward encourages visiting states that occur more often in the "expert" state distribution $p(s;g)$ than in the offline dataset, and the f-divergence regularization then ensures that the learned policy is supported by the offline dataset. This choice of reward function can be estimated in practice by training a discriminator [15] $c : S \times G \to (0,1)$ using the offline data:

$$\min_c \mathbb{E}_{g\sim p(g)}\left[\mathbb{E}_{p(s;g)}[\log c(s,g)] + \mathbb{E}_{d^O(s;g)}[\log 1 - c(s,g)]\right] \tag{7}$$

We can in fact obtain a looser lower bound that does not require training a discriminator (see B for a derivation):

$$-\mathrm{D}_{\mathrm{KL}}(d^\pi(s;g) \| p(s;g)) \geq \mathbb{E}_{(s,g)\sim d^\pi(s,g)}[\log p(s;g)] - \mathrm{D}_f(d^\pi(s,a;g) \| d^O(s,a;g)) \tag{8}$$

[2]This assumption is only needed in our technical derivation to avoid division-by-zero issue.

Because $p(s;g) \propto e^{r(s;g)}$, we may substitute $R(s;g) := r(s;g)$ (when the offline dataset contains reward labels) for $\log p(s;g)$ and bypass having to train a discriminator for reward;

Now, for either choice of lower bound, we may pose the optimization problem with respect to valid choices of state-action occupancies directly, introducing the Bellman flow constraint (4):

$$\begin{aligned} &\max_{d(s,a;g)\geq 0} \quad \mathbb{E}_{(s,g)\sim d(s,g)}\left[r(s;g)\right] - \mathrm{D}_f(d(s,a;g)\|d^O(s,a;g)) \\ \text{(P)} \qquad &\text{s.t.} \quad \sum_a d(s,a;g) = (1-\gamma)\mu_0(s) + \gamma \sum_{\tilde{s},\tilde{a}} T(s \mid \tilde{s},\tilde{a}) d(\tilde{s},\tilde{a};g), \forall s \in S, g \in G \end{aligned} \tag{9}$$

This reformulation does not solve the fundamental problem that (9) still requires sampling from $d(s;g)$; however, it has now written the problem in a way amenable to simplification using tools from convex analysis. Now, we show that its dual problem can be reduced to an *unconstrained* minimization problem over the dual variables which serve the role of a value function; importantly, the optimal solution to the dual problem can be used to directly retrieve the optimal primal solution:

Proposition 4.3. *The dual problem to* (9) *is*

$$\text{(D)} \quad \min_{V(s,g)\geq 0} (1-\gamma)\mathbb{E}_{(s,g)\sim\mu_0,p(g)}[V(s;g)] + \mathbb{E}_{(s,a,g)\sim d^O}\left[f_\star\left(R(s;g) + \gamma\mathcal{T}V(s,a;g) - V(s;g)\right)\right], \tag{10}$$

where $f_\star$ denotes the convex conjugate function of f, $V(s;g)$ is the Lagrangian vector, and $\mathcal{T}V(s,a;g) = \mathbb{E}_{s'\sim T(\cdot|s,a)}[V(s';g)]$. Given the optimal V^*, the primal optimal d^* satisfies:

$$d^*(s,a;g) = d^O(s,a;g) f_\star'\left(R(s;g) + \gamma\mathcal{T}V^*(s,a;g) - V^*(s;g)\right), \forall s \in S, a \in A, g \in G \tag{11}$$

A proof is given in Appendix B. Crucially, as neither expectation in (10) depends on samples from d, this objective can be estimated entirely using offline data, making it suitable for offline GCRL. For tabular MDPs, we show that for a suitable choice of f-divergence, the optimal V^* in fact admits closed-form solutions; see Appendix D for details. In the continuous control setting, we can optimize (10) using stochastic gradient descent (SGD) by parameterizing V using a deep neural network.

Then, once we have obtained the optimal (resp. converged) V^*, we propose learning the policy via the following supervised regression update:

$$\max_\pi \mathbb{E}_{g\sim p(g)}\mathbb{E}_{(s,a)\sim d^O(s,a;g)}\left[(f_\star'(R(s;g) + \gamma\mathcal{T}V^*(s,a;g) - V^*(s;g))\log\pi(a \mid s,g)\right] \tag{12}$$

We see that the regression weights are the first-order derivatives of the convex conjugate of f evaluated at the *dual optimal advantage*, $R(s;g) + \gamma\mathcal{T}V^*(s,a;g) - V^*(s;g)$; we refer to this weighting term as *f-advantage*. Hence, we name our overall method *Goal-conditioned f-Advantage Regression (GoFAR)*; an abbreviated high-level pseudocode is provided in Algorithm 1 and a detailed version is provided in Appendix C. In practice, we implement GoFAR with χ^2-divergence, a choice of f-divergence that is stable for off-policy optimization [49, 32, 26]; see Appendix C for details.

Algorithm 1 Goal-Conditioned f-Advantage Regression (Abbreviated); 3-disjoint steps

1: (Optionally) Train discriminator-based reward (7)
2: Train (optimal) dual value function $V^*(s;g)$ (10)
3: Train policy π via f-Advantage Regression (12)

Note that (12) forgos directly minimizing (1) offline, which has been found to suffer from training instability [21]. Instead, it naturally incorporates the primal-dual optimal solutions in a regression loss. Now, we will show that this policy objective has several theoretical and practical benefits for offline GCRL that make it particularly appealing.

4.2 Optimal Goal-Weighting Property

We show that optimizing (12) automatically obtains the *optimal goal-weighting distribution*. That is, GoFAR trains its policy as if all the data is coming from the optimal goal-conditioned policy for (9). In particular, this property is achieved without any explicit hindsight relabeling (see Appendix A for a technical definition), a mechanism that prior works heavily depend on. To this end, we first define $p(g \mid s,a)$ to be the conditional distribution of goals in the offline dataset conditioned on state-action pair (s,a). Then, according to Bayes rule, we have that

$$p(g \mid s,a) = \frac{d^O(s,a;g)p(g)}{d^O(s,a)} \Rightarrow p(g)d^O(s,a;g) = p(g \mid s,a)d^O(s,a) \tag{13}$$

Using this equality, we can rewrite the policy objective (12) as follows:

$$\min_{\pi} -\mathbb{E}_{(s,a)\sim d^O(s,a)}\mathbb{E}_{g\sim p(g|s,a)}\left[(f'_\star(R(s;g)+\gamma\mathcal{T}V^*(s,a;g)-V^*(s;g))\log\pi(a\mid s,g)\right] \quad (14)$$

$$=\min_{\pi} -\mathbb{E}_{(s,a)\sim d^O(s,a)}\mathbb{E}_{g\sim \tilde{p}(g|s,a)}\left[\log\pi(a\mid s,g)\right] \quad (15)$$

where

$$\tilde{p}(g\mid s,a)\propto p(g\mid s,a)\left(f'_\star(R(s;g)+\gamma\mathcal{T}V^*(s,a;g)-V^*(s;g)\right)=p(g\mid s,a)\frac{d^*(s,a;g)}{d^O(s,a;g)} \quad (16)$$

Thus, we see that GoFAR's f-advantage weighting scheme is equivalent to performing supervised policy regression where goals are sampled from $\tilde{p}(g\mid s,a)$. Now, combining (16) and Bayes rule gives

$$d^O(s,a)\tilde{p}(g\mid s,a)\propto d^O(s,a)p(g\mid s,a)\frac{d^*(s,a;g)}{\frac{p(g|s,a)d^O(s,a)}{p(g)}}=p(g)d^*(s,a;g) \quad (17)$$

Thus, we can replace the nested expectations in (15) and obtain that GoFAR policy update amounts to supervised regression of the state-action occupancy distribution of the *optimal* policy to the regularized GCRL problem (9):

$$\pi_{\text{GoFAR}}=\min_{\pi} -\mathbb{E}_{g\sim p(g)}\mathbb{E}_{(s,a)\sim d^*(s,a;g)}\left[\log\pi(a\mid s,g)\right] \quad (18)$$

This derivation makes clear GoFAR's connection with the hindsight goal relabeling mechanism [3] that is ubiquitous in GCRL: GoFAR automatically performs the optimal goal-weighting policy update without any explicit goal relabeling. Furthermore, our derivation also suggests why hindsight relabeling is sub-optimal without further assumption on the reward function [10]: it heuristically chooses $\tilde{p}(g\mid s,a)$ to be the empirical trajectory-wise future achieved goal distribution (i.e., HER), which generally does not coincide with the goals that the optimal policy would reach; see Appendix E.2 for further discussion.

4.3 Uninterleaved Optimization and Performance Guarantee

An additional algorithmic advantage of GoFAR is that it *disentangles* the optimization of the value network and the policy network. This can be observed by noting that GoFAR's advantage term (11) is computed using V^*, the *optimal* solution of the dual problem. This has the practical significance of disentangling the value-function update (10) from the policy update (12), as we do not need to optimize the latter until the former has converged. This disentanglement is in sharp contrast to prior GCRL works [47, 6, 11, 3], which typically involve alternating updates to the critic Q-network and the policy network, a training procedure that has found to be unstable in the offline setting [23]. This is unavoidable for prior works because their advantage functions are estimated using the Q-estimate of the *current* policy, whereas our advantage term naturally falls out from primal-dual optimality.

The uninterleaved and relabeling-free nature of GoFAR also allows us to derive strong performance guarantees. Because V^* is fixed in (12), this policy objective amounts to a *weighted* supervised learning problem. Therefore, we can extend and adapt mature theoretical results for analyzing Behavior Cloning [1, 42, 46] as well as finite sample error guarantees for weighted regression [7] to obtain statistical guarantees on GoFAR's performance with respect to the optimal π^* for (9):

Theorem 4.1. *Assume* $\sup_{s,a,g}\frac{d^*(s,a;g)}{d^O(s,a;g)}\leq M$ *and* $\sup|r(s,g)|\leq R_{\max}$. *Consider a policy class* $\Pi:\{S\to\Delta(A)\}$ *such that* $\pi^*\in\Pi$. *Then, for any* $\delta\in(0,1]$, *with probability at least* $1-\delta$, *GoFAR* (18) *will return a policy* $\hat{\pi}$ *such that:*

$$V^*-V^{\hat{\pi}}\leq\frac{2R_{\max}M}{(1-\gamma)^2}\sqrt{\frac{\ln(|\Pi|/\delta)}{N}} \quad (19)$$

Notably, the error shrinks as the size of the *offline* data N increases, requiring no dependency on access to data from the "expert" distribution d^*. This provides a theoretical basis for GoFAR's empirical scalability as we are guaranteed to obtain good results when the offline data becomes more expansive. In contrast, prior regression-based GCRL methods cannot be easily reduced to a simple weighted regression with respect to the desired goal distribution $p(g)$, so they only obtain weaker results under stronger assumptions on the offline data (e.g, full state-space coverage) as well as the policy; see Appendix E.3 for discussion.

Table 1: Discounted Return, averaged over 10 seeds.

Task	Supervised Learning			Actor-Critic	
	GoFAR (Ours)	**WGCSL**	**GCSL**	**AM**	**DDPG**
FetchReach	28.2 ± 0.61	21.9± 2.13 (1.0)	20.91 ± 2.78 (1.0)	**30.1** ± 0.32 (0.5)	29.8 ± 0.59 (0.2)
FetchPick	**19.7** ± 2.57	9.84 ± 2.58 (1.0)	8.94 ± 3.09 (1.0)	18.4 ± 3.51 (0.5)	16.8 ± 3.10 (0.5)
FetchPush (⋆)	**18.2** ± 3.00	14.7 ± 2.65 (1.0)	13.4 ± 3.02 (1.0)	14.0 ± 2.81 (0.5)	12.5 ± 4.93 (0.5)
FetchSlide	2.47 ± 1.44	**2.73** ± 1.64 (1.0)	1.75 ± 1.3(1.0)	1.46 ± 1.38 (0.5)	1.08 ± 1.35 (0.5)
HandReach (⋆)	**11.5** ± 5.26	5.97 ± 4.81 (1.0)	1.37 ± 2.21 (1.0)	0. ± 0.0 (0.5)	0.81 ± 1.73 (0.5)
D'ClawTurn (⋆)	**9.34** ± 3.15	0.0 ± 0.0 (1.0)	0.0 ± 0.0 (1.0)	2.82± 1.71 (1.0)	0.0± 0.0 (0.2)
Average Rank	**1.5**	3	4.17	2.83	4

4.4 Goal-Conditioned Planning

Next, we show that GoFAR can be used to learn a goal-conditioned planner that supports zero-shot transfer to other domains of the same task. The key insight is that GoFAR's value function objective (10) does not depend on actions assuming deterministic transitions. This is because given a transition $(s, a, s', g) \sim d^O$, $\mathcal{T}V(s, a; g) = V(s'; g)$, so we can rewrite the second term in (10) as $\mathbb{E}_{(s,a,s',g)\sim d^O}\left[f_\star\left(R(s; g) + \gamma V(s'; g) - V(s; g)\right)\right]$, which does not depend on the action. This property enables us to learn a *goal-centric* value function that is independent of the agent's state space because the agent's actions are not relevant. Specifically, we propose the following objective:

$$\min_{V(\phi(s),g)\geq 0}(1-\gamma)\mathbb{E}_{(s,g)\sim\mu_0,p(g)}[V(\phi(s); g)]+\mathbb{E}_{(s,s',g)\sim d^O}\left[f_\star\left(R(s; g) + \gamma V(\phi(s'), g) - V(\phi(s), g)\right)\right] \tag{20}$$

We can think of this objective as learning a value function with the inductive bias that the first operation transforms the state input to the goal space via ϕ. Since V is now independent of the agent, we can use f-advantage regression to instead learn a *goal-conditioned planner*:

$$\max_{\pi}\mathbb{E}_{g\sim p(g)}\mathbb{E}_{(s,s',g)\sim d^O}\left[(f'_\star(R(s; g) + \gamma V^*(\phi(s'); g) - V^*(\phi(s); g))\log\pi(\phi(s') \mid \phi(s), g)\right] \tag{21}$$

where π now outputs the next subgoal $\phi(s')$ conditioned on the current achieved goal $\phi(s)$ and the desired goal g. In our experiments, we show how this planner can achieve hierarchical control through zero-shot transfer subgoal plans to a new target domain.

5 Experiments

We pose the following questions and provide affirmative answers in our experiments:

1. Is GoFAR effective for offline GCRL? What components are important for performance?
2. Is GoFAR more robust to stochastic environments than hindsight relabeling methods?
3. Can GoFAR be applied to a real-robotics system?
4. Can GoFAR learn a goal-conditioned planner for zero-shot cross-embodiment transfer?

5.1 Offline GCRL

Tasks. In our simulation experiments, we consider six distinct environments. They include four robot manipulation environments[39]: FetchReach, FetchPickAndPlace, FetchPush, and FetchSlide, and two dexterous manipulation environments: HandReach [39], and D'ClawTurn [2]. All tasks use sparse reward, and their respective goal distributions are defined over valid configurations in either the robot space or object space, depending on whether the task involves object manipulation. The offline dataset for each task is collected by either a random policy or a mixture of 90% random policy and 10% expert policy [32, 21], depending on whether random data provides enough coverage of the desired goal distribution. The first five tasks and their datasets are taken from [47]. See Appendix G.3 for dataset details and Appendix F for detailed task descriptions and figures.

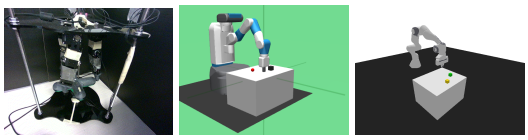


Figure 3: D'Claw (left); Cross-Embodiment transfer source (middle) and target (right) domains.

Algorithms. We compare to state-of-art offline GCRL algorithms, consisting of both regression-based and actor-critic methods. The regression-based methods are: (1) **GCSL** [14], which incorporates hindsight relabeling in conjunction with behavior cloning to clone actions that lead to a specified goal, and (2) **WGCSL** [47], which improves upon GCSL by incorporating discount factor and advantage weighting into the supervised policy learning update. The actor-critic methods are (1) **DDPG** [3], which adapts DDPG [29] to the goal-conditioned setting by incorporating hindsight relabeling, and (2) **ActionableModel (AM)** [6], which incorporates conservative Q-Learning [24] as well as goal-chaining on top of an actor-critic method.

We use tuned hyperparameters for each baseline on all tasks; in particular, we search the best HER ratio from $\{0.2, 0.5, 1.0\}$ for each baseline on each task separately and report the best-performing one. For GoFAR, we use identical hyperparameters as WGCSL for the shared network components and do not tune further. We train each method for 10 seeds, and each training run uses 400k minibatch updates of size 512. Complete architecture and hyperparameter table as well as additional training details are provided in Appendix G.

Evaluations and Results. We report the **discounted return** using the sparse binary task reward. This metric rewards algorithms that reach goals as *fast* as possible and stay in the goal region thereafter. In Appendix H, we also report the final **success rate** using the same sparse binary criterion. Because the task reward is binary, these metrics do not take account into how *precisely* a goal is being reached. Therefore, we additionally report the **final distance** to goal in Appendix H.

The full discounted return results are shown in Table 1; $(\star)$ indicates statistically significant improvement over the second best method under a two-sample t-test. The fraction inside $()$ indicates the best HER rate for each baseline. As shown, GoFAR attains the best overall performance across six tasks, and the results are statistically significant on three tasks, including the two more challenging dexterous manipulation tasks. In Table 6 in Appendix H, we find GoFAR is also superior on the final distance metric. In other words, GoFAR reaches the goals fastest and most precisely.

Ablations. Recall that GoFAR achieves optimal goal-weighting, and does not require the HER heuristic that has been key to prior GCRL approaches. We now experimentally confirm that the baselines are not performant without HER. Conversely, we investigate whether HER can help GoFAR. Additionally, we evaluate GoFAR using the sparse binary reward, which implements the looser lower bound in (8). Finally, we replace χ^2-divergence with KL to understand whether the f-divergence lower bound in (6) is necessary. The results are shown in Figure 4 (task-specific breakdown is included in Appendix H). Compared to their original performances (colored dash lines), baselines without HER suffer severely, especially GCSL and WGCSL: both these methods in fact require 100% relabeling and rely solely on HER for positive learning signals. In sharp contrast, GoFAR is also a regression-based method, yet it performs very well without HER and its performance is unaffected by adding HER. These findings are consistent with the theoretical results of Section 4.2. GoFAR (Binary) performs slightly worse than GoFAR, which is to be expected due to the looser bound; however, it is still better than all baselines, which use the same binary reward and are aided by HER. Again, this ablation highlights the merit of our optimization approach. Finally, GoFAR (KL) suffers from numerical instability and fails to learn, showing that our general f-divergence lower bound (6) is not only of theoretical value but also practically significant.

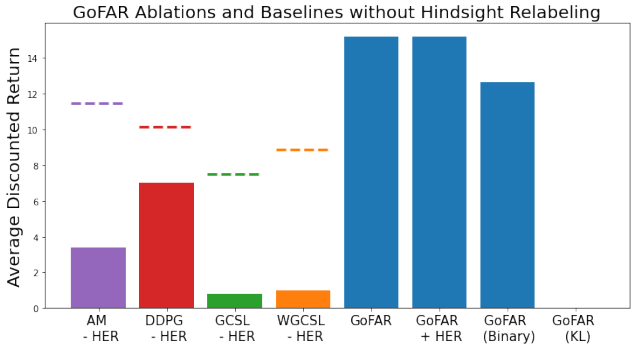


Figure 4: Offline GCRL ablation studies. While GoFAR is robust to hindsight relabeling, removing it is highly detrimental to all baselines.

5.2 Robustness in Stochastic Offline GCRL Settings

For this experiment, we create noisy variants of the FetchReach environment by adding varying levels of white Gaussian noise to policy action outputs. FetchReach is well-suited for this experiment because all baselines attain satisfactory performance and even outperform GoFAR in Table 1. Further-

more, to isolate the effect of noisy actions, we use GoFAR with binary reward so that all methods now use the same reward input and perform comparatively without added noise. We consider zero-mean Gaussian noise of σ value from $0.5, 1$, and 1.5. For each noise level, we re-collect offline data by executing random actions in the respective noisy environment and train on these noisy offline random data.

The average discounted returns over different noise levels are illustrated in Figure 5. As shown, at noise level 0.5, while GoFAR's performance is not effected, all baselines already exhibit degraded performance. In particular, DDPG sharply degrades, underperforming GoFAR despite better original performance. At 1.0, the gap continues to widen, and AM notably collapses, highlighting its sensitivity to noise. This is expected as AM's relabeling mechanism also samples future goals from other trajectories and labels the selected goals with their current Q-value estimates; this procedure becomes highly noisy when the dataset already exhibits high stochasticity, thus contributing to the instability observed in this experiment. It is only at noise level 1.5 that GoFAR's performance degradation becomes comparable to those of other methods. Thus, GoFAR's relabeling-free optimization indeed confers greater robustness to environmental stochasticity; in contrast, all baselines suffer from "false positive" relabeled goals due to such disturbances.

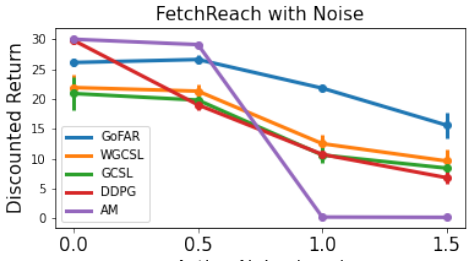


Figure 5: Stochastic environment evaluation. GoFAR is more robust to stochastic environments due to its lack of hindsight goal relabeling.

5.3 Real-World Robotic Dexterous Manipulation

Now, we evaluate on a real D'Claw [2] dexterous manipulation robot (Figure 3 (left)). The task is to have the tri-finger robot rotate the valve to a specified goal angle from its initial angle; see Appendix F for detail. The offline dataset contains 400K transitions, collected by executing purely random actions in the environment, which provides sufficient coverage of the goal space. Compared to the simulated D'Claw, an additional challenge is the inherent stochasticity on a real robot system due to imperfect actuation and hardware wear and tear. Given our conclusion in the stochastic environment experiment above, we should expect GoFAR to perform better.

The training curve of final distances averaged over 3 seeds is shown in Figure 6. We see that GoFAR exhibits smooth progress over training and significantly outperforms all baselines. At the end of training, GoFAR is able to rotate the valve on average within **10** degrees of the specified goal angle. In contrast, most baselines do not make any progress on this difficult task and learns policies that do not manage to turn the valve at all. AM is the only baseline that learns a non-trivial policy; however, its training fluctuates significantly, and the learned policy often executes extreme actions during evaluation that fail to reach the desired goal. We provide qualitative analysis in Appendix H.3, and videos are included in the supplement.

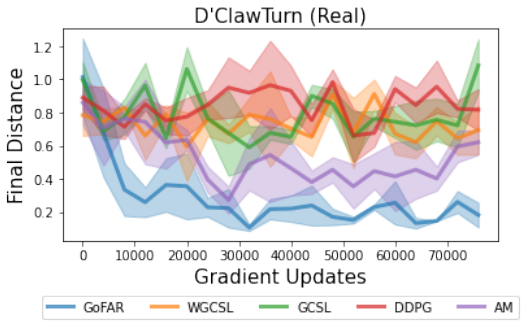


Figure 6: Real-robot results.

5.4 Zero-Shot Transfer Across Robots

For this experiment, we consider zero-shot transfer from the FetchPush task (source domain) to an equivalent planar object pushing task using the 7-DOF Franka Panda robot [12]; the environments are illustrated in Figure 3. In the target domain, we assume a pre-programmed or learned low-level goal-reaching controller $\pi_{\text{low}}(a \mid s, g)$

Table 2: Zero-shot transfer results.

Method	Success Rate
GoFAR Planner (Oracle)	84%
GoFAR Hierarchical Controller	37%
Low-Level Controller	19%

that can push objects over short distances but fails for longer tasks. For these distant goals, we train the goal-conditioned GoFAR value function $V^*(\phi(s); g)$ and planner $\pi(\phi(s') \mid \phi(s), g)$ according to (20) and (21), to set goals for π_{low}. We use the same offline FetchPush data as in Section 5.1. We compare two approaches: (1) naively using the **Low-Level Controller**, and (2) using the GoFAR planner to guide the low-level controller (**GoFAR Hierarchical Controller**). See Appendix G.4 for additional details. We report the success rate over 100 random distant test goals in Table 2. We see that naively commanding the low-level controller with these distant goals indeed results in very low success rate. When the controller is augmented with GoFAR planner, the success rate nearly doubles. Note that our planner itself is much superior: if all subgoals are perfectly reached, **GoFAR Planner (Oracle)** achieves 84% success. See Appendix G.4 for additional experimental details. These experiments validate GoFAR's ability to zero-shot transferring subgoal plans to enhance the capabability of low-level controllers. Note that we have no access to any data from the target domain, and this planning capability naturally emerges from our training objectives and does not require any change in the algorithm. We show qualitative results in Appendix H.4 and videos in the supplement.

6 Conclusion

We have presented GoFAR, a novel regression-based offline GCRL algorithm derived from a state-occupancy matching perspective. GoFAR is relabeling-free and enjoys uninterleaved training, making it both theoretically and practically advantageous compared to prior state-of-art. Furthermore, GoFAR supports training a goal-conditioned planner with promising zero-shot transfer capability. Through extensive experiments, we have validated the practical utility of GoFAR in various challenging problem settings, showing significant improvement over prior state-of-art. We believe that GoFAR's strong theoretical foundation and empirical performance make it an ideal candidate for scalable skill learning in the real world.

Acknowlegement

We thank Edward Hu and Oleh Rybkin for their feedback on an earlier draft of the paper, Kun Huang for assistance on the real-robot experiment setup, and Selina Li for assistance on graphic design. This work is funded in part by an Amazon Research Award, gift funding from NEC Laboratories America, NSF Award CCF-1910769, NSF Award CCF-1917852 and ARO Award W911NF-20-1-0080. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Alekh Agarwal, Nan Jiang, and Sham M Kakade. Reinforcement learning: Theory and algorithms. 2019.

[2] Michael Ahn, Henry Zhu, Kristian Hartikainen, Hugo Ponte, Abhishek Gupta, Sergey Levine, and Vikash Kumar. Robel: Robotics benchmarks for learning with low-cost robots. In *Conference on robot learning*, pages 1300–1313. PMLR, 2020.

[3] 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.

[4] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[5] Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals. In *International Conference on Machine Learning*, pages 1430–1440. PMLR, 2021.

[6] Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jake Varley, Alex Irpan, Benjamin Eysenbach, Ryan Julian, Chelsea Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*, 2021.

[7] Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. *Advances in neural information processing systems*, 23, 2010.

[8] Bo Dai, Niao He, Yunpeng Pan, Byron Boots, and Le Song. Learning from conditional distributions via dual embeddings, 2016.

[9] Sudeep Dasari, Frederik Ebert, Stephen Tian, Suraj Nair, Bernadette Bucher, Karl Schmeckpeper, Siddharth Singh, Sergey Levine, and Chelsea Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

[10] Ben Eysenbach, Xinyang Geng, Sergey Levine, and Russ R Salakhutdinov. Rewriting history with inverse rl: Hindsight inference for policy improvement. *Advances in neural information processing systems*, 33:14783–14795, 2020.

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

[12] Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. Multi-goal reinforcement learning environments for simulated franka emika panda robot. *arXiv preprint arXiv:2106.13687*, 2021.

[13] Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods, 2019.

[14] 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.

[15] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[16] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[17] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[18] Leslie Pack Kaelbling. Learning to achieve goals. Citeseer.

[19] Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv preprint arXiv:2104.08212*, 2021.

[20] Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization, 2020.

[21] Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. DemoDICE: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022.

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

[23] Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *arXiv preprint arXiv:1906.00949*, 2019.

[24] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

[25] Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pages 45–73. Springer, 2012.

[26] Jongmin Lee, Wonseok Jeon, Byung-Jun Lee, Joelle Pineau, and Kee-Eung Kim. Optidice: Offline policy optimization via stationary distribution correction estimation. *arXiv preprint arXiv:2106.10783*, 2021.

[27] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

[28] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[29] 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.

[30] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020.

[31] Yecheng Ma, Dinesh Jayaraman, and Osbert Bastani. Conservative offline distributional reinforcement learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[32] Yecheng Jason Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Smodice: Versatile offline imitation learning via state occupancy matching. *arXiv preprint arXiv:2202.02433*, 2022.

[33] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *arXiv preprint arXiv:1906.04733*, 2019.

[34] Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality, 2020.

[35] Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv preprint arXiv:1912.02074*, 2019.

[36] Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. *Advances in Neural Information Processing Systems*, 32, 2019.

[37] 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.

[38] Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[39] 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.

[40] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[41] R. Tyrrell Rockafellar. *Convex analysis*. Princeton Mathematical Series. Princeton University Press, Princeton, N. J., 1970.

[42] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[43] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1312–1320, Lille, France, 07–09 Jul 2015. PMLR.

[44] Yannick Schroecker and Charles Lee Isbell. Universal value density estimation for imitation learning and goal-conditioned reinforcement learning, 2021.

[45] Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158*, 2018.

[46] Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. *Advances in Neural Information Processing Systems*, 33:15737–15749, 2020.

[47] 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. In *International Conference on Learning Representations*, 2022.

[48] Siyuan Zhang and Nan Jiang. Towards hyperparameter-free policy selection for offline reinforcement learning. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[49] Zhuangdi Zhu, Kaixiang Lin, Bo Dai, and Jiayu Zhou. Off-policy imitation learning from observations. *Advances in Neural Information Processing Systems*, 33, 2020.

[50] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix B.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We include the code and data for our simulation experiments in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix G.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix H.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix G.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include videos in our supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Part I

Appendix

These appendices contain material supplementary to submission 7863: "Offline Goal-Conditioned Reinforcement Learning via f-Advantage Regression"

Table of Contents

A Additional Technical Background **15**
- A.1 f-Divergence and Fenchel Duality . . . 16
- A.2 Hindsight Goal-Relabeling . . . 16

B Proofs **16**
- B.1 Proof of Proposition 4.2 . . . 16
- B.2 Proof of Proposition 4.3 . . . 18
- B.3 Proof of Theorem 4.1 . . . 18

C GoFAR Technical Details **19**
- C.1 Discriminator Training . . . 19
- C.2 GoFAR with common f-Divergences . . . 19
- C.3 Full Pseudocode . . . 20

D GoFAR for Tabular MDPs **20**

E Additional Technical Discussion **21**
- E.1 Connecting Goal-Conditioned State-Matching and Probabilistic GCRL . . . 21
- E.2 Optimality Conditions for Hindsight Relabeling . . . 22
- E.3 Theoretical Comparison to Prior Regression-based GCRL methods . . . 22

F Task Descriptions **22**
- F.1 Fetch Tasks . . . 23
- F.2 Hand Reach . . . 24
- F.3 D'ClawTurn (Simulation) . . . 24
- F.4 D'ClawTurn (Real) . . . 24

G Experimental Details **24**
- G.1 Baseline Implementation Details . . . 24
- G.2 Architectures and Hyperparameters . . . 25
- G.3 Offline GCRL Experiments . . . 25
- G.4 Zero-Shot Transfer Experiments . . . 25

H Additional Results **26**
- H.1 Offline GCRL Full Results . . . 26
- H.2 Ablations . . . 27
- H.3 Real-World Dexterous Manipulations . . . 27
- H.4 Zero-Shot Plan Transfer . . . 28

A Additional Technical Background

We provide some technical definitions that are needed in our proofs in Appendix B and discussions of hindsight relabeling in Section 4.1 of the main text as well as Appendix E.2.

A.1 f-Divergence and Fenchel Duality

These definitions are adapted from [32, 34].

Definition A.1 (f-divergence)**.** For any continuous, convex function f and two probability distributions $p, q \in \Delta(\mathcal{X})$ over a domain $\mathcal{X}$, the f-divergence of p computed at q is defined as

$$\mathrm{D}_f(p\|q) = \mathbb{E}_{x\sim q}\left[f\left(\frac{p(x)}{q(x)}\right)\right] \tag{22}$$

Some common choices of f-divergence includes the KL-divergence and the χ^2-divergence, which corresponds to choosing $f(x) = x\log x$ and $f(x) = \frac{1}{2}(x-1)^2$, respectively.

Definition A.2 (Fenchel conjugate)**.** Given a vector space $\mathcal{X}$ with inner-product $\langle\cdot,\cdot\rangle$, the *Fenchel conjugate* $f_\star : \mathcal{X}_\star \to \mathbb{R}$ of a convex and differentiable function $f : \Omega \to \mathbb{R}$ is

$$f_\star(y) \coloneqq \max_{x\in\mathcal{X}} \langle x, y\rangle - f(x) \tag{23}$$

and any maximizer x^* of $f_\star(y)$ satisfies $x^* = f'_\star(y)$.

For an f-divergence, under mild realizability assumptions [8] on f, the Fenchel conjugate of $D_f(p\|q)$ at $y : \mathcal{X} \to \mathbb{R}$ is

$$\mathrm{D}_{\star,f}(y) = \max_{p\in\Delta(\mathcal{X})} \mathbb{E}_{x\sim p}[y(x)] - \mathrm{D}_f(p\|q) \tag{24}$$

$$= \mathbb{E}_{x\sim q}[f_\star(y(x))] \tag{25}$$

and any maximizer p^* of $\mathrm{D}_{\star,f}(y)$ satisfies $p^*(x) = q(x)f'_\star(y(x))$. These optimality conditions can be seen as extensions of the KKT-condition.

A.2 Hindsight Goal-Relabeling

We provide a mathematical formalism of hindsight goal relabeling [3].

Definition A.3. Given a state s_t from a trajectory $\tau = \{s_0, a_0, r_0, ..., s_T; g\}$, hindsight goal-relabeling is the goal-relabeling distribution

$$p_{\mathrm{HER}}(g \mid s_t, a_t, \tau) = q[\phi(s_t), ..., \phi(s_T)] \tag{26}$$

where q is some categorical distribution taking values in $\{\phi(s_t), ..., \phi(s_T)\}$.

That is, the relabeled goal is selected from some distribution goals that are reached in the future in the same trajectory. The most canonical choice of q, known as *hindsight experience replay* (HER), selects q to be the uniform distribution. Once a goal $\tilde{g}$ is chosen, the reward label is also re-computed using the reward function assumed by the algorithm: $r_t \coloneqq r(s_t, \tilde{g})$.

B Proofs

In this section, we restate propositions and theorems in the paper and present their proofs.

B.1 Proof of Proposition 4.2

Proposition B.1. *Assume for all g in support of $p(g)$, $\forall s, d^O(s;g) > 0$ if $p(s;g) > 0$. Then, for any f-divergence that upper bounds the KL-divergence,*

$$-\mathrm{D}_{\mathrm{KL}}(d^\pi(s;g)\|p(s;g)) \geq \mathbb{E}_{(s,g)\sim d^\pi(s,g)}\left[\log\frac{p(s;g)}{d^O(s;g)}\right] - \mathrm{D}_f(d^\pi(s,a;g)\|d^O(s,a;g)) \tag{27}$$

$$\geq \mathbb{E}_{(s,g)\sim d^\pi(s,g)}\left[\log p(s;g)\right] - \mathrm{D}_f(d^\pi(s,a;g)\|d^O(s,a;g)) \tag{28}$$

Proof. We first present and prove some technical lemma needed to prove this result. The following lemmas and proofs are adapted from [32]; in particular, we extend these known results to the goal-conditioned setting.

Lemma B.1. *For any pair of valid occupancy distributions d_1 and d_2, we have*

$$\mathrm{D_{KL}}(d_1(s;g)\|d_2(s;g) \leq \mathrm{D_{KL}}(d_1(s,a;g)\|d_2(s,a;g)) \tag{29}$$

Proof. This lemma hinges on proving the following lemma first.

Lemma B.2.

$$\mathrm{D_{KL}}\left(d_1(s,a,s';g)\|d_2(s,a,s';g)\right) = \mathrm{D_{KL}}\left(d_1(s,a;g)\|d_2(s,a;g)\right) \tag{30}$$

Proof.

$$\begin{aligned}
&\mathrm{D_{KL}}\left(d_1(s,a,s';g)\|d_2(s,a,s';g)\right)\\
=&\int_{S\times A\times S\times G} p(g)d_1(s,a,s';g)\log\frac{d_1(s,a;g)\cdot T(s'\mid s,a)}{d_2(s,a;g)\cdot T(s'\mid s,a)}ds'dadsdg\\
=&\int_{S\times A\times S\times G} p(g)d_1(s,a,s';g)\log\frac{d_1(s,a;g)}{d_2(s,a;g)}ds'dadsdg\\
=&\int_{S\times A\times G} p(g)d_1(s,a;g)\log\frac{d_1(s,a;g)}{d_2(s,a;g)}dadsdg\\
=&\mathrm{D_{KL}}\left(d_1(s,a;g)\|d_2(s,a;g)\right)
\end{aligned}$$

□

Using this result, we can prove Lemma B.1:

$$\begin{aligned}
&\mathrm{D_{KL}}\left(d_1(s,a;g)\|d_2(s,a;g)\right)\\
=&\mathrm{D_{KL}}\left(d_1(s,a,s';g)\|d_2(s,a,s';g)\right)\\
=&\int_{S\times A\times S\times G} p(g)d_1(s,a,s';g)\log\frac{d_1(s,a;g)\cdot T(s'\mid s,a)}{d_2(s,a;g)\cdot T(s'\mid s,a)}ds'dadsdg\\
=&\int_{S\times A\times S\times G} p(g)d_1(s;g)\pi_1(a\mid s,g)T(s'\mid s,a)\log\frac{d_1(s,a;g)\cdot T(s'\mid s,a)}{d_2(s,a;g)\cdot T(s'\mid s,a)}ds'dadsdg\\
=&\int p(g)d_1(s;g)\pi_1(a\mid s,g)T(s'\mid s,a)\log\frac{d_1(s;g)}{d_2(s;g)}ds'dadsdg\\
+&\int p(g)d_1(s;g)\pi_1(a\mid s,g)T(s'\mid s,a)\log\frac{\pi_1(a\mid s,g)T(s'\mid s,a)}{\pi_2(a\mid s,g)T(s'\mid s,a)}ds'dadsdg\\
=&\int p(g)d_1(s;g)\log\frac{d_1(s;g)}{d_2(s;g)}dsdg+\int p(g)d_1(s;g)\pi_1(a\mid s,g)\log\frac{\pi_1(a\mid s,g)}{\pi_2(a\mid s,g)}dadsdg\\
=&\mathrm{D_{KL}}\left(d_1(s;g)\|d_2(s;g)\right)+\mathrm{D_{KL}}\left(\pi_1(a\mid s,g)\|\pi_2(a\mid s,g)\right)\\
\geq&\mathrm{D_{KL}}\left(d_1(s;g)\|d_2(s;g)\right)
\end{aligned}$$

□

Now given these technical lemmas, we have

$$\begin{aligned}
&D_{\mathrm{KL}}\left(d^\pi(s;g)\|p(s;g)\right)\\
=&\int p(g)d^\pi(s;g)\log\frac{d^\pi(s;g)}{p(s;g)}\cdot\frac{d^O(s;g)}{d^O(s;g)}dsdg, \quad \text{we assume that } d^O(s;g)>0 \text{ whenever } p(s;g)>0.\\
=&\int p(g)d^\pi(s;g)\log\frac{d^O(s;g)}{p(s;g)}dsdg+\int p(g)d^\pi(s;g)\log\frac{d^\pi(s;g)}{d^O(s;g)}dsdg\\
\leq&\mathbb{E}_{(s,g)\sim d^\pi(s,g)}\left[\log\frac{d^O(s;g)}{p(s;g)}\right]+\mathrm{D_{KL}}\left(d^\pi(s,a;g)\|d^O(s,a;g)\right)
\end{aligned}$$

where the last step follows from Lemma B.1. Then, for any $\mathrm{D}_f \geq \mathrm{D_{KL}}$, we have that

$$-D_{\mathrm{KL}}\left(d^\pi(s;g)\|p(s;g)\right) \geq \mathbb{E}_{(s,g)\sim d^\pi(s,g)}\left[\log\frac{p(s;g)}{d^O(s;g)}\right]-\mathrm{D}_f\left(d^\pi(s,a;g)\|d^O(s,a;g)\right) \tag{31}$$

Then, since $\mathbb{E}_{(s,g)\sim d^\pi(s,g)}\left[\log \frac{1}{d^O(s;g)}\right] \geq 0$, we also obtain the following looser bound:

$$-D_{\mathrm{KL}}\left(d^\pi(s;g)\|p(s;g)\right) \geq \mathbb{E}_{(s,g)\sim d^\pi(s,g)}\left[\log p(s;g)\right] - \mathrm{D}_f\left(d^\pi(s,a;g)\|d^O(s,a;g)\right) \tag{32}$$

□

B.2 Proof of Proposition 4.3

Proposition B.2. *The dual problem to* (9) *is*

$$\text{(D)} \quad \min_{V(s,g)\geq 0}(1-\gamma)\mathbb{E}_{(s,g)\sim\mu_0,p(g)}[V(s;g)]+\mathbb{E}_{(s,a,g)\sim d^O}\left[f_\star\left(r(s,g)+\gamma\mathcal{T}V(s,a;g)-V(s;g)\right)\right], \tag{33}$$

where $f_\star$ denotes the convex conjugate function of f, $V(s;g)$ is the Lagrangian vector, and $\mathcal{T}V(s,a;g) = \mathbb{E}_{s'\sim T(\cdot|s,a)}[V(s';g)]$. Given the optimal V^*, the primal optimal d^* satisfies:

$$d^*(s,a;g) = d^O(s,a;g)f'_\star\left(r(s,g)+\gamma\mathcal{T}V^*(s,a,g)-V^*(s,g)\right), \forall s\in S, a\in A, g\in G \tag{34}$$

Proof. We begin by writing the Lagrangian dual of the primal problem:

$$\begin{aligned}&\min_{V(s;g)\geq 0}\max_{d(s,a;g)\geq 0}\mathbb{E}_{(s,g)\sim d(s,g)}\left[\log\left(r(s;g)\right)\right] - \mathrm{D}_f(d(s,a;g)\|d^O(s,a;g))\\ &+\sum_{s,g}p(g)V(s;g)\left((1-\gamma)\mu_0(s)+\gamma\sum_{\tilde{s},\tilde{a}}T(s\mid\tilde{s},\tilde{a})d(\tilde{s},\tilde{a};g)-\sum_a d(s,a;g)\right)\end{aligned} \tag{35}$$

where $p(g)V(s;g)$ is the Lagrangian vector. Then, we note that

$$\sum_{s,g}V(s;g)\sum_{\tilde{s},\tilde{a}}T(s\mid\tilde{s},\tilde{a})d(\tilde{s},\tilde{a};g) = \sum_{\tilde{s},\tilde{a},g}d(\tilde{s},\tilde{a};g)\sum_s T(s\mid\tilde{s},\tilde{a})V(s;g) = \sum_{s,a,g}d(s,a;g)\mathcal{T}V(s,a;g) \tag{36}$$

Using this, we can rewrite (35) as

$$\begin{aligned}\min_{V(s;g)\geq 0}\max_{d(s,a;g)\geq 0}&(1-\gamma)\mathbb{E}_{(s,g)\sim(\mu_0,p(g))}[V(s;g)] + \mathbb{E}_{(s,a,g)\sim d}\left[(r(s;g)+\gamma\mathcal{T}V(s,a;g)-V(s;g))\right]\\ &- \mathrm{D}_f(d(s,a;g)\|d^O(s,a;g))\end{aligned} \tag{37}$$

And finally,

$$\begin{aligned}\min_{V(s,g)\geq 0}(1-\gamma)\mathbb{E}_{(s,g)\sim(\mu_0,p(g))}[V(s;g)] + \max_{d(s,a;g)\geq 0}&\mathbb{E}_{(s,a,g)\sim d}\left[(r(s,g)+\gamma\mathcal{T}V(s,a;g)-V(s;g))\right]\\ &- \mathrm{D}_f(d(s,a;g)\|d^O(s,a;g))\end{aligned} \tag{38}$$

Now, we make the key observation that the inner maximization problem in (38) is in fact the Fenchel conjugate of $\mathrm{D}_f(d(s,a,g)\|d^O(s,a,g))$ at $r(s,g)+\gamma\mathcal{T}V(s,a,g)-V(s,g)$. Therefore, we can reduce (38) to an unconstrained minimization problem over the dual variables

$$\min_{V(s,g)\geq 0}(1-\gamma)\mathbb{E}_{(s,g)\sim\mu_0,p(g)}[V(s;g)] + \mathbb{E}_{(s,a,g)\sim d^O}\left[f_\star\left(r(s,g)+\gamma\mathcal{T}V(s,a;g)-V(s;g)\right)\right], \tag{39}$$

and consequently, we can relate the dual-optimal V^* to the primal-optimal d^* using Fenchel duality (see Appendix A:

$$d^*(s,a;g) = d^O(s,a;g)f'_\star\left(r(s,g)+\gamma\mathcal{T}V^*(s,a,g)-V^*(s,g)\right), \forall s\in S, a\in A, g\in G, \tag{40}$$

as desired. □

B.3 Proof of Theorem 4.1

Theorem B.3. *Assume* $\sup_{s,a,g}\frac{d^*(s,a;g)}{d^O(s,a;g)} \leq M$ *and* $\sup|r(s,g)| \leq R_{\max}$. *Consider a policy class* $\Pi : \{S \to \Delta(A)\}$ *such that* $\pi^* \in \Pi$. *Then, for any* $\delta \in (0,1]$, *with probability at least* $1-\delta$, *GoFAR* (18) *will return a policy* $\hat{\pi}$ *such that:*

$$V^* - V^{\hat{\pi}} \leq \frac{2R_{\max}M}{(1-\gamma)^2}\sqrt{\frac{\ln(|\Pi|/\delta)}{N}} \tag{41}$$

where $V^\pi := \mathbb{E}_{(s,g)\sim(\mu_0,g)}[V(s;g)]$.

Proof. We begin by deriving an upper bound using the performance difference lemma [1]:

$$V^* - V^{\hat{\pi}} \leq \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^*, g\sim p(g)}\mathbb{E}_{a\sim\pi^*(\cdot|s,g)}A^{\hat{\pi}}(s,a;g) \tag{42}$$

Then, using standard algebraic manipulations, we have:

$$\begin{aligned}
&\frac{1}{1-\gamma}\mathbb{E}_{s\sim d^*, g\sim p(g)}\mathbb{E}_{a\sim\pi^*(\cdot|s,g)}A^{\hat{\pi}}(s,a;g)\\
=&\frac{1}{1-\gamma}\mathbb{E}_{s\sim d^*, g\sim p(g)}\left[\mathbb{E}_{a\sim\pi^*(\cdot|s,g)}A^{\hat{\pi}}(s,a;g) - \mathbb{E}_{a\sim\hat{\pi}(\cdot|s,g)}A^{\hat{\pi}}(s,a;g)\right]\\
\leq&\frac{R_{\max}}{(1-\gamma)^2}\mathbb{E}_{s\sim d^*, g\sim p(g)}\left[\|\pi^*(\cdot \mid s,g) - \hat{\pi}(\cdot \mid s,g)\|_1\right]\\
\leq&\frac{2R_{\max}}{(1-\gamma)^2}\mathbb{E}_{s\sim d^*, g\sim p(g)}\left[\|\pi^*(\cdot \mid s,g) - \hat{\pi}(\cdot \mid s,g)\|_{\mathrm{TV}}\right]
\end{aligned} \tag{43}$$

Then, since $\sup_{s,a,g}\frac{d^*(s,a;g)}{d^O(s,a;g)} \leq M$, we can use Hoeffding's inequality with weighted empirical loss [7] to obtain that:

$$V^* - V^{\hat{\pi}} \leq \frac{2R_{\max}M}{(1-\gamma)^2}\sqrt{\frac{\ln(|\Pi|/\delta)}{N}} \tag{44}$$

□

C GoFAR Technical Details

In this section, we provide additional technical details of GoFAR that are omitted in the main text. These include (1) detail of the GoFAR discriminator training, (2) mathematical expressions of GoFAR specialized to common f-Divergences, and (3) a full pseudocode.

C.1 Discriminator Training

Training the discriminator 7 in practice requires choosing $p(s;g)$. For simplicity, we set $p(s;g)$ to be the Dirac distribution centered at g: $\mathbb{I}(\phi(s) = g)$; this precludes having to choose hyperparameters for $p(s;g)$.

Once the discriminator has converged, we can retrieve the reward function $R(s;g) = \log\frac{p(s;g)}{d^O(s;g)} = -\log\left(\frac{1}{c^*(s;g)} - 1\right)$, since $c^*(s;g) = \frac{d^O(s;g)}{p(s;g)+d^O(s;g)}$.

C.2 GoFAR with common f-Divergences

GoFAR requires choosing a f-divergence. Here, we specialize GoFAR to χ^2-divergence as well as KL-divergence as examples. Our practical implementation uses χ^2-divergence, which we found to be significantly more stable than KL-divergence (see Section 5.1).

Example 1 (GoFAR with χ^2-divergence)**.** $f(x) = \frac{1}{2}(x-1)^2$, and we can show that $f_\star(x) = \frac{1}{2}(x+1)^2$ and $f'_\star(x) = x+1$. Hence, the GoFAR objective amounts to

$$\min_{V(s;g)\geq 0}(1-\gamma)\mathbb{E}_{(s,g)\sim(\mu_0,p(g))}[V(s;g)] + \frac{1}{2}\mathbb{E}_{(s,a,g)\sim d^O}\left[\left(R(s;g) + \gamma\mathcal{T}V(s,a;g) - V(s;g) + 1\right)^2\right] \tag{45}$$

and

$$d^*(s,a;g) = d^O(s,a;g)\max\left(0, R(s,a;g) + \gamma\mathcal{T}V^*(s,a;g) - V^*(s;g) + 1\right) \tag{46}$$

Example 2 (GoFAR with KL-divergence)**.** We have $f(x) = x\log x$ and that $\mathrm{D}_{\star,f}(y) = \log\mathbb{E}_{x\sim q}[\exp y(x)]$ [4]. Hence, the KL-divergence GoFAR objective is

$$\min_{V(s;g)\geq 0}(1-\gamma)\mathbb{E}_{(s,g)\sim(\mu_0,p(g))}[V(s;g)] + \log\mathbb{E}_{(s,a,g)\sim d^O}\left[\exp\left(R(s;g) + \gamma\mathcal{T}V(s,a;g) - V(s;g)\right)\right] \tag{47}$$

and

$$d^*(s,a;g) = d^O(s,a;g)\mathrm{softmax}\left(R(s;g) + \gamma\mathcal{T}V^*(s,a;g) - V^*(s;g)\right) \tag{48}$$

Now, we provide the full pseudocode for GoFAR implemented using χ^2-divergence in Algorithm 2.

C.3 Full Pseudocode

Algorithm 2 GoFAR for Continuous MDPs

Require: Offline dataset d^O, choice of f-divergence f, choice of $p(s; g)$
Randomly initialize discriminator c_ψ, value function V_θ, and policy π_ϕ.
`// Train Discriminator (Optional)`
for number of discriminator iterations **do**
  Sample minibatch $\{s_d^i, g^i\}_{i=1}^N \sim d^O$
  Sample $\{s_g^j\}_{j=1}^M \sim p(s; g^i) \quad \forall i \in 1 \dots N$
  Discriminator objective: $\mathcal{L}_c(\psi) = \frac{1}{N}\sum_{i=1}^N [\log(1 - c_\psi(s_d^j, g^i)) + \frac{1}{M}\sum_{j=1}^M [\log c_\psi(s_g^j, g^i)]]$
  Update c_ψ using SGD: $c_\psi \leftarrow c_\psi - \alpha_c \nabla \mathcal{L}_c(\psi)$
end for
`// Train Dual Value Function`
for number of value iterations **do**
  Sample minibatch of offline data $\{s_t^i, a_t^i, s_{t+1}^i, g_t^i\}_{i=1}^N \sim d^O, \{s_0^i\}_{i=1}^M \sim \mu_0, \{g_0^i\}_{i=1}^M \sim d^O$
  If discriminator, obtain reward: $R(s_t^i; g_t^i) = -\log\left(\frac{1}{c_\psi(s_t^i; g_t^i)} - 1\right) \quad \forall i = 1 \dots N$
  If no discriminator, obtain reward: $\{R(s_t^i; g_t^i)\}_{i=1}^N \sim d^O$
  Value objective: $\mathcal{L}_V(\theta) = \frac{1-\gamma}{M}\sum_{i=1}^M [V_\theta(s_0^i; g_0^i)] + \frac{1}{N}\sum_{i=1}^N \left[f_\star(R_t^i + \gamma V(s_{t+1}^i; g_t^i) - V(s_t^i; g_t^i))\right]$
  Update V_θ using SGD: $V_\theta \leftarrow V_\theta - \alpha_V \nabla \mathcal{L}_V(\theta)$
end for
`// Train Policy With` f`-Advantage Regression`
for number of policy iterations **do**
  Sample minibatch of offline data $\{s_t^i, a_t^i, s_{t+1}^i, g_t^i\}_{i=1}^N \sim d^O$
  If discriminator, obtain reward: $R(s_t^i; g_t^i) = -\log\left(\frac{1}{c_\psi(s_t^i; g_t^i)} - 1\right) \quad \forall i = 1 \dots N$
  If no discriminator, obtain reward: $\{R(s_t^i; g_t^i)\}_{i=1}^N \sim d^O$
  Policy objective: $\mathcal{L}_\pi(\phi) = \sum_{i=1}^N \left[\left(f_\star'\left(R_t^i + \gamma V_\theta(s_{t+1}^i; g_t^i) - V_\theta(s_t^i; g_t^i)\right)\right) \log \pi(a \mid s, g)\right]$
  Update π_ϕ using SGD: $\pi_\phi \leftarrow \pi_\phi - \alpha_\pi \nabla \mathcal{L}(\phi)$
end for

D GoFAR for Tabular MDPs

In Section 4.1 of the main text, we have stated that in tabular MDPs, GoFAR's optimal dual value function (10) admits closed-form solution when we choose χ^2-divergence. Here, we provide a derivation of this result.

Recall the dual problem (1)

$$\min_{V(s;g)\geq 0} (1-\gamma)\mathbb{E}_{(s,g)\sim(\mu_0,p(g))}[V(s;g)] + \frac{1}{2}\mathbb{E}_{(s,a,g)\sim d^O}\left[(R(s;g) + \gamma \mathcal{T}V(s,a;g) - V(s;g) + 1)^2\right] \tag{49}$$

To derive a closed-form solution, we rewrite the problem in vectorized notation; we borrow our notations from [32]. We first augment the state-space by concatenating the state dimensions and the goal dimensions so that the new state space $\tilde{S}$ has dimension $S + G$. Then, the new transition function, with slight abuse of notation, $T((s', g') \mid (s, g), a) = T(s \mid s, a)\mathbb{I}(g' = g)$; the new initial state distribution is thus $\mu_0(s, g) = \mu_0(s)p(g)$. Therefore, $\tilde{T} \in \mathbb{R}_+^{|S||G||A|\times|S||G|}$ and $\mu_0 \in \mathbb{R}_+^{|S||G|}$.

We assume that the offline dataset $\mathcal{D}^O$ is collected by a behavior policy π_b. We construct a surrogate MDP $\hat{\mathcal{M}}$ using maximum likelihood estimation; that is, $\hat{T}((s', g') \mid (s, g), a) = \frac{n(s,a,s')}{n(s,a)}\mathbb{I}(g' = g)$, and we impute $\hat{T}((s', g) \mid (s, g), a) = \frac{1}{S}$ when $n(s, a) = 0$. Then, using $\hat{M}$, we can compute $d^O \in \mathbb{R}_+^{|S||G||A|}$ using linear programming and define reward $r \in \mathbb{R}_+^{|S||G|}$ as $r(s; g) = \log \frac{p(s;g)}{d^O(s;g)}$, where $p(s; g) \in \mathbb{R}_+^{|S||G|}$. Now, define $\mathcal{T} \in \mathbb{R}^{|S||G||A|\times|S||G|}$ such that $(\mathcal{T}V)(s, a; g) = \sum s' T((s', g) \mid (s, g), a)V(s'; g)$, where $V \in \mathbb{R}_+^{|S||G|}$ is the dual optimization variables. We also define $\mathcal{B} \in \mathbb{R}^{|S||G||A|\times|S||G|}$ such that $(\mathcal{B}V)(s, a; g) = V(s; g)$. Finally, we define

$D = \text{diag}(d^O) \in \mathbb{R}^{|S||G||A|\times|S||G||A|}$. Now, we can rewrite the dual problem as follow:

$$\begin{aligned}
&\min_{V(s;g)\geq 0}(1-\gamma)\mathbb{E}_{(s,g)\sim(\mu_0,p(g))}[V(s;g)] + \frac{1}{2}\mathbb{E}_{(s,a,g)\sim d^O}\left[(R(s;g)+\gamma\mathcal{T}V(s,a;g)-V(s;g)+1)^2\right] \\
&\Rightarrow \min_{V(s;g)}(1-\gamma)\mu_0^\top V + \frac{1}{2}\mathbb{E}_{(s,a,G)\sim d^O}\left[\left(\underbrace{\mathcal{B}R(s,a;g)+\gamma\mathcal{T}V(s,a;g)-\mathcal{B}V(s,a;g)}_{R_V(s,a;g)}+1\right)^2\right] \\
&\Rightarrow \min_{V(s;g)}(1-\gamma)\mu_0^\top V + \frac{1}{2}(R_V+I)^\top D(R_V+I)
\end{aligned} \tag{50}$$

Now, we recognize that (50) is equivalent to Equation 49 in [32], as we have reduced goal-conditioned RL to regular RL with an augmented state-space. Now, using the same derivation as in [32], we have that

$$V^* = \left((\gamma\mathcal{T}-\mathcal{B})^\top D(\gamma\mathcal{T}-\mathcal{B})\right)^{-1}\left((\gamma-1)\mu_0 + (\mathcal{B}-\gamma\mathcal{T})^\top D(I+BR)\right) \tag{51}$$

and we can recover $d^*(s,a;g)$:

$$d^*(s,a;g) = d^O(s,a;g)\left(\mathcal{B}R(s,a;g)+\gamma\mathcal{T}V^*(s,a;g)-\mathcal{B}V^*(s,a;g)+1\right) \tag{52}$$

Given d^*, we may extract the optimal policy π^* by marginalizing over actions:

$$\pi^*(a \mid s,g) = \frac{d^*(s,a;g)}{\sum_a d^*(s,a';g)} = \frac{d^O(s,a;g)\left(R(s;g)+\gamma\mathcal{T}V(s,a;g)-V(s;g)\right)}{\sum_a d^O(s,a;g)\left(R(s;g)+\gamma\mathcal{T}V(s,a;g)-V(s;g)\right)} \tag{53}$$

E Additional Technical Discussion

E.1 Connecting Goal-Conditioned State-Matching and Probabilistic GCRL

Suppose the GCRL problem comes with a reward function $r(s;g)$. We also show that there is an equivalent goal-conditioned state-occupancy matching problem with a target distribution $p(s;g)$ defined based on $r(s;g)$.

Proposition E.1. *(Proposition 4.1 in the paper) Given any $r(s;g)$, for each g in the support of $p(g)$, define $p(s;g) = \frac{e^{r(s;g)}}{Z(g)}$, where $Z(g) := \int e^{r(s;g)}ds$ is the normalizing constant. Then, the following equality holds:*

$$-\mathrm{D}_{\mathrm{KL}}(d^\pi(s;g)\|p(s;g)) + C = (1-\gamma)J(\pi) + \mathcal{H}(d^\pi(s;g)) \tag{54}$$

where $J(\pi)$ is the GCRL objective (Eq. (2)) with reward $r(s;g)$ and $C := \mathbb{E}_{g\sim p(g)}[\log Z(g)]$ is a constant.

Proof. We have that

$$\begin{aligned}
&(1-\gamma)J(\pi) \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[r(s;g)\right] \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[\log e^{r(s;g)}\right] \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[\log \frac{e^{r(s;g)}Z(g)}{Z(g)}\right] \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[\log \frac{e^{r(s;g)}}{Z(g)}\right] + \mathbb{E}_{g\sim p(g)}[\log Z(g)] \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[\log \frac{e^{r(s;g)}}{Z(g)}\cdot\frac{d^\pi(s;g)}{d^\pi(s;g)}\right] + C \\
=&\mathbb{E}_{g\sim p(g)}\mathbb{E}_{s\sim d^\pi(s;g)}\left[\log \frac{p(s;g)}{d^\pi(s;g)}\right] + \mathbb{E}_{g\sim p(g)}\mathbb{E}_{d^\pi(s;g)}[\log d^\pi(s;g)] + C \\
=&\mathbb{E}_{g\sim p(g)}\left[-\mathrm{D}_{\mathrm{KL}}(d^\pi(s;g)\|p(s;g)) - \mathcal{H}(d^\pi(s;g))\right] + C
\end{aligned} \tag{55}$$

Rearranging the inequality gives the desired result. □

A constant term C appear in the equality to account for the need for normalizing $e^{r(s;g)}$ to make it a proper distribution. This, however, does not change the optimal solution for the goal-conditioned state-occupancy matching objective. Therefore, we have shown that for any choice of reward $r(s;g)$, solving the GCRL problem with a maximum state-entropy regularization is equivalent to optimizing for the goal-conditioned state-occupancy matching objective with target distribution $p(s;g) := \frac{e^{r(s;g)}}{Z(g)}$.

E.2 Optimality Conditions for Hindsight Relabeling

In section 4.2, we have stated that HER is not optimal for most choices of reward functions. In this section, we investigate conditions under which hindsight relabeling methods such as HER would be optimal.

Let the goal-relabeling distribution for HER be $p_{\text{HER}}(g \mid s, a)$; we do not specify the functional form of $p_{\text{HER}}(g \mid s, a)$ for generality (see 26). Then, in order for this distribution to be optimal, then it must satisfy

$$p_{\text{HER}}(g \mid s, a) = p(g \mid s, a)(f'_\star(R(s;g) + \gamma \mathcal{T} V^*(s, a; g) - V^*(s;g)) \tag{56}$$

Then, the choice of $r(s;g)$ such that this equality holds is the reward function for which HER would be optimal. However, solving for $r(s;g)$ is generally challenging and we leave it to future work for investigating whether doing so is possible for general f-divergence coupled with neural networks.

This optimality condition is related to a prior work [10], which has found that hindsight relabeling is optimal in the sense of maximum-entropy inverse RL [50] for a peculiar choice of reward function (see Equation 9 in [10]), which cannot be implemented in practice. Our result is more general as it applies to any choice of f-divergence, and is not restricted to the form of maximum-entropy inverse RL.

E.3 Theoretical Comparison to Prior Regression-based GCRL methods

In section 4.3, we have stated that GoFAR's theoretical guarantee (Theorem 4.1) is stronger in nature compared to prior regression-based GCRL methods. Here, we provide an in-depth discussion.

Both GCSL [14] and WGCSL [47] prove that their objectives are lower bounds of the true RL objective (Theorem 3.1 in [14] and Theorem 1 in [47], respectively); however, in both works, the lower bounds are loose due to constant terms that do not depend on the policy and hence do not vanish to zero. In contrast, GoFAR's objective (6) is, by construction, a lower bound on the RL objective, as it simply incorporates a f-divergence regularization. If the offline data d^O is *on-policy*, then our lower bound is an equality. In contrast, even with on-policy data, the lower bounds in both GCSL and WGCSL are still loose due to the unavoidable constant terms.

GCSL also proves a sub-optimality guarantee (Theorem 3.2 in [14]) under the assumption of full state-space coverage. Though full state-space coverage has been considered in some prior offline RL works [24, 31], it is much stronger than the concentrability assumption in our Theorem 4.1, which only applies to d^*. Furthermore, this guarantee is not statistical in nature, and instead directly makes a strong assumption on the *maximum* total-variance distance between π and optimal π^* for the GCSL objective, which is difficult to verify in practice. In contrast, our bound suggests asymptotic optimality: given enough offline data, the solution to GoFAR's policy objective will converge to π^*. Finally, WGCSL proves a policy improvement guarantee (Proposition 1 in [47]) under their exponentially weighted advantage; the improvement is not a strict equality, and consequently there is no convergence guarantee to the optimal policy. Furthermore, this result is not directly dependent on their use of an advantage function, so it is not clear the precise role of their advantage function in their algorithm.

F Task Descriptions

In this section, we describe the tasks in our experiments in Section 5.

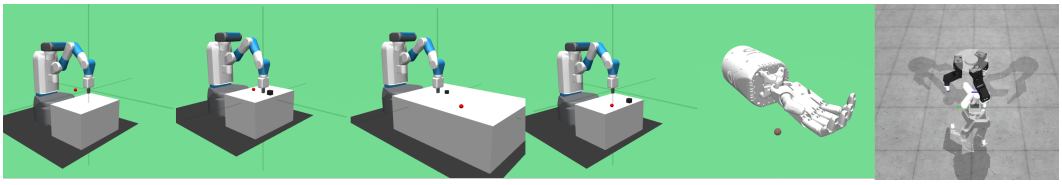


Figure 7: Tasks from left to right: FetchReach, FetchPush, FetchSlide, FetchPick, HandReach, D'Claw (Simulation)

F.1 Fetch Tasks

The Fetch environments involve the Fetch robot with the following specifications, developed by Plappert et al. [39].

- Seven degrees of freedom
- Two pronged parallel gripper
- Three dimensional goal representing Cartesian coordinates of target
- Sparse, binary reward signal: 0 when at goal with tolerance 5cm and -1 otherwisejk
- 25 Hertz simulation frequency
- Four dimensional action space
  - Three Cartesian dimensions
  - One dimension to control gripper

F.1.1 Fetch Reach

The task is to place the end effector at the target goal position. Observations consist of the end effector's positional state, whether the gripper is closed, and the end effector's velocity. The reward is given by:

$$r(s, a, g) = 1 - \mathbb{1}(\|s_{xyz,\text{ee}} - g_{xyz}\|_2 \leq 0.05)$$

F.1.2 Fetch Push

The task is to push an object to the target goal position. Observations consist of the end effector's position, velocity, and gripper state as well as the object's position, rotational orientation, linear velocity, and angular velocity. The reward is given by:

$$r(s, a, g) = 1 - \mathbb{1}(\|s_{xyz,\text{obj}} - g_{xyz}\|_2 \leq 0.05)$$

F.1.3 Fetch Slide

In this task, the goal position lies outside of the robot's reach and the robot must slide the puck-like object across the table to the goal. Observations consist of the end effector's position, velocity, and gripper state as well as the object's position, rotational orientation, linear velocity, and angular velocity. The reward is given by:

$$r(s, a, g) = 1 - \mathbb{1}(\|s_{xyz,\text{obj}} - g_{xyz}\|_2 \leq 0.05)$$

F.1.4 Fetch Pick

The task is to grasp the object and hold it at the goal, which could be on or above the table. Observations consist of the end effector's position, velocity, and gripper state as well as the object's position, rotational orientation, linear velocity, and angular velocity. The reward is given by:

$$r(s, a, g) = 1 - \mathbb{1}(\|s_{xyz,\text{obj}} - g_{xyz}\|_2 \leq 0.05)$$

F.2 Hand Reach

Uses a 24 DoF robot hand with a 20 dimensional action space. Observations consist of each of the 24 joints' positions and velocities. The goal space is 15 dimensional corresponding to the positions of each of its five fingers. The goal is achieved when the mean distance of the fingers to their goals is less than 1cm. The reward is binary and sparse: 0 if the goal is reached and -1 otherwise, i.e.

$$r(s, a, g) = 1 - \mathbb{1}\left(\frac{1}{5}\sum_{i=1}^{5} \|s_i - g_i\|_2 \leq 0.01\right)$$

F.3 D'ClawTurn (Simulation)

First introduced by Ahn et al. [2], the D'Claw environment has a 9 DoF three-fingered robotic hand. The turn task consists of turning the valve to a desired angle. The initial angle is randomly chosen from $[-\frac{\pi}{3}, \frac{\pi}{3}]$; the target angle is randomly chosen from $[-\frac{2*\pi}{3}, \frac{2*\pi}{3}]$. The observation space is 21D, consisting of the current joint angles θ_t, their velocities $\dot{\boldsymbol{\theta}}$, angle between current and goal angle, and the previous action. The environment terminates after 80 steps. The reward function is defined as:

$$r = \mathbb{1}\left(\left|\text{arctan2}\left(\frac{s_{y,obj}}{s_{x,obj}}\right) - \text{arctan2}\left(\frac{g_{y,obj}}{g_{x,obj}}\right)\right| \leq 0.1\right)$$

F.4 D'ClawTurn (Real)

To make real-world data collection easier, we slightly modify the initial and target angle distributions. The initial angle is randomly chosen from $[-\frac{\pi}{3}, \frac{\pi}{3}]$; the target angle is randomly chosen from $[-\frac{\pi}{2}, \frac{\pi}{2}]$. Using this task distribution, collecting 400K transitions with random actions takes about 15 hours. In Figure 8, we also include a larger picture of the robot platform.

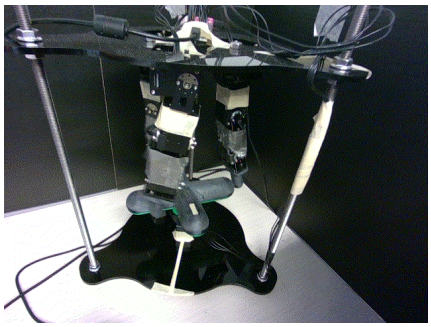


Figure 8: The D'Claw tri-finger platform.

G Experimental Details

In this section, we provide experimental details omitted in Section 5 of the main text. These include (1) technical details of the baseline methods, (2) hyperparameter and architecture details for all methods, (3) offline GCRL dataset details, and finally, (4) experimental details of the zero-shot transfer experiment.

G.1 Baseline Implementation Details

DDPG. We use an open-source implementation of DDPG, which has already tuned DDPG on the set of Fetch tasks. We implement all other methods on top of this implementation, keeping identical architectures and hyperparameters when appropriate. The critic objective is

$$\min_Q \mathbb{E}_{(s_t,a_t,s_{t+1},g)\sim d^{\bar{\mathcal{O}}}}[(r(s_t, g) + \gamma\bar{Q}(s_{t+1}, \pi(s_{t+1}, g), g) - Q(s_t, a_t, g))^2] \tag{57}$$

where $\bar{Q}$ denotes the stop-gradient operation. The policy objective is

$$\min_\pi -\mathbb{E}_{(s_t,a_t,s_{t+1},g)\sim d^{\bar{\mathcal{O}}}}[Q(s_t, \pi(s_t, g), g)] \tag{58}$$

DDPG updates the critic and the policy in an alternating fashion.

ActionableModel. We implement AM on top of DDPG. Specifically, we add a CQL loss in the critic update:

$$\mathbb{E}_{(s,g)\sim d^{\tilde{O}},a\sim\exp(Q)}[Q(s,a,g)] \tag{59}$$

where $d^{\tilde{O}}$ is the distribution of the relabelled dataset. In practice, we sample 10 random actions from the action-space to approximate this expectation. Furthermore, we implement goal-chaining, where for half of the relabeled transitions in each minibatch update, the relabelled goals are randomly sampled from the offline dataset. We found goal-chaining to not be stable in some environments, in particular, FetchPush, FetchPickAndPlace, and FetchSlide. Therefore, to obtain better results, we remove goal-chaining for these environments in our experiments.

GCSL. We implement GCSL by removing the DDPG critic component and changing the policy loss to maximum likelihood:

$$\min_{\pi} -\mathbb{E}_{(s,a,g)\sim d^{\tilde{O}}}\left[\log \pi(a \mid s,g)\right] \tag{60}$$

WGCSL. We implement WGCSL on top of GCSL by including a Q-function. The Q-function is trained using TD error as in DDPG and provided an advantage weighting in the regression loss. The advantage term we compute is $A(s_t,a_t,g) = r(s_t;g) + \gamma Q(s_{t+1},\pi(s_{t+1},g);g) - Q(s_t,a_t;g)$. Using this, the WGCSL policy objective is

$$\min_{\pi} -\mathbb{E}_{(s_t,a_t,\phi(s_i))\sim d^{\tilde{O}}}\left[\gamma^{i-t}\exp_{clip}(A(s_t,a_t,\phi(s_i)))\log \pi(a_t \mid s_t,\phi(s_i))\right] \tag{61}$$

where we clip $\exp(\cdot)$ for numerical stability. The original WGCSL uses different HER rates for the critic and the actor training. To make the implementation simple and consistent with all other approaches, we use the same HER rate for both components. We note that the original WGCSL computes the advantage term slightly differently as $A(s_t,a_t,g) = r(s_t;g) + \gamma Q(s_{t+1},\pi(s_{t+1},g);g) - Q(s_t,\pi(s_t,g);g)$; this version of WGCSL[3] is incorporated in our open-sourced code.

With the exception of AM, all baselines set the goal-relabeling distribution $d^{\tilde{O}}$ to be the uniform distribution over future states in the same trajectory (See Equation (26)).

G.2 Architectures and Hyperparameters

Each algorithm uses their own set of fixed hyperparameters for all tasks. WGCSL, GCSL, and DDPG are already tuned on our set of tasks [39, 47], so we use the reported values from prior works; AM, in our implementation, shares same networks as DDPG, so we use DDPG's values. For GoFAR, we use identical hyperparameters as WGCSL because they share similar network components; GoFAR additionally trains a discriminator, for which we use the same architecture and learning rate as the value network. We impose a small discriminator gradient penalty [16] to prevent overfitting. For all experiments, We train each method for 3 seeds, and each training run uses 400k minibatch updates of size 512. The architectures and hyperparameters for all methods are reported in Table 3.

G.3 Offline GCRL Experiments

Datasets. For each environment, the offline dataset composition is determined by whether data collected by random actions provides sufficient coverage of the desired goal distribution. For FetchReach and D'ClawTurn, we find this to be the case and choose the offline dataset to be 1 million random transitions. For the other four tasks, random data does not capture meaningful goals, so we create a mixture dataset with 100K transitions from a trained DDPG-HER agent and 900K random transitions; the transitions are not labeled with their sources. This mixture setup has been considered in prior works [21, 32] and is reminiscent of real-world datasets, where only a small portion of the dataset is task-relevant but all transitions provide useful information about the environment.

G.4 Zero-Shot Transfer Experiments

We use GoFAR (Binary) variant for trainning the GoFAR planner. The low-level controller is trained using an online DDPG algorithm on a narrow goal distribution, set to be closed to the object's initial positions.

[3] We thank Joey Hejna for pointing out this difference in an email correspondence.

Table 3: Offline GCRL Hyperparameters.

	Hyperparameter	Value
Hyperparameters	Optimizer	Adam [22]
	Critic learning rate	5e-4 (1e-3 for AM/DDPG)
	Actor learning rate	5e-4 (1e-3 for AM/DDPG)
	Discriminator learning rate	5e-4
	Discriminator gradient penalty	0.01
	Mini-batch size	256
	Discount factor	0.98
Architecture	Discriminator hidden dim	256
	Discriminator hidden layers	2
	Discriminator activation function	ReLU
	Critic (resp. Value) hidden dim	256
	Critic (resp. Value) hidden layers	2
	Critic (resp.Value) activation function	ReLU
	Actor hidden dim	256
	Actor hidden layers	2
	Actor activation function	ReLU

GoFAR Hierarchical Controller operates by first generating a sequence of subgoals $(g_1, ..., g_T)$ using π_{high} by recursively feeding the newest generated goal and conditioning on the final goal g. Then, at each time step t, the low-level controller executes action $\pi_{\text{low}}(a_t \mid s_t, g_t)$. The high-level subgoals are not re-planned during low-level controller execution. We note that this is a simple planning algorithm, and improvement in performance can be expected by considering more sophisticated planning approaches.

H Additional Results

H.1 Offline GCRL Full Results

In this section, we provide the full results table for discounted return, final distance, and success rate metrics, including error bars over 10 random seeds. The number inside the parenthesis indicates the best HER rate for the baseline methods on the task. Star ($\star$) indicate statistically significant improvement over the second best-performing method under a 2-sample t-test.

Table 4: Discounted Return on offline GCRL tasks, averaged over 10 random seeds.

Task	**Supervised Learning**			**Actor-Critic**	
	GoFAR (Ours)	**WGCSL**	**GCSL**	**AM**	**DDPG**
FetchReach	28.2 ± 0.61	21.9± 2.13 (1.0)	20.91 ± 2.78 (1.0)	**30.1** ± 0.32 (0.5)	29.8 ± 0.59 (0.2)
FetchPick	**19.7** ± 2.57	9.84 ± 2.58 (1.0)	8.94 ± 3.09 (1.0)	18.4 ± 3.51 (0.5)	16.8 ± 3.10 (0.5)
FetchPush ($\star$)	**18.2** ± 3.00	14.7 ± 2.65 (1.0)	13.4 ± 3.02 (1.0)	14.0 ± 2.81 (0.5)	12.5 ± 4.93 (0.5)
FetchSlide	2.47 ± 1.44	**2.73** ± 1.64 (1.0)	1.75 ± 1.3(1.0)	1.46 ± 1.38 (0.5)	1.08 ± 1.35 (0.5)
HandReach ($\star$)	**11.5** ± 5.26	5.97 ± 4.81 (1.0)	1.37 ± 2.21 (1.0)	0. ± 0.0 (0.5)	0.81 ± 1.73 (0.5)
D'ClawTurn ($\star$)	**9.34** ± 3.15	0.0 ± 0.0 (1.0)	0.0 ± 0.0 (1.0)	2.82± 1.71 (1.0)	0.0± 0.0 (0.2)
Average Rank	**1.5**	3	4.17	2.83	4

Table 5: Final Distance on offline GCRL tasks, averaged over 10 random seeds.

Task	**Supervised Learning**			**Actor-Critic**	
	GoFAR (Ours)	**WGCSL**	**GCSL**	**AM**	**DDPG**
FetchReach	0.018 ± 0.003	0.007 ± 0.0043(1.0)	0.008 ± 0.008(1.0)	**0.007**± 0.001 (0.5)	0.041 ± 0.005 (0.2)
FetchPickAndPlace	**0.036** ± 0.013	0.094 ± 0.043(1.0)	0.108 ± 0.060(1.0)	0.040 ± 0.020(0.5)	0.043 ± 0.021(0.5)
FetchPush	**0.033** ± 0.008	0.041 ± 0.020(1.0)	0.042 ± 0.018 (1.0)	0.070 ± 0.039(0.5)	0.060 ± 0.026 (0.5)
FetchSlide ($\star$)	**0.120** ± 0.02	0.173 ± 0.04(1.0)	0.204 ± 0.051 (1.0)	0.198 ± 0.059 (0.5)	0.353 ± 0.248 (0.5)
HandReach ($\star$)	**0.024** ± 0.009	0.035 ± 0.012 (1.0)	0.038 ± 0.013(1.0)	0.037 ± 0.004(0.5)	0.038 ± 0.013 (0.5)
D'ClawTurn ($\star$)	**0.92** ± 0.28	1.49 ± 0.26 (1.0)	1.54 ± 0.15 (1.0)	1.28 ± 0.26 (1.0)	1.54 ± 0.13 (0.2)
Average Rank	**1.5**	2.33	4.25	2.67	4.5

Table 6: Success Rate on offline GCRL tasks, averaged over 10 random seeds.

Task	Supervised Learning			Actor-Critic	
	GoFAR (Ours)	**WGCSL**	**GCSL**	**AM**	**DDPG**
FetchReach	**1.0** ± 0.0	0.99 ± 0.01 (1.0)	0.98 ± 0.05 (1.0)	**1.0** ± 0.0 (0.5)	0.99 ± 0.02 (0.2)
FetchPickAndPlace	**0.84** ± 0.12	0.54 ± 0.16 (1.0)	0.54 ± 0.20(1.0)	0.78 ± 0.15(0.5)	0.81 ± 0.13(0.5)
FetchPush (⋆)	**0.88** ± 0.09	0.76 ± 0.12(1.0)	0.72 ± 0.15(1.0)	0.67 ± 0.14 (0.5)	0.65 ± 0.18 (0.5)
FetchSlide	**0.18** ± 0.12	**0.18** ± 0.14(1.0)	0.17 ± 0.13 (1.0)	0.11 ± 0.09 (0.5)	0.08 ± 0.11 (0.5)
HandReach	**0.40** ± 0.20	0.25 ± 0.23 (1.0)	0.047 ± 0.10 (1.0)	0.0 ± 0.0 (0.5)	0.023 ± 0.054 (0.5)
D'ClawTurn (⋆)	**0.26** ± 0.13	0.0 ± 0.0 (1.0)	0.0 ± 0.0 (1.0)	0.13 ± 0.14 (1.0)	0.01 ± 0.02 (0.2)
Average Rank	**1**	3	4	3.33	3.67

H.2 Ablations

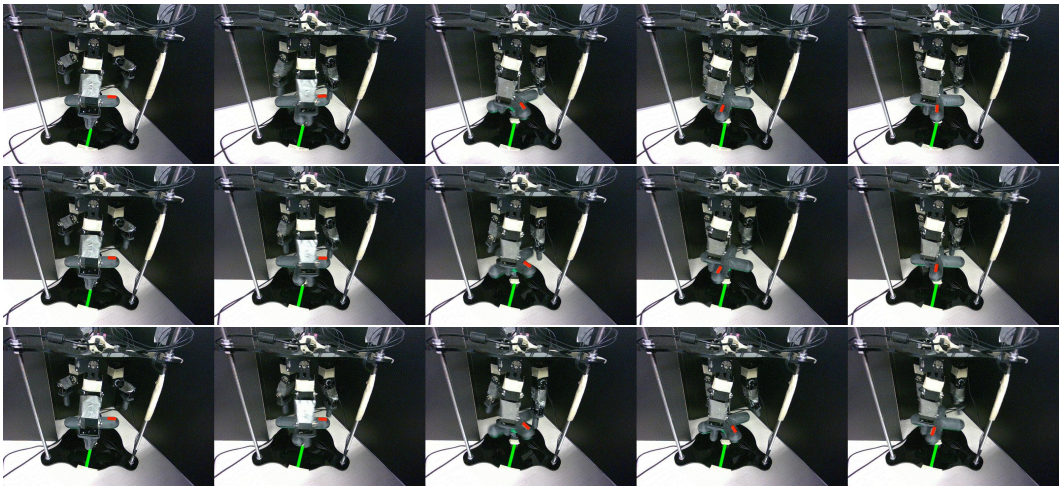
We also include the full task-breakdown table of GoFAR ablations presented in Figure 4 for completeness. As shown in 7, GoFAR and GoFAR (HER) perform comparatively on all tasks. GoFAR (binary) is slightly worse across tasks, and GoFAR (KL) collapses due to the use of an unstable f-divergence.

Table 7: GoFAR Ablation Studies

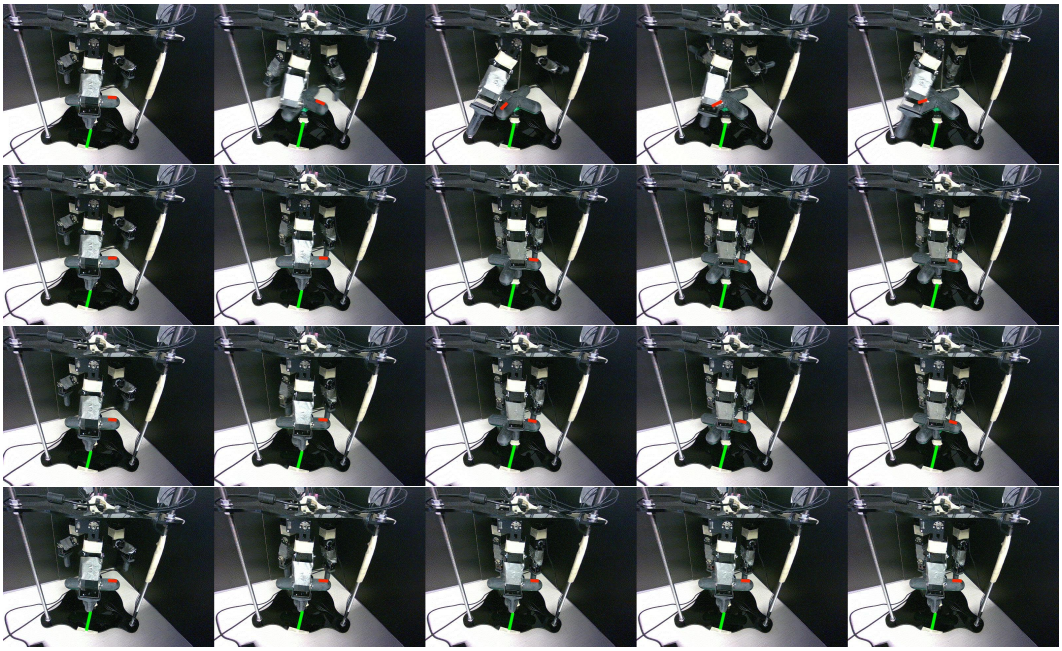
Variants	FetchReach	FetchPickAndPlace	FetchPush	FetchSlide	HandReach	DClawTurn
GoFAR	27.8 ± 0.55	19.5 ± 4.13	18.9 ± 3.87	3.67 ± 0.78	11.9 ± 3.00	9.34 ± 3.15
GoFAR (HER)	28.3± 0.65	19.8± 2.82	20.5± 2.29	3.85± 0.80	8.02± 5.70	10.51 ± 3.51
GoFAR (Binary)	26.1± 1.14	17.4±1.78	17.4 ± 2.67	3.69± 1.75	6.01± 1.62	5.13 ± 4.05
GoFAR (KL)	0±0.0	0±0.0	0±0.0	0±0.0	0± 0.0	0± 0.0

H.3 Real-World Dexterous Manipulations

In our qualitative analysis, we visualize all methods on a specific task instance of turning the valve prong (marked by the red strip) clockwise for 90 degree; the goal location is marked by the green strip. The robot initial pose is randomized. As shown in Figure 9, GoFAR reaches the goal with three random initial poses, whereas all baselines fail. See the figure captions for detail. Policy videos are included in the supplementary material.



(a) GoFAR robustly achieved the goal with three random initial poses; in the first two runs, it demonstrates "recovery" behavior, as the robot would initially overshoot and then turn the valve counterclockwise. In the last run, the robot initially undershoots and then turns again to reach the goal.

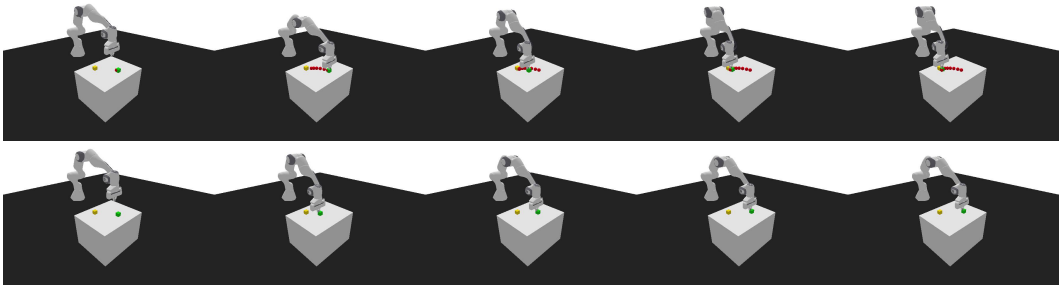


(b) Baselines fail to turn the volve prong (marked by the red strip) to the goal angle (marked by the green strip). AM is the only method that is able to rotate the prong to some degree, though it overshoots in this case and exhibits unnatural behavior.

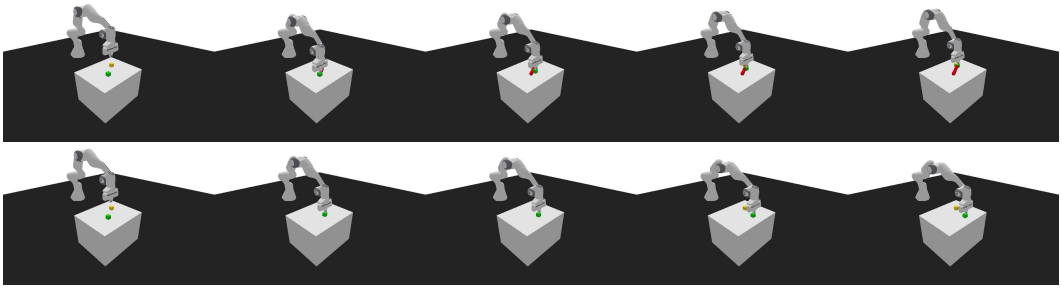
Figure 9: D'ClawTurn policy visualization.

H.4 Zero-Shot Plan Transfer

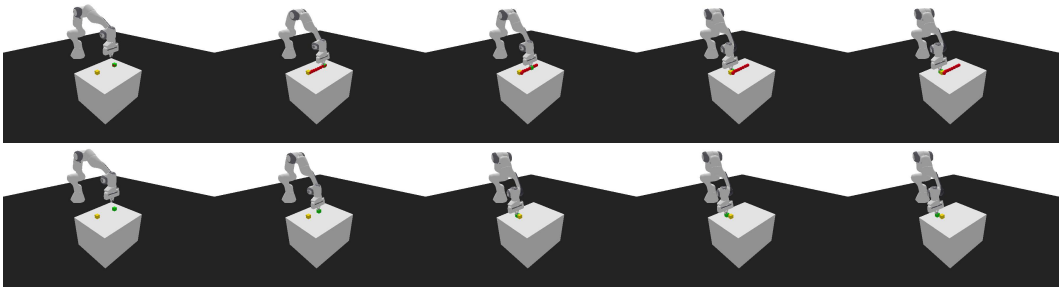
We visualize GoFAR hierarchical controller and the plain low-level controller on three distinct goals in Figure 10. See the figure caption for detail. Policy videos are included in the supplementary material.



(a) Goal 1



(b) Goal 2



(c) Goal 3

Figure 10: Qualitative comparison of GoFAR hierarchical controller (top) vs. plain low-level controller (bottom) on representative goals in the Franka pushing task. Red circles represent intermediate subgoals generated by the GoFAR planner. As shown, the low-level controller only succeeds in Goal 3, whereas the hierarchical controller achieves the distant goals in all three cases.