What is Essential for Unseen Goal Generalization of Offline Goal-conditioned RL?

Rui Yang [1] **Yong Lin** [1] **Xiaoteng Ma** [2] **Hao Hu** [2] **Chongjie Zhang** [2] **Tong Zhang** [1]

Abstract

Offline goal-conditioned RL (GCRL) offers a way to train general-purpose agents from fully offline datasets. In addition to being conservative within the dataset, the generalization ability to achieve unseen goals is another fundamental challenge for offline GCRL. However, to the best of our knowledge, this problem has not been well studied yet. In this paper, we study out-of-distribution (OOD) generalization of offline GCRL both theoretically and empirically to identify factors that are important. In a number of experiments, we observe that weighted imitation learning enjoys better generalization than pessimism-based offline RL method. Based on this insight, we derive a theory for OOD generalization, which characterizes several important design choices. We then propose a new offline GCRL method, **G**eneralizable **O**ffline go**A**l-condi**T**ioned RL (**GOAT**), by combining the findings from our theoretical and empirical studies. On a new benchmark containing 9 independent identically distributed (IID) tasks and 17 OOD tasks, GOAT outperforms current state-of-the-art methods by a large margin.

1. Introduction

Deep reinforcement learning (DRL) makes it possible for a learning agent to achieve superhuman performance on a range of challenging tasks (Silver et al., 2016; 2018; Vinyals et al., 2019; Li et al., 2020b). However, recent studies have found that DRL is prone to overfitting the training tasks and is sensitive to environmental changes (Cobbe et al., 2019; Wang et al., 2020; Han et al., 2021; Kirk et al., 2023). Goal-conditioned reinforcement learning (GCRL) is gaining increasing attention because it enables learning general-purpose decision-making rather than overfitting to a single task (Andrychowicz et al., 2017; Ghosh et al., 2019; Li et al., 2020a). Particularly, offline GCRL (Chebotar et al., 2021; Yang et al., 2022b), which learns as many skills as possible from previously collected datasets without any exploration in the environment, is promising for large-scale and general-purpose pre-training. Nevertheless, prior works (Chebotar et al., 2021; Yang et al., 2022b; Ma et al., 2022b) have largely focused on reaching goals in the dataset, without systematically studying the problem of out-of-distribution (OOD) goal generalization. There are a number of questions: what is the OOD generalization performance of current offline GCRL algorithms? And more importantly, **what is essential for OOD generalization of offline GCRL?**

To answer these questions, we first design a 2D goal-reaching task with different types of offline data. We find that (1) pessimism-based offline RL is restrained from generalizing to OOD goals and (2) imitation learning overfits the data noise and fails to generalize when given non-expert data. On the contrary, (3) weighted imitation learning is a strong baseline for OOD generalization across different types of training data. The observation motivates us to derive a generalization theory from the perspective of domain generalization (Muandet et al., 2013; Zhang et al., 2012; Zhou et al., 2021a). Through analyzing our theory, we find several techniques that are essential to minimize the generalization bound, including advantage re-weighting, data selection, density re-weighting, and goal-relabeling. Particularly, we find re-weighting the training state-goal distribution with the reciprocal of its density can minimize the worst-case distribution shift. Based on these results, we propose, **G**eneralizable **O**ffline go**A**l-condi**T**ioned RL (**GOAT**), by integrating these techniques into a general weighted imitation learning framework, which encourages optimistic goal sampling while still maintaining pessimism on action selection.

Due to the lack of benchmarks for evaluating the OOD generalization performance of offline GCRL, we develop a challenging robot manipulation benchmark based on a robotic arm or an anthropomorphic hand. The benchmark comprises nine offline datasets and 26 evaluation tasks, 9 of which contain independent and identically distributed

[1]The Hong Kong University of Science and Technology [2]Tsinghua University. Correspondence to: Tong Zhang <tongzhang@tongzhang-ml.org>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

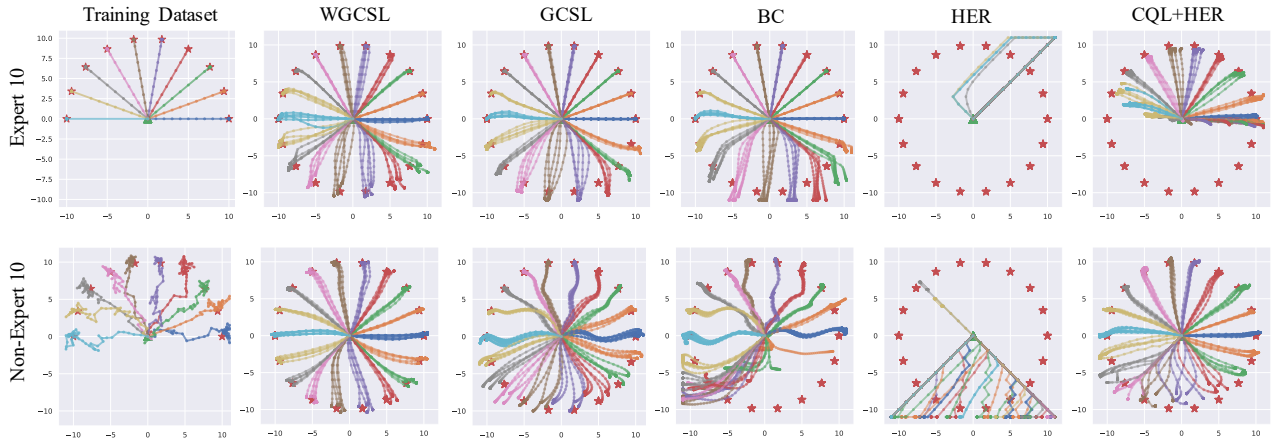


Figure 1. Training datasets and trajectories generated by different agents trained on "Expert 10" and "Non-Expert 10" datasets.

(IID) goals, while the rest 17 tasks involve various types of OOD goals. In our experiments[1], we demonstrate that GOAT considerably improves the OOD generalization performance of existing offline GCRL methods, as well as enhances efficiency in online fine-tuning for unseen goals. Furthermore, we conduct in-depth ablation studies to validate the effectiveness of each component used in GOAT, which may benefit future research on OOD generalization for offline RL.

2. Preliminaries

2.1. Goal-conditioned RL

Goal-conditioned RL (GCRL) considers a goal-augmented Markov Decision Process (GMDP), denoted by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{G}, \mathcal{P}, r, \gamma)$. $\mathcal{S}$, $\mathcal{G}$ $\mathcal{A}$ refer to state, goal, and action spaces, respectively. γ is the discount factor, and $r : \mathcal{S} \times \mathcal{G} \times \mathcal{A} \to \mathbb{R}$ is the goal-conditioned reward function. Generally, we consider a sparse and binary reward function $r(s, a, g) = 1[\|\phi(s) - g\|_2^2 \leq \delta]$, where δ is a threshold and ϕ is a known state-to-goal mapping (Andrychowicz et al., 2017). A policy $\pi : \mathcal{S} \times \mathcal{G} \to \mathcal{A}$ aims to maximize the expected return:

$$J(\pi) = \mathbb{E}_{\substack{g \sim p(g), s_0 \sim \mu(s_0), \\ a_t \sim \pi(\cdot|s_t, g), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)}} \Big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, g) \Big],$$

where $\mu(s_0)$ is the distribution of initial states. The value function is defined as $V^\pi(s, g) = \mathbb{E}_{a_t \sim \pi(\cdot|s_t, g), s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)} \big[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, g) | s_0 = s \big]$. For offline GCRL, the agent cannot interact with the environment during training, and the training data is sampled from a static dataset $D = \{(s_t, a_t, g, r_t, s_{t+1})\}$.

[1]Code is available at https://github.com/YangRui2015/GOAT

2.2. Domain Generalization

Domain Generalization (DG) was first studied in the supervised learning setting (Blanchard et al., 2011). A domain is defined as a joint distribution P_{XY} on $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the label space. DG learns a model from K different training domains $\mathcal{S} = \{(x^{(k)}, y^{(k)})\}_{k=1}^K$ that aims to generalize on unseen testing domains $\mathcal{T} = \{x^{\mathcal{T}}\}, P_{XY}^{\mathcal{T}} \neq P_{XY}^k, k \in \{1, \cdots, K\}$. DG mainly handles covariate shift (Zhou et al., 2021a), assuming that the labeling function $P_{Y|X}$ is stable across domains (Muandet et al., 2013) and only the marginal distribution changes $P_X^{\mathcal{T}} \neq P_X^k, k \in \{1, \cdots, K\}$.

3. OOD Generalization for Offline GCRL

In this section, we first compare different GCRL algorithms in a 2D goal-reaching environment, showing that weighted imitation learning method is preferable to other methods across different data settings. Based on the observations, we formulate the OOD generalization problem as domain generalization, and then derive a theoretical framework to analyze the essential techniques for OOD generalization.

3.1. Didactic Example

We design a 2D point environment as shown in Figure 2(a) to characterize the generalization ability of different offline GCRL algorithms, including BC, GCSL (Ghosh et al., 2019), WGCSL (Yang et al., 2022b), DDPG+HER (Andrychowicz et al., 2017), and CQL+HER (Chebotar et al., 2021). There are three types of training data, namely "Expert N" and "Non-Expert N", where N refers to the number of trajectories in the dataset. In the training datasets, trajectories and goals are mainly distributed on the top semi-circle with a radius of 10. Unlike the training data, the

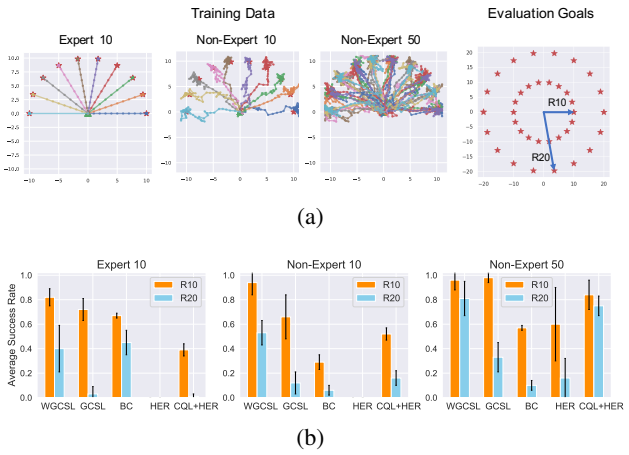


Figure 2. (a) Visualization of three 2D goal-reaching datasets and two groups of evaluation goals. "R10" and "R20" refer to the radius (10 or 20) of the desired goals for evaluation. (b) Average success rates of different agents over 5 random seeds.

evaluation goals are on the full circles of radius 10 and 20. Both states and goals in this environment are represented as 2D coordinates indicating their positions, while actions are 2D vectors of the displacement. In this example, the optimal policy is $\pi(s, g) = \text{clip}(g - s, 0, 1)$, where the maximum movement in one dimension is 1. If the agent learns the optimal policy, it can successfully generalize to any unseen goal.

From the results in Figure 1 and Figure 2(b), we can draw the following conclusions:

- Given a clean expert dataset, BC generalizes well for OOD goals. However, in the case of training with non-expert and noisy data, it can overfit the noise and thus fail to generalize.
- DDPG+HER (short for "HER") suffers from overestimating values of OOD actions. As a result, it avoids in-dataset actions and produces odd trajectories.
- For the pessimism-based approach CQL+HER, its trajectories are restricted to the upper semicircle and fail to generalize to the lower part when given clean expert data. It can only generalize relatively well when the data size and coverage are sufficiently large.
- WGCSL significantly improves the OOD generalization ability over GCSL by re-weighting samples and performs consistently well across different datasets.

The designed task is simple but representative for characterizing the characteristics of different algorithms. More results can be found in Appendix D.1. As suggested by the empirical results, the weighted imitation-based method enjoys better OOD generalization than pessimism-based method. Moreover, pessimism-based offline RL methods are inhibited from reaching OOD area in theory (Jin et al., 2021; Kumar et al., 2021). In contrast, weighted imitation learning method has theoretical guarantees for OOD generalization, which we will show in Section 3.3.

3.2. Problem Formulation

We define $\mathcal{X} = \mathcal{S} \times \mathcal{G}$ as the input space, $\mathcal{Y} = \mathcal{A}$ as the action space. The offline data $D = \{(s_t, a_t, g, r_t, s_{t+1})\}$ is collected by any behavior policy π_b, where $(s_t, g) \sim P_X^{\mathcal{S}}$. In the testing phase, initial states and desired goals can be sampled from any unknown distribution $P_X^{\mathcal{T}}, P_X^{\mathcal{T}} \neq P_X^{\mathcal{S}}$, which is named "OOD distribution" in this paper. We assume the expert policy $\pi_E(a|s, g)$ (or $P_{Y|X}$) is stable with P_X and generalizes well across different state-goal pairs, which is reasonable because OOD generalization is meaningless when π_E cannot generalize. The objective is to minimize the suboptimality on the testing domain $P_X^{\mathcal{T}}$:

$$\text{SubOpt}(\pi_E, \pi) = \mathbb{E}_{(s_0,g)\sim P_X^{\mathcal{T}}}[V^{\pi_E}(s_0, g) - V^{\pi}(s_0, g)] \tag{1}$$

3.3. A Domain Generalization View

By establishing a link between weighted imitation learning and supervised learning, we can analyze the OOD generalization performance according to the domain generalization bound (Ben-David et al., 2010; Zhang et al., 2012; Mansour et al., 2009).

Our following analysis is based on the Total Variation Distance D_{TV} between any two policies π_1 and π_2:

$$D_{\text{TV}}(\pi_1(\cdot|s, g), \pi_2(\cdot|s, g)) = \sup_{B\subset\mathcal{A}} \left| \int_{a\in B} (\pi_1(a|s, g) - \pi_2(a|s, g)) \right|,$$

where B is any measurable subset of the action space $\mathcal{A}$. Denote the discounted occupancy of state as $d_\pi(s|s_0, g) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s|\pi, s_0, g)$. We define the policy discrepancy on any state-goal distribution ρ as:

$$\varepsilon^\rho(\pi_1, \pi_2) = \mathbb{E}_{\substack{(s_0,g)\sim P_X^\rho \\ s\sim d_{\pi_E}(s|s_0,g)}} \left[D_{\text{TV}}\big(\pi_1(\cdot|s, g), \pi_2(\cdot|s, g)\big) \right]$$

Generally, we do not have access to the true expert policy π_E, but we can imitate a surrogate policy $\hat{\pi}_E$ instead. Then, we provide the following OOD generalization theorem.

Theorem 3.1. *Consider finite hypothesis space Π and we minimize the empirical loss function $\hat{\varepsilon}^{\mathcal{S}}$ with m samples. For a policy π and a surrogate expert policy $\hat{\pi}_E$, with probability*

at least $1-\delta$, the following bound holds:

$$\mathrm{SubOpt}(\pi_E, \pi) \leq \frac{2R_{max}}{(1-\gamma)^2}\Bigg[\underbrace{\hat{\varepsilon}^{\mathcal{S}}(\hat{\pi}_E, \pi)}_{\textit{empirical imitation loss}} + \underbrace{\varepsilon^{\mathcal{S}}(\hat{\pi}_E, \pi_E)}_{\textit{expert estimation gap}} + \underbrace{d_1(\mathcal{T}, \mathcal{S})}_{\textit{distribution shift}} + \sqrt{\frac{\log 2|\Pi| + \log\frac{1}{\delta}}{2m}}\Bigg]$$

where $d_1(\cdot,\cdot)$ is the variation divergence defined as follows:

$$d_1(S_1, S_2) = 2\sup_{\mathcal{J}\subset\mathcal{X}}\left|\int_{x\in\mathcal{J}}(P_{S_1}(x) - P_{S_2}(x))\,dx\right|,$$

here $\mathcal{J}$ is any measurable subset of $\mathcal{X}$.

The proof is deferred to Appendix B.2. Theorem 3.1 suggests that the overall OOD generalization suboptimality can be controlled by minimizing the empirical imitation learning loss, the distance between π_E and $\hat{\pi}_E$, and controlling the distribution shift between training and testing domains. We now analyze how to minimize each term in this bound.

Empirical Imitation Loss We can use a weighted behavior policy as the surrogate policy: $\hat{\pi}_E(a|s,g) \propto w(s,a,g)\pi_b(a|s,g)$. According to Pinsker's inequality (Csiszár & Körner, 2011), this loss can be bounded by KL-divergence. Thus, we have

$$\begin{aligned}
&\min_\theta \mathbb{E}_{(s_0,g)\sim P_X^{\mathcal{S}}, s\sim d_{\pi_E}(s|s_0,g)}\left[D_{\mathrm{KL}}\big(\hat{\pi}_E(\cdot|s,g), \pi_\theta(\cdot|s,g)\big)\right] \\
\iff &\max_\theta \mathbb{E}_{(s_0,g)\sim P_X^{\mathcal{S}}, s\sim d_{\pi_E}(s|s_0,g), a\sim\hat{\pi}_E}\left[\log \pi_\theta(a|s,g)\right] \\
\iff &\max_\theta \mathbb{E}_{\substack{(s_0,g)\sim P_X^{\mathcal{S}} \\ s\sim d_{\pi_E}(s|s_0,g), a\sim\pi_b}}\left[\log \pi_\theta(a|s,g)\cdot w(s,a,g)\right]
\end{aligned}$$

Empirically, following (Wang et al., 2018; Nair et al., 2020b) we omit the difference in d_{π_E} and conduct weighted imitation learning on the offline data to minimize this loss.

Expert Estimation Gap Although we do not have access to π_E, we know π_E has the highest expected value. Instead of minimizing the TV distance to π_E, this problem can be reformulated as maximizing the expected value of the surrogate policy $\hat{\pi}_E$. Following (Wang et al., 2018; Peng et al., 2019), advantage re-weighting $\hat{\pi}_E(a|s,g) \propto \pi_b(a|s,g)\exp(\beta\cdot A(s,a,g))$ brings improved expected value over π_b. However, when the behavior policy is multi-modal and the expert policy is deterministic, as often encountered in multi-goal RL, there is a risk of interpolating between modalities, leading to a widened expert estimation gap. A viable solution to this issue is to eliminate samples from inferior modalities, $\hat{\pi}_E(a|s,g) = \pi_b(a|s,g)\exp(A(s,a,g))\cdot 1[A(s,a,g)\geq c]$, which is the Best Advantage Weight introduce by (Yang et al., 2022b). Ideally, we can eliminate all data from other modalities to obtain a minimum expert estimation gap, but the size of the training data decreases as c grows. There is a trade-off of balancing data quality versus quantity when setting c. Note that our analysis considers an oracle advantage function, but in practice, an imprecise estimation of the advantage function can exacerbate the expert estimation gap. Therefore, an improved method for estimating the advantage function is also crucial.

Distribution Shift The distribution shift term is hard to minimize without any information about the testing distribution $\mathcal{T}$. Instead, we consider minimizing the worst-case of this term by re-weighting the training distribution $\mathcal{S}$.

Define a family of possible testing distributions as $\mathcal{Z} := \left\{Z \,\middle|\, \int_x P_Z(x) = 1; 0 \leq P_Z(x) \leq C, \forall x\in\mathcal{X}\right\}$. Here $C > 1/|\mathcal{X}|$ is a universal positive constant. Our goal is to re-weight the training distribution S that can minimize the worst-case distribution shift, i.e., $\sup_{Z\in\mathcal{Z}} d_1(Z,S)$.

Let $\mathcal{S}$ denote the family of distributions that are generated by re-weighting S, i.e., $\mathcal{S} := \{S' | P_{S'}(x) = h(x)P_S(x); h(x) > 0, \forall x\in\mathcal{X}; \int_x P_{S'}(x) = 1\}$.

Let $\bar{S}$ denote the uniform distribution, i.e., $P_{\bar{S}}(x) = 1/|\mathcal{X}|, \forall x\in\mathcal{X}, a.s.$. We denote the subset of $\mathcal{S}$ that contains all "non-uniform" distributions as $\mathcal{S}^-$, i.e.,

$$\begin{aligned}
\mathcal{S}^- := \{&S' | P_{S'}(x) = h(x)P_S(x); h(x) > 0, \forall x\in\mathcal{X}; \\
&\exists \mathcal{J}\subset\mathcal{X}, \int_{x\in\mathcal{J}} P_{S'}(x)dx < |\mathcal{J}|/|\mathcal{X}|; \int_x P_{S'}(x) = 1\}
\end{aligned}$$

Theorem 3.2. *For all $\forall S\in\mathcal{S}^-$, we have*

$$\sup_{Z\in\mathcal{Z}} d_1(Z,S) > \sup_{Z\in\mathcal{Z}} d_1(Z,\bar{S})$$

The proof can be found in Appendix B.3. Theorem 3.2 suggests that we can re-weight the training distribution $\mathcal{S}$ to a uniform distribution to obtain a smaller worst-case distribution shift. To achieve this, we can approximate the reciprocal of density or uncertainty via the kernel density estimator (Zhao et al., 2019; Pitis et al., 2020) or ensemble (Pathak et al., 2019; Bai et al., 2022).

The Last Term Note that the last term in the above bound is dependent on the dataset size m. Therefore, increasing the size of the dataset through augmentation techniques can lead to a more tighter upper bound. This gives justification to use goal relabeling (Andrychowicz et al., 2017; Li et al., 2020a) for offline GCRL. Relabeling goals with achieved goals expands the size of the offline dataset, which enables training agents on more diverse state-goal pairs, subsequently improving an agent's ability to achieve goals in unknown testing distributions.

3.4. A Brief Summary

In this section, we have discussed several useful techniques for OOD generalization from the generalization theory. These techniques include: (1) weighted imitation learning, which minimizes the empirical imitation loss; (2) advantage re-weighting and data selection, which narrow the expert estimation gap; (3) re-weighting with the reciprocal of density, which minimizes the worst-case distribution shift; and (4) goal relabeling, which minimizes the last term related to the dataset size. Based on our analysis, a weighted imitation learning framework that integrates all of these techniques is highly desirable for OOD goal generalization.

4. Algorithm

Motivated by our theoretical insights, we present the GOAT algorithm, which builds upon the weighted imitation learning framework of WGCSL (Yang et al., 2022b). The existing framework already incorporates several techniques beneficial for the generalization bound, including goal relabeling, advantage re-weighting, and data selection. To further minimize the generalization bound, GOAT improves the surrogate expert policy through better value function estimation and minimizes the worst-case distribution shift by re-weighting samples with uncertainty, where the uncertainty is introduced as an alternative to the reciprocal of density.

We denote a trajectory of horizon T in the offline dataset as $D = \{(s_t, a_t, r_t, s_{t+1}, g)\}, t \in [1, T]$. As suggested by our theory, we perform hindsight relabeling (Andrychowicz et al., 2017) to augment the dataset and obtain the relabeled data $D_{relabel} = \{(s_t, a_t, r'_t, s_{t+1}, g')\}, t \in [1, T]$, where $g' = \phi(s_i), r'_t = r(s_t, a_t, \phi(s_i)), i \geq t$. We then perform weighted imitation learning based on $D_{relabel}$.

Weighted Supervised Policy Learning The overall weighted imitation learning framework is as follows:

$$J(\pi_\theta) = \mathbb{E}_{(s,a,g')\sim D_{relabel}}\left[w(s, a, g') \log \pi_\theta(a|s, g')\right], \tag{2}$$

where the weight w contains three parts, i.e., the uncertainty weight(UW), the exponential advantage weight (EAW) and the data selection weight (DSW). Formally, we define

$$w(s, a, g') = u(s, g') \cdot \exp(\beta A(s, a, g')) \cdot \epsilon(A(s, a, g')),$$

where u is the uncertainty weight to replace the density, $A(s, a, g')$ is the advantage function, and $\epsilon(A(s, a, g')) = 1[A(s, a, g') \geq c]$ is the DSW. In DSW, the constant c may be established as the α quantile of advantage values, in recognition of the fact that the best value for α is more consistently applicable across different environments. Additionally, we also discuss an adaptive variant of DSW and we refer the readers to Appendix D.10. In the subsequent section, we mainly focus on how to estimate the advantage function and the uncertainty weight.

Ensemble Value Functions To better estimate the advantage value for both EAW and DSW, we train N randomly initialized value functions. Each of the value function $Q_i(s, a, g), 1 \leq i \leq N$ minimizes the TD loss:

$$\begin{aligned}\mathcal{L}_{TD} =& \mathbb{E}_{(s_t,a_t,r'_t,s_{t+1},g')\sim D_{relabel}}[L_2(r'_t + \\ & \gamma \hat{Q}_i(s_{t+1}, \pi_\theta(s_{t+1}, g'), g') - Q_i(s_t, a_t, g'))].\end{aligned} \tag{3}$$

In Eq (3), $L_2(u) = u^2$ and $\hat{Q}_i$ refers to the target network of Q_i. Although π_θ is regularized to be near the dataset policy, it can still produce OOD actions to affect the value estimation during training. To mitigate this problem, we can replace L_2 with the expectile regression (ER): $L_2^\tau(u) = |\tau - 1(u < 0)|u^2$, where $\tau \in (0, 1)$.

The group of value function is then leveraged to estimate the advantage value and the uncertainty weight. Specifically, we utilize the mean of the Q functions to estimate $V(s, g')$:

$$V(s, g') = \frac{1}{N} \sum_{i=1}^{N} Q_i(s, \pi_\theta(s, g'), g')$$

Then, the advantage value can be estimated by $A(s_t, a_t, g') = r(s_t, a_t, g') + \gamma V(s_{t+1}, g') - V(s_t, g')$.

Uncertainty Estimation Estimating the density of high-dimensional state-goal space is generally challenging. In this work, we utilize uncertainty to replace density as a fact that the bootstrapped uncertainty is approximately proportional to the reciprocal of density in tabular MDP (Bai et al., 2022). The uncertainty is calculated as the standard deviation of value functions:

$$\text{Std}(s, g') = \sqrt{\frac{\sum_{i=1}^{N} \left(Q_i(s, \pi_\theta(s, g'), g') - V(s, g')\right)^2}{N}}$$

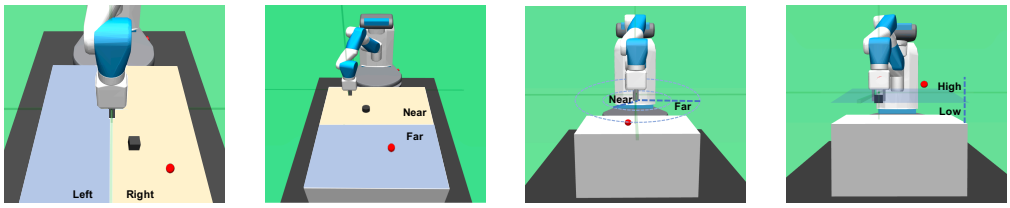
However, the range of $\text{Std}(s, g')$ varies for different environments. To make the uncertainty weight stable, we normalize the standard deviation to $[0, 1]$:

$$\text{Std}_{norm}(s, g') = \frac{\text{Std}(s, g') - \text{Std}_{min}}{\text{Std}_{max} - \text{Std}_{min}},$$

where $\text{Std}_{max}, \text{Std}_{min}$ are the maximum and minimum values of $\text{Std}(s, g')$ stored in a First In First Out (FIFO) queue. Finally, we transform $\text{Std}_{norm}(s, g')$ to reduce more weight for data with lower variance and define the uncertainty weight $u(s, g')$ as:

$$u(s, g') = \text{clip}(\tanh(\text{Std}_{norm}(s, g') \times w) + w_{min}, 0, 1) \tag{4}$$

where w_{min} is set to 0.5. Intuitively, w is the hyperparameter to adjust the proportion of ranked samples to down-weight, i.e., the smaller w is, the more data will be down-weighted, and vice versa.



(a) (b) (c) (d)

Figure 3. Examples of designed benchmark tasks. (a) Push Left-Right, (b) Slide Near-Far, (c) Reach Near-Far, and (d) Pick Low-High.

5. Experiments

In this section, we introduce a new benchmark consisting of 9 task groups and 26 tasks to evaluate the OOD generalization performance of offline GCRL algorithms.

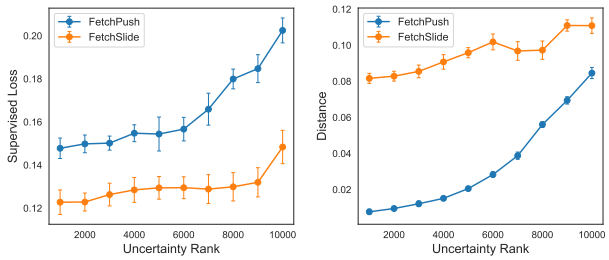
5.1. Environments and Experimental Setup

Environments The introduced benchmark is modified from MuJoCo robotic manipulation environments (Plappert et al., 2018). Agents aim to move a box, a robot arm, or a bionic hand to reach desired positions. The reward for each environment is sparse and binary, i.e., 1 for reaching the desired goal and 0 otherwise. As listed in Table 1, there are 9 task groups with a total of 26 tasks, 17 of which are OOD tasks whose goals are not in the training data. For example, as shown in Figure 3(a), the dataset of Push Left-Right contains trajectories where both the initial object and achieved goals are on the right side of the table. Then the IID task is evaluating agents with object and goals on the right side (i.e., Right2Right). The OOD tasks can be generated by changing the side of the initial object or desired goals. Following (Yang et al., 2022b), we collect datasets with the online DDPG+HER agent. More information about the task design and offline datasets can be found in Appendix C.

Experimental Setup We compare GOAT with current SOTA offline GCRL methods, including WGCSL (Yang et al., 2022b), GoFAR (Ma et al., 2022b), CQL+HER (Chebotar et al., 2021), GCSL (Ghosh et al., 2019), and DDPG+HER (Andrychowicz et al., 2017). Besides, we also include a SOTA ensemble-based offline RL methods, MSG (Ghasemipour et al., 2022), namely "MSG+HER". To evaluate performance, we assess agents across 200 randomly generated goals for each task and benchmark their average success rates. More details and additional experiments are provided in Appendix C and Appendix D.

5.2. Understanding the Uncertainty Weight

In our theoretical analysis, the uncertainty weight (UW) has the effect of reducing the worst-case distance between the training and unknown testing distributions. To make it more clear, we collect 10000 relabeled samples (s, a, g')



(a) (b)

Figure 4. Correlation between (a) supervised loss, (b) state-goal distance and the uncertainty rank.

and rank these samples according to the UW in Eq (4). For a sample (s, a, g'), we record two values, the supervised loss (i.e., $\|a - \pi_\theta(s, g')\|_2^2$), and the distance between the desired goal and the achieved goal (i.e., $\|(g' - \phi(s))\|_2^2$, short for "state-goal distance"). Then, we average their values for every 1000 ranked samples. The results are shown in Figure 4. Interestingly, UW assigns more weights to samples with larger supervised loss, which may also be related to Distributionally Robust Optimization (Rahimian & Mehrotra, 2019; Goh & Sim, 2010), thereby improving performance on worst-case scenarios. Moreover, UW prefers samples with larger state-goal distance. Since every state-goal pair (s, g') defines a task from s to g', UW enhances harder tasks with larger state-goal distance. In general, OOD goals are relatively further away than IID goals, which also interprets why UW works for OOD generalization.

5.3. Generalizing to OOD Goals

Table 1 reports the average success rates of GOAT and other baselines on the introduced benchmark. We denote GOAT with expectile regression as GOAT(τ), where $\tau < 0.5$. From the results, we can conclude that OOD generalization is more challenging than IID tasks. For example, the performance of GoFAR, GCSL, and BC drops by more than half on OOD tasks. On the contrary, GOAT and GOAT(τ) achieve the highest OOD success rates over 16 out of 17 tasks. Compared with WGCSL, GOAT improves the IID performance slightly but considerably enhances the OOD performance.

Table 1. Average success rates (%) with standard deviation over 5 random seeds. Blue lines and purple lines refer to IID and OOD tasks, respectively. Top two success rates for each task are highlighted.

Task Group	Task	GOAT(τ)	GOAT	WGCSL	GCSL	BC	GoFAR	DDPG+HER	CQL+HER	MSG+HER
Reach Left-Right	Right	100.0±0.0	100.0±0.0	100.0±0.0	93.6±4.3	92.0±3.0	100.0±0.0	99.6±0.6	100.0±0.0	99.4±0.6
	Left	99.9±0.2	99.0±2.0	97.8±4.4	36.3±10.9	30.4±15.2	54.2±9.3	73.8±27.6	94.5±6.3	85.6±15.7
	Average	**99.9**	**99.5**	98.9	65.0	61.2	77.1	86.7	97.2	92.5
Reach Near-Far	Near	100.0±0.0	100.0±0.0	100.0±0.0	79.7±3.0	85.3±4.3	100.0±0.0	95.9±2.0	100.0±0.0	98.6±2.8
	Far	90.9±1.5	97.6±1.1	89.0±2.1	33.5±5.5	37.9±9.7	85.0±1.9	66.8±6.9	88.0±2.1	77.8±9.7
	Average	**95.4**	**98.8**	94.5	56.6	61.6	92.5	81.4	94.0	88.2
Push Left-Right	Right2Right	96.2±1.2	95.9±1.2	93.2±0.9	82.1±3.7	78.9±3.8	95.9±1.4	60.1±6.0	83.3±2.7	92.8±0.9
	Right2Left	75.6±3.6	69.3±6.6	63.3±8.9	40.1±6.0	25.6±2.7	43.8±4.7	28.5±4.3	46.2±7.1	52.9±6.5
	Left2Right	78.8±6.8	76.0±7.4	67.6±7.1	38.8±6.8	33.5±8.1	59.7±4.3	20.6±11.5	40.4±12.1	59.3±7.7
	Left2Left	75.6±12.1	61.1±7.6	47.7±7.4	35.4±6.6	20.9±3.2	32.5±5.8	27.0±3.8	34.9±5.9	38.8±7.9
	Average	**81.5**	**75.6**	68.0	49.1	39.7	58.0	34.1	51.2	61.0
Push Near-Far	Near2Near	97.2±0.7	92.0±2.6	93.5±1.0	77.6±4.7	67.5±3.6	92.6±2.2	39.3±22.4	77.7±3.9	84.7±6.1
	Near2Far	78.4±3.5	70.3±5.7	67.0±5.4	43.1±7.2	24.9±5.9	60.9±3.8	30.5±12.1	60.0±6.2	58.4±2.1
	Far2Near	70.5±2.4	69.5±3.6	68.0±2.4	47.4±3.5	40.2±7.5	65.0±4.8	25.0±12.8	61.1±4.3	56.5±6.0
	Far2Far	55.1±2.4	50.8±1.8	51.1±4.7	27.9±4.1	15.3±2.7	41.3±3.1	18.0±7.0	47.1±2.4	41.7±5.4
	Average	**75.3**	**70.6**	69.9	49.0	37.0	65.0	28.2	61.5	60.3
Pick Left-Right	Right2Right	96.5±1.1	97.3±1.2	93.8±5.3	53.4±14.1	52.9±7.5	56.9±4.3	40.4±13.1	91.9±6.8	94.9±2.2
	Right2Left	87.9±5.1	88.6±1.1	89.4±3.9	20.7±6.9	5.6±2.1	9.3±1.8	52.7±14.9	82.4±12.6	89.3±6.8
	Left2Right	91.4±2.3	93.9±1.9	90.0±4.1	47.0±10.9	37.2±6.4	51.1±6.5	9.8±5.7	86.4±8.6	60.8±16.5
	Left2Left	87.6±5.7	88.3±3.7	87.0±5.1	24.7±7.8	3.3±1.4	6.0±2.0	26.4±10.9	83.5±9.1	66.9±7.0
	Average	**90.8**	**92.0**	90.0	36.4	24.8	30.8	32.3	86.1	78.0
Pick Low-High	Low	99.3±0.5	99.8±0.2	98.6±1.3	84.4±3.6	72.4±5.4	95.2±1.6	50.4±23.9	100.0±0.0	97.3±2.2
	High	78.3±6.3	71.9±6.4	66.6±6.6	28.4±6.9	3.0±1.6	7.6±3.1	17.0±10.2	44.6±9.2	23.3±7.8
	Average	**88.8**	**85.8**	82.6	56.4	37.7	51.4	33.7	72.3	60.3
Slide Left-Right	Right2Right	82.0±3.2	79.0±5.8	70.8±13.5	62.2±7.0	60.3±4.7	62.6±8.7	4.7±1.5	20.3±2.5	20.8±5.0
	Right2Left	45.1±8.8	41.3±7.1	36.2±8.6	11.5±2.0	15.7±6.0	31.6±3.9	0.3±0.4	8.6±3.0	7.3±4.9
	Left2Right	79.6±2.7	59.0±7.6	50.7±12.7	29.1±4.8	41.8±7.2	51.0±10.5	0.2±0.2	1.7±0.7	3.6±4.3
	Left2Left	52.5±8.3	50.1±9.5	35.3±11.3	25.5±5.4	33.7±10.6	28.2±2.6	2.1±1.1	4.3±2.5	7.1±3.3
	Average	**64.8**	**57.4**	48.3	32.1	37.9	43.4	1.8	8.7	9.7
Slide Near-Far	Near	77.4±4.5	76.9±3.3	73.1±5.8	28.0±7.1	26.6±8.3	69.3±2.8	11.3±4.5	43.5±3.3	28.3±9.5
	Far	25.1±3.9	29.0±4.5	17.4±3.2	0.0±0.0	0.0±0.0	24.1±2.9	4.4±3.7	7.4±3.8	2.6±1.4
	Average	**51.2**	**53.0**	45.2	14.0	13.3	46.7	7.8	25.5	15.4
HandReach Near-Far	Near	72.6±5.3	71.9±3.2	70.0±3.6	0.0±0.0	0.0±0.0	77.4±1.7	0.0±0.0	1.8±3.6	0.0±0.0
	Far	33.1±4.5	38.4±4.1	31.8±3.8	0.1±0.2	0.0±0.0	36.9±3.1	0.0±0.0	0.0±0.0	0.0±0.0
	Average	52.8	**55.2**	50.9	0.0	0.0	**57.1**	0.0	0.9	0.0
Average	IID Tasks	**91.2**	**90.3**	88.1	62.3	59.5	83.3	44.6	68.7	68.5
	OOD Tasks	**70.9**	**67.9**	62.1	28.8	21.7	40.5	23.7	46.5	43.1

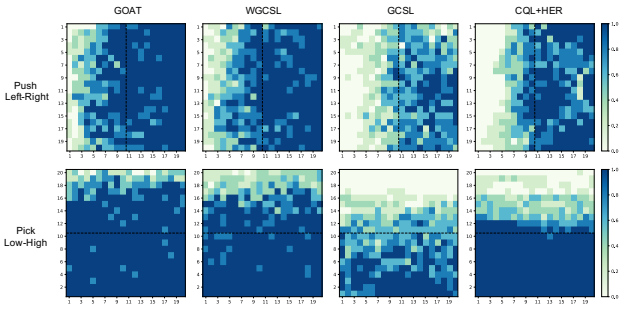


Figure 5. The coverage of successful goals. The darkness of color represents the success rate of each goal for 5 random seeds. The black dotted line is the dividing line between IID and OOD goals. The IID areas are the right half (top row) and the lower half (bottom row) rectangles for the two tasks.

While CQL+HER and MSG+HER exhibit better performance than GCSL and BC, they are worse than weighted imitation learning methods WGCSL and GOAT, possibly due to pessimism restraining generalization. Besides, they fail on hard tasks such as Slide and HandReach. Another observation is that although GOAT, WGCSL, GoFAR are all weighted imitation learning methods, their OOD performance varies significantly, indicating components of weighted imitation learning also matter. To better understand these components, we will present an in-depth ablation analysis in Section 5.4.

In Figure 5, we visualize the coverage of successful goals in Push Left-Right and Pick Low-High tasks, given fixed initial states at the right center and bottom center, respectively. Each small square represents a goal in the goal space, and their darkness represents the average success rate for 5 random seeds. The results demonstrate that GOAT has the largest coverage of successful goals among the baselines, including the strong baseline WGCSL. Notably, both CQL+HER and GCSL exhibit limitations in their capacity to generalize to unseen goals. Specifically, CQL+HER is restricted to the training distribution, whereas GCSL displays inadequate coverage for even IID goals due to overfitting to noise. The observed results are also in alignment with our didactic example in Section 3.1.

Table 2. Ablations of each component of GOAT.

Success Rate (%)	BC	+HER	+EAW	+DSW	+Ens	+ UW	+ ER
OOD Tasks	21.7	28.8	53.1	62.1	63.4	67.9	70.9
Increment	+0	+7.1	+24.3	+9.0	+1.3	+4.5	+3.0
All Tasks	34.8	50.7	65.4	71.1	72.2	75.7	77.9
Increment	+0	+15.9	+14.7	+5.7	+1.1	+3.5	+2.2

5.4. Ablations

To measure the contribution of each component of GOAT, we gradually add one component from BC to GOAT and record the performance increment caused by each component. As shown in Table 2, the recorded results are average success rates of 17 OOD tasks and all 26 tasks. On average, each component brings improvement for OOD generalization of offline GCRL. For OOD tasks, EAW and DSW contribute the most by improving the surrogate expert policy for imitating. Besides, HER and UW also bring considerable improvement through data augmentation and uncertainty reweighting. In addition, ensemble technique (Ens) improves the estimation of value functions but has the least effect on the overall performance. Expectile regression (ER) improves the average performance, but slightly reduces OOD performance on hard tasks such as Slide Near-Far and HandReach as shown in Table 1. Furthermore, we also compare variants of GOAT with V functions and χ^2-divergence in Appendix D.4.

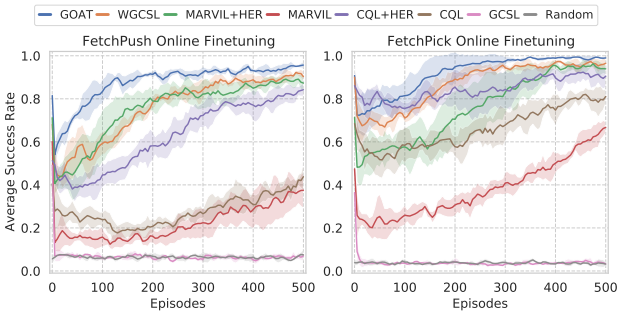


Figure 6. Online fine-tuning using DDPG+HER for different pre-trained agents on FetchPush and FetchPick tasks.

5.5. Online Fine-tuning to Unseen Goals

We design an experiment to fine-tune pre-trained agents with online samples to verify whether the generalization ability of pre-trained agents is beneficial for online learning. The pre-trained agents are trained on offline datasets with partial coverage (Right2Right) and fine-tuned to full coverage (Right2Right, Right2Left, Left2Right, Left2Left). We apply DDPG+HER to fine-tune the policies and value functions after each episode collection. Additional Gaussian noise and random actions are applied for exploration. More detailed description can be found in Appendix D.12.

The experimental results are show in Figure 6, which demonstrate that (1) most pre-trained agents learn faster than the randomly initialized agent (namely "random") and (2) different initializations for goal-conditioned agents perform significantly different during fine-tuning. Specifically, GOAT outperforms other methods on the efficiency of online fine-tuning, while CQL, MARVIL (Wang et al., 2018) and GCSL result in slow-growing curves. We observe that the performance of GCSL initialization is similar to that of random initialization. It is likely that value networks contain valuable information for DDPG+HER agents to transfer from offline to online. This also explains why GOAT brings improvement, as it enhances value function learning via ensemble and expectile regression.

6. Related Work

Goal-conditioned RL GCRL is a branch of reinforcement learning where agents need to achieve multiple goals sharing the same environmental dynamics (Schaul et al., 2015; Andrychowicz et al., 2017). Goal relabeling (Andrychowicz et al., 2017; Li et al., 2020a; Eysenbach et al., 2020; Yang et al., 2021a) is an effective technique that handles the sparse reward problem in GCRL and augments the data for policy learning. To improve the generalization ability, several prior works mainly focus on learning generalizable representations, e.g., combining Successor Feature with UVFA (Ma et al., 2018; Borsa et al., 2018), decomposing Q value via Bilinear Value Networks (Hong et al., 2022), and learning discretization bottleneck representation for goals (Islam et al., 2022). Han et al. (2021) propose to learn invariant representation via aligned sampling to tackle the spurious feature problem. Our work differs from previous works in that we consider the offline GCRL setting, where pessimism can inhibit OOD generalization.

Offline RL and Offline GCRL Offline RL handles the distribution shift challenge and learns policies from static datasets (Levine et al., 2020). Generally, offline RL methods can be divided into two main directions, i.e., policy regularization and value underestimation. The first direction includes methods that constrain the learned policy to be close to the behavior policy under certain distance measure (Wang et al., 2018; Fujimoto et al., 2019; Nair et al., 2020b; Yang et al., 2021b; Fujimoto & Gu, 2021). Another direction is to underestimate values for OOD actions (Kumar et al., 2020; Yu et al., 2021; An et al., 2021; Bai et al., 2022; Yang et al., 2022a; Ghasemipour et al., 2022). As for offline GCRL, current methods can also be grouped into policy regularization (Yang et al., 2022b; Ma et al., 2022b) and value underestimation (Chebotar et al., 2021) methods. Different from prior works, our work focuses on learning policies from offline data and improving the ability to generalize to out-of-distribution goals.

Domain Generalization (DG) DG aims to learn a model from training domains that can generalize on unseen testing domains (Zhou et al., 2021a; Wang et al., 2022). Solutions to DG include data augmentation (Zhou et al., 2020; 2021b), meta learning (Li et al., 2018; Balaji et al., 2018; Yong et al., 2023), invariant representation learning (Arjovsky et al., 2019; Lin et al., 2022a; Zhou et al., 2022; 2023; Lin et al., 2022b) and distributionally robust optimization (Sagawa et al., 2019). In reinforcement learning, DG is handled with data augmentation (Wang et al., 2020), environment generation (Jiang et al., 2021), and representation learning (Mazoure et al., 2021; Sonar et al., 2021; Han et al., 2021). Unlike these works, we mainly consider the covariate shift and handle pessimism and generalization simultaneously for OOD generalization of offline GCRL.

7. Conclusion

Learning from purely offline datasets and generalizing to unseen goals is one of the pursuits of the RL community. In this paper, we investigate the problem of out-of-distribution (OOD) generalization of offline GCRL. Through theoretical analysis and empirical evaluation, we demonstrate that (1) the choice of offline RL methods, particularly weighted imitation learning, and (2) the techniques to minimize the generalization bound, are crucial for this problem. With these insights, we propose GOAT, a new weighted imitation learning method that achieves strong OOD generalization performance across a variety of tasks. In the future, we believe our work will inspire more scalable and generalizable reinforcement learning research.

8. Limitations

The major limitation of this work is that we mainly consider algorithmic designs motivated by the OOD generalization theory. There are many interesting future directions not included in this paper, e.g., studying representation learning (Mazoure et al., 2021), goal embeddings (Islam et al., 2022), world models (Anand et al., 2021; Ding et al., 2022), and network designs (Lee et al., 2022; Xu et al., 2022; Hong et al., 2022) to improve OOD generalization for offline RL and offline GCRL.

Acknowledgements

This work is supported by GRF 16310222 and GRF 16201320, in part by Science and Technology Innovation 2030 - "New Generation Artificial Intelligence" Major Project (No. 2018AAA0100904) and the National Natural Science Foundation of China (62176135). The authors would like to thank the anonymous reviewers for their comments to improve the paper.

References

An, G., Moon, S., Kim, J.-H., and Song, H. O. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

Anand, A., Walker, J., Li, Y., Vértes, E., Schrittwieser, J., Ozair, S., Weber, T., and Hamrick, J. B. Procedural generalization by planning with self-supervised world models. *arXiv preprint arXiv:2111.01587*, 2021.

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

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *ArXiv*, abs/1907.02893, 2019.

Bai, C., Wang, L., Yang, Z., Deng, Z.-H., Garg, A., Liu, P., and Wang, Z. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *International Conference on Learning Representations*, 2022.

Balaji, Y., Sankaranarayanan, S., and Chellappa, R. Metareg: Towards domain generalization using meta-regularization. *Advances in neural information processing systems*, 31, 2018.

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Blanchard, G., Lee, G., and Scott, C. Generalizing from several related classification tasks to a new unlabeled sample. *Advances in neural information processing systems*, 24, 2011.

Borsa, D., Barreto, A., Quan, J., Mankowitz, D., Munos, R., Van Hasselt, H., Silver, D., and Schaul, T. Universal successor features approximators. *arXiv preprint arXiv:1812.07626*, 2018.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Chebotar, Y., Hausman, K., Lu, Y., Xiao, T., Kalashnikov, D., Varley, J., Irpan, A., Eysenbach, B., Julian, R., Finn, C., et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. *arXiv preprint arXiv:2104.07749*, 2021.

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, J. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 1282–1289. PMLR, 2019.

Csiszár, I. and Körner, J. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

Ding, W., Lin, H., Li, B., and Zhao, D. Generalizing goal-conditioned reinforcement learning with variational causal reasoning. *arXiv preprint arXiv:2207.09081*, 2022.

Eysenbach, B., Geng, X., Levine, S., and Salakhutdinov, R. R. Rewriting history with inverse rl: Hindsight inference for policy improvement. *Advances in neural information processing systems*, 33:14783–14795, 2020.

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

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Ghasemipour, S. K. S., Gu, S. S., and Nachum, O. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters. *arXiv preprint arXiv:2205.13703*, 2022.

Ghosh, D., Gupta, A., Reddy, A., Fu, J., Devin, C., Eysenbach, B., and Levine, S. Learning to reach goals via iterated supervised learning. *arXiv preprint arXiv:1912.06088*, 2019.

Goh, J. and Sim, M. Distributionally robust optimization and its tractable approximations. *Operations research*, 58 (4-part-1):902–917, 2010.

Han, B., Zheng, C., Chan, H., Paster, K., Zhang, M., and Ba, J. Learning domain invariant representations in goal-conditioned block mdps. *Advances in Neural Information Processing Systems*, 34:764–776, 2021.

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

Hong, Z.-W., Yang, G., and Agrawal, P. Bi-linear value networks for multi-goal reinforcement learning. In *International Conference on Learning Representations*, 2022.

Islam, R., Zang, H., Goyal, A., Lamb, A., Kawaguchi, K., Li, X., Laroche, R., Bengio, Y., and des Combes, R. T. Discrete compositional representations as an abstraction for goal conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Jiang, M., Grefenstette, E., and Rocktäschel, T. Prioritized level replay. In *International Conference on Machine Learning*, pp. 4940–4950. PMLR, 2021.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.

Kirk, R., Zhang, A., Grefenstette, E., and Rocktäschel, T. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76: 201–264, 2023.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

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

Kumar, A., Hong, J., Singh, A., and Levine, S. Should i run offline reinforcement learning or behavioral cloning? In *International Conference on Learning Representations*, 2021.

Lee, K.-H., Nachum, O., Yang, M., Lee, L., Freeman, D., Xu, W., Guadarrama, S., Fischer, I., Jang, E., Michalewski, H., et al. Multi-game decision transformers. *arXiv preprint arXiv:2205.15241*, 2022.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, A., Pinto, L., and Abbeel, P. Generalized hindsight for reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020a.

Li, D., Yang, Y., Song, Y.-Z., and Hospedales, T. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Li, J., Koyamada, S., Ye, Q., Liu, G., Wang, C., Yang, R., Zhao, L., Qin, T., Liu, T.-Y., and Hon, H.-W. Suphx: Mastering mahjong with deep reinforcement learning. *arXiv preprint arXiv:2003.13590*, 2020b.

Lin, Y., Dong, H., Wang, H., and Zhang, T. Bayesian invariant risk minimization. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, 2022a.

Lin, Y., Zhu, S., Tan, L., and Cui, P. Zin: When and how to learn invariance without environment partition? In *Neural Information Processing Systems*, 2022b.

Ma, C., Wen, J., and Bengio, Y. Universal successor representations for transfer reinforcement learning. *arXiv preprint arXiv:1804.03758*, 2018.

Ma, Y. J., Sodhani, S., Jayaraman, D., Bastani, O., Kumar, V., and Zhang, A. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022a.

Ma, Y. J., Yan, J., Jayaraman, D., and Bastani, O. How far i'll go: Offline goal-conditioned reinforcement learning via f-advantage regression. *arXiv preprint arXiv:2206.03023*, 2022b.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Mazoure, B., Kostrikov, I., Nachum, O., and Tompson, J. Improving zero-shot generalization in offline reinforcement learning using generalized similarity functions. *arXiv preprint arXiv:2111.14629*, 2021.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

Muandet, K., Balduzzi, D., and Schölkopf, B. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pp. 10–18. PMLR, 2013.

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

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020b.

Nikulin, A., Kurenkov, V., Tarasov, D., and Kolesnikov, S. Anti-exploration by random network distillation. *arXiv preprint arXiv:2301.13616*, 2023.

Pathak, D., Gandhi, D., and Gupta, A. Self-supervised exploration via disagreement. In *International conference on machine learning*, pp. 5062–5071. PMLR, 2019.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Pitis, S., Chan, H., Zhao, S., Stadie, B., and Ba, J. Maximum entropy gain exploration for long horizon multi-goal reinforcement learning. In *International Conference on Machine Learning*, pp. 7750–7761. PMLR, 2020.

Pitis, S., Creager, E., Mandlekar, A., and Garg, A. Mocoda: Model-based counterfactual data augmentation. In *Advances in Neural Information Processing Systems*, 2022.

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

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Rahimian, H. and Mehrotra, S. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

Rezaeifar, S., Dadashi, R., Vieillard, N., Hussenot, L., Bachem, O., Pietquin, O., and Geist, M. Offline reinforcement learning as anti-exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8106–8114, 2022.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In *International Conference on Machine Learning*, pp. 1312–1320, 2015.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Sonar, A., Pacelli, V., and Majumdar, A. Invariant policy optimization: Towards stronger generalization in reinforcement learning. In *Learning for Dynamics and Control*, pp. 21–33. PMLR, 2021.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, 2019.

Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W., and Yu, P. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Wang, K., Kang, B., Shao, J., and Feng, J. Improving generalization in reinforcement learning with mixture regularization. *Advances in Neural Information Processing Systems*, 33:7968–7978, 2020.

Wang, Q., Xiong, J., Han, L., Liu, H., Zhang, T., et al. Exponentially weighted imitation learning for batched historical data. *Advances in Neural Information Processing Systems*, 31, 2018.

Xu, M., Shen, Y., Zhang, S., Lu, Y., Zhao, D., Tenenbaum, J., and Gan, C. Prompting decision transformer for few-shot policy generalization. In *International Conference on Machine Learning*, pp. 24631–24645. PMLR, 2022.

Xu, T., Li, Z., and Yu, Y. Error bounds of imitating policies and environments. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Yang, R., Fang, M., Han, L., Du, Y., Luo, F., and Li, X. Mher: Model-based hindsight experience replay. In *Deep RL Workshop NeurIPS 2021*, 2021a.

Yang, R., Bai, C., Ma, X., Wang, Z., Zhang, C., and Han, L. Rorl: Robust offline reinforcement learning via conservative smoothing. In *Advances in Neural Information Processing Systems*, 2022a.

Yang, R., Lu, Y., Li, W., Sun, H., Fang, M., Du, Y., Li, X., Han, L., and Zhang, C. Rethinking goal-conditioned supervised learning and its connection to offline rl. In *International Conference on Learning Representations*, 2022b.

Yang, Y., Ma, X., Li, C., Zheng, Z., Zhang, Q., Huang, G., Yang, J., and Zhao, Q. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:10299–10312, 2021b.

Yong, L., Pi, R., Zhang, W., Xia, X., Gao, J., Zhou, X., Liu, T., and Han, B. A holistic view of label noise transition matrix in deep learning and beyond. In *The Eleventh International Conference on Learning Representations*, 2023.

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Zhang, C., Zhang, L., and Ye, J. Generalization bounds for domain adaptation. *Advances in neural information processing systems*, 25, 2012.

Zhao, R., Sun, X., and Tresp, V. Maximum entropy-regularized multi-goal reinforcement learning. In *International Conference on Machine Learning*, pp. 7553–7562. PMLR, 2019.

Zhou, K., Yang, Y., Hospedales, T., and Xiang, T. Learning to generate novel domains for domain generalization. In *European conference on computer vision*, pp. 561–578. Springer, 2020.

Zhou, K., Liu, Z., Qiao, Y., Xiang, T., and Loy, C. C. Domain generalization: A survey. 2021a.

Zhou, K., Yang, Y., Qiao, Y., and Xiang, T. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021b.

Zhou, X., Lin, Y., Zhang, W., and Zhang, T. Sparse invariant risk minimization. In *International Conference on Machine Learning*, 2022.

Zhou, X., Lin, Y., Pi, R., Zhang, W., Xu, R., Cui, P., and Zhang, T. Model agnostic sample reweighting for out-of-distribution learning. *ArXiv*, abs/2301.09819, 2023.

Appendix

A. Algorithm Pseudo Code

Algorithm 1 GOAT Algorithm

Initialize policy π_θ and N value functions $Q_1, \ldots, Q_N$, and two FIFO queues $B_a = \{\}$ and $B_{std} = \{\}$;
for training step $= 1, 2, \ldots$ **do**
  Sample a mini-batch from the offline dataset: $\{(s_t, a_t, g, r_t, s_{t+1})\} \sim D$;
  Relabel the mini-batch with a probability of $p_{relabel}$: $\{(s_t, a_t, g', r'_t, s_{t+1})\} \sim D_{relabel}$;
  Update value functions $Q_i, i \in [1, N]$ to minimize Eq (3) with the mini-batch ;
  Estimate advantage values $A(s_t, a_t, g')$ using $Q_i, i \in [1, N]$ and store them into the queue B_a;
  Get the α percentile advantage value from B_a to calculate the DSW;
  Estimate the bootstrapped uncertainty Std(s_t, g) and store them into B_{std};
  Compute the UW according to Eq (4);
  Update policy π_θ to maximize the objective in Eq (2) with the mini-batch:
end for

B. Theoretical Proofs

B.1. Useful Lemmas

Lemma B.1. *Assume the maximum reward is R_{max}, For any two goal-conditioned policies π and π_E, we have that*

$$V^{\pi_E}(s_0, g) - V^{\pi}(s_0, g) \leq \frac{2R_{\max}}{(1-\gamma)^2} \mathbb{E}_{s \sim d_{\pi_E}(s|s_0,g)} \left[D_{\mathrm{TV}}\big(\pi(\cdot|s,g), \pi_E(\cdot|s,g)\big) \right]$$

Proof. For any policy π, its value function can be formulated as $V^\pi = \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim\rho_\pi}[r(s,a)]$ (Puterman, 2014). In the goal-conditioned setting, we also need to include goals into consideration. Then, we can derive

$$\begin{aligned}
|V^{\pi_E}(s_0, g) - V^{\pi}(s_0, g)| &= \left| \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim\rho_{\pi_E}(\cdot|s_0,g)}[r(s,a,g)] - \frac{1}{1-\gamma}\mathbb{E}_{(s,a,g)\sim\rho_\pi(\cdot|s_0,g)}[r(s,a,g)] \right| \\
&\leq \frac{1}{1-\gamma} \sum_{(s,a)\in\mathcal{S}\times\mathcal{A}} \left| \big(\rho_{\pi_E}(s,a|s_0,g) - \rho_\pi(s,a|s_0,g)\big) r(s,a,g) \right| \\
&\leq \frac{2R_{\max}}{1-\gamma} D_{\mathrm{TV}}(\rho_{\pi_E}(\cdot|s_0,g), \rho_\pi(\cdot|s_0,g)).
\end{aligned}$$

With Lemma 5 in (Xu et al., 2020), we comlete the proof:

$$|V^{\pi_E}(s_0, g) - V^{\pi}(s_0, g)| \leq \frac{2R_{\max}}{1-\gamma} D_{\mathrm{TV}}(\rho_{\pi_E}(\cdot|s_0,g), \rho_\pi(\cdot|s_0,g)) \leq \frac{2R_{\max}}{(1-\gamma)^2} \mathbb{E}_{s \sim d_{\pi_E}(s|s_0,g)} \left[D_{\mathrm{TV}}\big(\pi(\cdot|s,g), \pi_E(\cdot|s,g)\big) \right]$$

□

Lemma B.2. *Assume the maximum reward is R_{max}, For any two goal-conditioned policies π and π_E, we have that*

$$\mathrm{SubOpt}(\pi_E, \pi) = \mathbb{E}_{(s_0,g)\sim P_X^{\mathcal{T}}}[V^{\pi_E}(s_0, g) - V^{\pi}(s_0, g)] \leq \frac{2R_{\max}}{(1-\gamma)^2} \mathbb{E}_{\substack{(s_0,g)\sim P_X^{\mathcal{T}} \\ s \sim d_{\pi_E}(s|s_0,g)}} \left[D_{\mathrm{TV}}\big(\pi(\cdot|s,g), \pi_E(\cdot|s,g)\big) \right]$$

Lemma B.2 is a direct result of combining Lemma B.1 and the definition of the suboptimality in Eq (1).

Definition B.3. For any state-goal distribution $\rho(s, g)$, we define

$$\varepsilon^\rho(\pi_E, \pi) = \mathbb{E}_{(s,g)\sim\rho(s,g)} \left[D_{\mathrm{TV}}\big(\pi(\cdot|s,g), \pi_E(\cdot|s,g)\big) \right].$$

Lemma B.4. *For any three policy π_1, π_2, π_3 and any state-goal distribution ρ, we have:*

$$\varepsilon^{\rho}(\pi_1, \pi_2) \leq \varepsilon^{\rho}(\pi_1, \pi_3) + \varepsilon^{\rho}(\pi_3, \pi_2)$$

Proof. This can be proved by noticing that D_{TV} is a distance metric.

$$\begin{aligned}
\varepsilon^{\rho}(\pi_1, \pi_2) &= \mathbb{E}_{(s,g)\sim\rho(s,g)} \left[D_{\mathrm{TV}}\big(\pi_1(\cdot|s,g), \pi_2(\cdot|s,g)\big) \right] \\
&\leq \mathbb{E}_{(s,g)\sim\rho(s,g)} \left[D_{\mathrm{TV}}\big(\pi_1(\cdot|s,g), \pi_3(\cdot|s,g)\big) + D_{\mathrm{TV}}\big(\pi_3(\cdot|s,g), \pi_2(\cdot|s,g)\big) \right] \\
&\leq \varepsilon^{\rho}(\pi_1, \pi_3) + \varepsilon^{\rho}(\pi_3, \pi_2)
\end{aligned}$$

□

Lemma B.5. *Assume the expert policy π_E is invariant across training and testing domains. For a policy π, we have*

$$\varepsilon^{\mathcal{T}}(\pi_E, \pi) \leq \varepsilon^{\mathcal{S}}(\pi_E, \pi) + d_1(\mathcal{T}, \mathcal{S})$$

where the variation divergence d_1 between two distribution S_1 and S_2 is defined as follows:

$$d_1(S_1, S_2) = 2 \sup_{\mathcal{J} \subset \mathcal{X}} \left| \sum_{x \in \mathcal{J}} \left(P_{S_1}(x) - P_{S_2}(x) \right) \right|,$$

Proof. With the definition of variation divergence, we have

$$\begin{aligned}
\varepsilon^{\mathcal{T}}(\pi_E, \pi) &= \varepsilon^{\mathcal{T}}(\pi_E, \pi) + \varepsilon^{\mathcal{S}}(\pi_E, \pi) - \varepsilon^{\mathcal{S}}(\pi_E, \pi) \leq \varepsilon^{\mathcal{S}}(\pi_E, \pi) + |\varepsilon^{\mathcal{T}}(\pi_E, \pi) - \varepsilon^{\mathcal{S}}(\pi_E, \pi)| \\
&= \varepsilon^{\mathcal{S}}(\pi_E, \pi) + \frac{1}{2} \sum_{(s_0, g)} |P_X^{\mathcal{T}}(s_0, g) - P_X^{\mathcal{S}}(s_0, g)| \sum d_{\pi_E}(s|s_0, g) \sum_a |\pi_E(a|s, g) - \pi(a|s, g)| \\
&\leq \varepsilon^{\mathcal{S}}(\pi_E, \pi) + \sum_{(s_0, g)} |P_X^{\mathcal{T}}(s_0, g) - P_X^{\mathcal{S}}(s_0, g)| \\
&\leq \varepsilon^{\mathcal{S}}(\pi_E, \pi) + d_1(\mathcal{T}, \mathcal{S})
\end{aligned}$$

□

Lemma B.6 (Generalization Bound for Finite ERM)**.** *Consider finite hypothesis space $\mathcal{F}$ and bounded loss function in $[a, b]$. When optimizing empirical loss function $\hat{L}(f) = \frac{1}{m} \sum_i^m \mathcal{L}(f(x_i), y_i)$ instead of the expected one $L(f) = \mathbb{E}_{(x,y)} \mathcal{L}(f(x), y)$, with probability as least $1 - \delta$, the true loss can be bounded as:*

$$L(f) \leq \hat{L}(f) + \sqrt{\frac{(b-a)^2 (\log 2|\mathcal{F}| + \log \frac{1}{\delta})}{2m}}$$

Lemma B.6 is a well-known result from (Mohri et al., 2018).

B.2. Proof of Theorem 3.1

Proof. With Lemma B.2 and the definition of policy discrepancy on training and testing distributions:

$$\begin{aligned}
\varepsilon^{\mathcal{T}}(\pi_E, \pi) &= \mathbb{E}_{\substack{(s_0, g)\sim P_X^{\mathcal{T}} \\ s\sim d_{\pi_E}(s|s_0, g)}} \left[D_{\mathrm{TV}}\big(\pi(\cdot|s, g), \pi_E(\cdot|s, g)\big) \right], \\
\varepsilon^{\mathcal{S}}(\pi_E, \pi) &= \mathbb{E}_{\substack{(s_0, g)\sim P_X^{\mathcal{S}} \\ s\sim d_{\pi_E}(s|s_0, g)}} \left[D_{\mathrm{TV}}\big(\pi(\cdot|s, g), \pi_E(\cdot|s, g)\big) \right].
\end{aligned}$$

we have

$$\mathrm{SubOpt}(\pi_E, \pi) \leq \frac{2R_{\max}}{(1-\gamma)^2} \mathbb{E}_{\substack{(s_0, g)\sim P_X^{\mathcal{T}} \\ s\sim d_{\pi_E}(s|s_0, g)}} \left[D_{\mathrm{TV}}\big(\pi(\cdot|s, g), \pi_E(\cdot|s, g)\big) \right] = \frac{2R_{\max}}{(1-\gamma)^2} \varepsilon^{\mathcal{T}}(\pi_E, \pi)$$

Regarding $\varepsilon^{\mathcal{T}}(\pi_E, \pi)$, we use Lemma B.4 and Lemma B.5 to obtain an upper bound:

$$\varepsilon^{\mathcal{T}}(\pi_E, \pi) \leq \varepsilon^{\mathcal{S}}(\pi_E, \pi) + d_1(\mathcal{T}, \mathcal{S}) \leq \varepsilon^{\mathcal{S}}(\hat{\pi}_E, \pi) + \varepsilon^{\mathcal{S}}(\pi_E, \hat{\pi}_E) + d_1(\mathcal{T}, \mathcal{S})$$

When we use finite sample to estimate $\varepsilon^{\mathcal{S}}(\hat{\pi}_E, \pi)$, the true loss can be bounded with Lemma B.6. Note that the D_{TV} can be bounded in $[0, 1]$. Therefore, we can complete the proof. With probability at least $1 - \delta$,

$$\mathrm{SubOpt}(\pi_E, \pi) \leq \frac{2R_{max}}{(1-\gamma)^2}\left[\hat{\varepsilon}^{\mathcal{S}}(\hat{\pi}_E, \pi) + \varepsilon^{\mathcal{S}}(\hat{\pi}_E, \pi_E) + d_1(\mathcal{T}, \mathcal{S}) + \sqrt{\frac{\log 2|\Pi| + \log \frac{1}{\delta}}{2m}}\right]$$

□

B.3. Proof of Theorem 3.2

Proof. **Step 1**. First we explicitly show

$$\sup_{Z \in \mathcal{Z}} d_1(Z, \bar{S}) = 2(1 - \frac{1}{C|\mathcal{X}|}). \tag{5}$$

On one side, simple algebra shows that the following distribution $S' \in \mathcal{S}$ will induce the distance shown in Eq (5):

$$P_{S'}(x) = \begin{cases} C, \text{ if } x \in \mathcal{J}, \\ 0, \text{ otherwise }. \end{cases}$$

where $|\mathcal{J}| = 1/C < |\mathcal{X}|$. On the other hand, we are going to show there is no distribution that can elicit a distance larger than that in Eq (5). We prove it by contradiction by assuming a distribution S'' which has

$$d_1(S'', \bar{S}) > 2(1 - \frac{1}{C|\mathcal{X}|}). \tag{6}$$

Then there exists $\mathcal{J}'' \subset \mathcal{X}$ such that

$$\left|\int_{x \in \mathcal{J}''} (P_{S''}(x) - P_{\bar{S}}(x))\, dx\right| > (1 - \frac{1}{C|\mathcal{X}|}). \tag{7}$$

With out loss of generality, we assume

$$\int_{x \in \mathcal{J}''} (P_{S''}(x) - P_{\bar{S}}(x))\, dx > (1 - \frac{1}{C|\mathcal{X}|}) \tag{8}$$

Denote $\bar{x}_{\mathcal{J}''} = \frac{1}{|\mathcal{J}''|}\int_{x \in \mathcal{J}''} P_{S''}(x)dx$. It is clear that $\bar{x}_{\mathcal{J}''} \leq C$. Then the RHS of Eq (8) is

$$|\mathcal{J}''|(\bar{x}_{\mathcal{J}''} - \frac{1}{|\mathcal{X}|}) = |\mathcal{J}''|\bar{x}_{\mathcal{J}''}(1 - \frac{1}{|\mathcal{X}|\bar{x}_{\mathcal{J}''}}) \tag{9}$$

$$\leq (1 - \frac{1}{|\mathcal{X}|\bar{x}_{\mathcal{J}''}})| \tag{10}$$

$$\leq (1 - \frac{1}{C|\mathcal{X}|}), \tag{11}$$

where the first inequality is due to $|\mathcal{J}''|\bar{x}_{\mathcal{J}''} \leq 1$ and the second inequality is due to $\bar{x}_{\mathcal{J}''} \leq C$. Thus we arrive at a contradiction. So Eq (6) does not hold. Putting these together, we show that Eq (5) holds.

Step 2. We now proceed to show $\forall S \in \mathcal{S}^-$,

$$\sup_{Z \in \mathcal{Z}} d_1(Z, S) > 2(1 - \frac{1}{C|\mathcal{X}|}). \tag{12}$$

We know that for any S in $\mathcal{S}^-$, there exists a subset $\mathcal{J} \subset \mathcal{X}$ such that

$$\int_{x \in \mathcal{J}} P_S(x)dx < |\mathcal{J}|/|\mathcal{X}| \tag{13}$$

Let J_M be the subset of $\mathcal{X}/\mathcal{J}$ which contains the smallest $1/C - |\mathcal{J}|$ points:

$$\mathcal{J}_M := \min_{\mathcal{M} \subset \mathcal{X}/\mathcal{J}, |\mathcal{M}| = 1/C - |\mathcal{J}|} \int_{\mathcal{M}} P_S(x)dx. \tag{14}$$

By the definition of $\mathcal{J}_M$, it is easy to see that the mean density ratio $\mathcal{X} \ / \ (\mathcal{J} \ \cup \ \mathcal{J}_M)$ is larger than that of $\mathcal{J}_M$,

$$\frac{1}{|\mathcal{X}| - 1/C} \int_{x \in \mathcal{X}/(\mathcal{J} \cup \mathcal{J}_M)} P_S(x)dx \geq \frac{1}{1/C - |\mathcal{J}|} \int_{x \in \mathcal{J}_M} P_S(x)dx, \tag{15}$$

We now proceed to prove (by contradiction) that

$$\frac{1}{1/C} \int_{x \in \mathcal{J}_M \cup \mathcal{J}} P_S(x)dx < 1/|\mathcal{X}|. \tag{16}$$

We first assume

$$\frac{1}{1/C} \int_{x \in \mathcal{J}_M \cup \mathcal{J}} P_S(x)dx \geq 1/|\mathcal{X}| \tag{17}$$

By Eq (13) and (17), we have

$$\int_{x \in \mathcal{J}_M} P_S(x)dx > \frac{1/C - |\mathcal{J}|}{|\mathcal{X}|}, \tag{18}$$

and further with Eq (15) we have

$$\int_{x \in \mathcal{X}/(\mathcal{J} \cup \mathcal{J}_M)} P_S(x)dx > \frac{|\mathcal{X}| - 1/C}{|\mathcal{X}|}. \tag{19}$$

Putting Eq (19) and Eq (17) together, we have

$$\int_{x \in \mathcal{X}} P_S(x)dx > 1, \tag{20}$$

which arrives at a contradiction. So Eq (16) holds. We then construct Z as

$$P_Z(x) = \begin{cases} C, \text{ if } x \in \mathcal{J} \cup \mathcal{J}_{\mathcal{M}}, \\ 0, \text{ otherwise .} \end{cases} \tag{21}$$

With Eq (21) and Eq (16), we have

$$\int_{x \in \mathcal{J}} \left(P_Z(x) - P_S(x)\right) dx > 1 - \frac{1}{C|\mathcal{X}|}. \tag{22}$$

So we prove Eq (12).

Putting Step 1 and 2 together, we finish the proof. □

C. Offline Datasets and Implementation Details

C.1. Offline Datasets

For the benchmark tasks, offline datasets are collected by the final online policy trained with HER (Andrychowicz et al., 2017). Additional Gaussian noise with zero mean and 0.2 standard deviation and random actions with probability 0.3 is used for data collection to increase the diversity, following previous work (Yang et al., 2022b). For the FetchSlide task, we only use noise with a standard deviation of 0.1 because the behavior policy is already suboptimal. After data collection, different from (Yang et al., 2022b), we need additional data processing to select trajectories whose achieved goals are all in the IID region. The IID region is defined by each task group, which is shown in Table 3. A special case is the HandReach task, where we do not divide the dataset due to its high dimensional space and we use different scales of evaluation goals instead. Compared with prior offline GCRL works (Yang et al., 2022b; Ma et al., 2022b), we use relatively smaller datasets to study the OOD generalization problem. Our datasets encompass trajectories of different length, ranging from 200 to 20000, with each trajectory comprising 50 transitions. A comprehensive summary of this information is presented in Table 3. The dataset division standard refers to the location requirements of initial states and desired goals for IID tasks (e.g., Right2Right). For OOD tasks, the initial state or the the desired goal are designed to deviate from the IID requirement (e.g., Right2Left, Left2Right, Left2Left).

Table 3. Information about 9 Task Groups and Datasets.

Datasets (Task Group)	IID task	OOD task	Trajectory number	Size (M)	Dataset Division Standard
Reach Left-Right	Right	Left	200	1.6	the gripper's y coordinate value > the initial position
Reach Near-Far	Near	Far	200	1.6	the l_2 distance between gripper and the initial position ≤ 0.15
Push Left-Right	Right2Right	Right2Left, Left2Right, Left2Left	5000	67	the object's y coordinate value > the initial position
Push Near-Far	Near2Near	Near2Far, Far2Near, Far2Far	5000	67	the l_2 distance between the object and the initial position ≤ 0.15
Pick Left-Right	Right2Right	Right2Left, Left2Right, Left2Left	5000	67	the object's y coordinate value > the initial position
Pick Low-High	Low	High	5000	67	the object's z coordinate value < 0.6
Slide Left-Right	Right2Right	Right2Left, Left2Right, Left2Left	20000	266	the object's y coordinate value > the initial position
Slide Near-Far	Near	Far	20000	266	the object's x coordinate value ≤ 0.14
HandReach Near-Far	Near	Far	10000	429	the range of meeting position for two fingers

Table 4. w and τ used for GOAT and GOAT(τ).

Task Group	GOAT	GOTA(τ)
Reach Left-Right	$w = 1.5$	$w = 2.5, \tau = 0.3$
Reach Near-Far	$w = 2.0$	$w = 1.5, \tau = 0.1$
Push Left-Right	$w = 2.5$	$w = 1.5, \tau = 0.1$
Push Near-Far	$w = 1.5$	$w = 2.5, \tau = 0.1$
Pick Left-Right	$w = 1.0$	$w = 2.5, \tau = 0.3$
Pick Low-High	$w = 2.0$	$w = 1.5, \tau = 0.3$
Slide Left-Right	$w = 1.5$	$w = 2.5, \tau = 0.1$
Slide Near-Far	$w = 2.0$	$w = 1.5, \tau = 0.3$
HandReach Near-Far	$w = 2.5$	$w = 2.0, \tau = 0.1$

C.2. Implementation Details

Implementations Following (Yang et al., 2022b; Ma et al., 2022b), value functions and policy networks (along with their target networks) are all 3-layer MLPs with 256-unit layers and relu activations. We use a batch size of 512, a discount factor of $\gamma = 0.98$, and an Adam optimizer with learning rate 5×10^{-4} for all algorithms. We also normalize the observations and goals with estimated mean and standard deviation. The relabel probability $p_{relabel} = 1$ for most environments except for Slide Left-Right and Slide Near-Far, where $p_{relabel} = 0.2$ and 0.5, respectively. In **EAW**, the ratio β is set to 2 and EAW is clipped into range $(0, M]$ for numerical stability, where M is set to 10 in our experiments. For **DSW**, we utilize a First-In-First-Out (FIFO) queue B_a of size 5×10^4 to store recent calculated advantage values, and the percentile threshold α gradually increases from 0 to α_{max}. We use $\alpha_{max} = 80$ for all tasks except HandReach and Slide Left-Right, and $\alpha_{max} = 50$ for HandReach, $\alpha_{max} = 0$ for Slide Left-Right. When $A(s, a, g') < c$ and c is the α quantile value of B_a, we set $\epsilon(A(s, a, g')) = 0.05$ instead of 0 following (Yang et al., 2022b). For the uncertainty weight (**UW**), we use $N = 5$ ensemble Q networks to calculate the standard deviation $\text{Std}(s, g)$ and maintain another FIFO queue B_{std} to store recent

$\text{Std}(s, g)$ values. The $\text{Std}(s, g)$ values are then normalized to $[0, 1]$ with the maximum and minimum values in B_{std}. Besides, w_{min} is set to 0.5 and w is searched from $\{1, 1.5, 2, 2.5\}$. In our experiments, we still find w is unstable for different tasks. It is therefore necessary to develop a more stable uncertainty weight estimation method with less hyperparameter tuning in the future. Regarding the expectile regression (**ER**), we search $\tau \in \{0.1, 0.3\}$ because empirical results in Appendix D.5 shows that $\tau \in \{0.1, 0.3\}$ performs the best. The hyperparameters w and τ for GOAT and GOAT(τ) are listed in Table 4.

Baseline Descriptions In our experiments, all the baselines share the same policy and value network structures, as well as hyperparameters. As regards to WGCSL (Yang et al., 2022b) and GoFAR (Ma et al., 2022b), we use their official implementations. Denote the original dataset as D and the relabeled dataset as $D_{relabel}$. In the following part, we introduce several baselines used in our paper.

- **WGCSL**: using relabeled offline samples to maximize

$$J_{WGCSL}(\pi_\theta) = \mathbb{E}_{(s_t,a_t,\phi(s_i))\sim D_{relabel}}[w_{t,i} \cdot \log \pi_\theta(a_t|s_t, \phi(s_i))]$$

  where $w_{t,i} = \gamma^{i-t} \exp_{clip}(\beta A(s_t, a_t, \phi(s_i))) \cdot \epsilon(A(s_t, a_t, \phi(s_i)))$. In our paper, since we find DRW (i.e., γ^{i-t}) is not useful for OOD generalization (see Appendix D.9), we just set DRW=1. Besides, β is set to 2 as GOAT.

- **GoFAR**:

$$J_{\text{GoFAR}}(\pi_\theta) = \mathbb{E}_{(s_t,a_t,g)\sim D}[\log \pi_\theta(a_t|s_t, g) \max(A(s, a, g) + 1, 0)],$$

  where the advantage function is estimated by discriminator-based rewards. The discriminator c is learned to minimize $\mathbb{E}_{g\sim p(g)}[\mathbb{E}_{p(s;g)}\big[\log c(s, g)\big] + \mathbb{E}_{(s,g)\sim D}[\log(1 - c(s, g))]\big]$. The value function V is learned to minimize $(1-\gamma)\mathbb{E}_{(s,g)\sim\mu_0,p(g)}[V(s, g)] + \frac{1}{2}\mathbb{E}_{(s,a,g,s')\sim D}[(r(s; g) + \gamma V(s'; g) - V(s; g) + 1)^2]$ for $V \geq 0$.

- **BC**: behavior cloning on original offline samples to maximize $J_{BC}(\pi) = \mathbb{E}_{(s_t,a_t,g)\sim D}[\log \pi(a_t|s_t, g)]$.

- **GCSL**: using relabeled offline dataset to maximize $J_{GCSL}(\pi_\theta) = \mathbb{E}_{(s_t,a_t,g')\sim D_{relabel}}[\log \pi_\theta(a_t|s_t, g')]$.

- **MARWIL+HER**: using relabeled offline dataset to maximize

$$J_{\text{MARWIL+HER}}(\pi_\theta) = \mathbb{E}_{(s_t,a_t,g')\sim D_{relabel}}[\log \pi_\theta(a_t|s_t, g') \exp(\beta A(s_t, a_t, g'))].$$

  We also clip the exponential weight to $(0, 10]$ for numerical stability. β is set to 2 similar to WGCSL and GOAT.

- **IQL+HER**: using expectile regression L_2^τ, an additional V network and weighted imitation learning to remove OOD actions from value estimation. Hyperparameters are set the same as WGCSL, MARVIL and GOAT.

$$L_V(\psi) = \mathbb{E}_{(s_t,a_t,g')\sim D_{relabel}}[L_2^\tau(Q_\phi(s_t, a_t, g') - V_\psi(s_t, g'))]$$

$$L_Q(\phi) = \mathbb{E}_{(s_t,a_t,s_{t+1},g')\sim D_{relabel}}[(r(s_t, a, g') + \gamma V_\psi(s_{t+1}, g') - Q_\phi(s_t, a_t, g'))^2]$$
$$J(\pi_\theta) = \mathbb{E}_{(s_t,a_t,g')\sim D_{relabel}}[\exp\left(\beta Q_\phi(s_t, a_t, g') - V_\psi(s, g')\right) \log \pi_\theta(a|s_t, g')]$$

- **CQL+HER**: For a fair comparison, we implement CQL on top of DDPG+HER. The objective of CQL+HER is:

$$J_{\text{CQL+HER}}(\pi_\theta) = \mathbb{E}_{(s_t,g')\sim D_{relabel}}[Q(s_t, \pi_\theta(s_t, g'), g')]$$

  The Q function of CQL+HER is learned by minimizing the following loss:

$$L_{\text{CQL+HER}} = \mathbb{E}_{(s_t,a_t,s_{t+1},g')\sim D_{relabel}}\big[(Q(s_t, a_t, g') - \mathcal{B}^\pi Q(s_t, a_t, g'))^2\big] + \alpha\mathbb{E}_{(s_t,g')\sim D_{relabel},a\sim\exp(Q)}\big[Q(s_t, a, g')\big]$$

  where $\mathcal{B}^\pi$ is the Bellman operator. α is the ratio to balance the CQL loss and the TD loss. Another baseline DDPG+HER is exactly $\alpha = 0$.

- **MSG+HER**: We implement MSG based on ensemble DDPG with $N = 5$ independent Q networks and an LCB objective. Each Q network learns to minimize the TD loss in Eq (3). The policy learns to maximize

$$J_{MSG+HER}(\pi_\theta) = \mathbb{E}_{(s_t,g')\sim D_{relabel},a\sim\pi_\theta(s_t,g')}[\frac{1}{N}\sum_{i=1}^{N} Q_i(s_t, a, g') - c \cdot \sqrt{\text{Var}(Q_1(s_t, a, g'), \ldots, Q_N(s_t, a, g'))}]$$

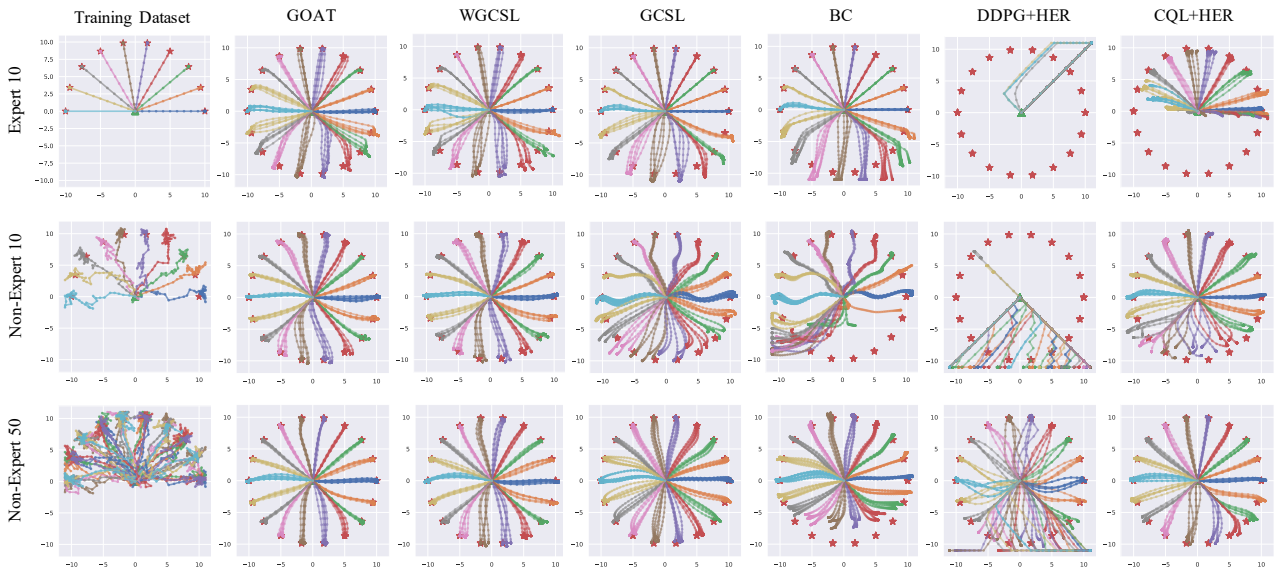


Figure 7. Evaluation of different agents on 2D PointReach task over 5 random seeds. "Expert 10" refers to the clean expert dataset, "Non-Expert N" refers to the noisy dataset with $N = 10, 50$ trajectories, respectively. Training trajectories are mainly on the upper semicircle, and the evaluation goals are on the full circle of radius 10.

Table 5. Final average success rates of different agents trained on three types of PointReach datasets. The results are averaged over 5 random seeds.

		GOAT(ours)	WGCSL (tuned)	WGCSL	GCSL	Goal BC	HER	CQL+HER
Expert 10	R10	**0.86 ± 0.08**	**0.94 ± 0.06**	0.82 ± 0.07	0.72 ± 0.09	0.67 ± 0.02	0.0 ± 0.0	0.39 ± 0.05
	R20	**0.48 ± 0.21**	0.30 ± 0.14	0.40 ± 0.19	0.03 ± 0.06	**0.45 ± 0.10**	0.0 ± 0.0	0.01 ± 0.02
Non-Expert 10	R10	**0.94 ± 0.08**	0.89 ± 0.12	**0.94 ± 0.1**	0.66 ± 0.18	0.29 ± 0.06	0.0 ± 0.0	0.52 ± 0.05
	R20	**0.69 ± 0.18**	**0.63 ± 0.19**	0.53 ± 0.1	0.12 ± 0.09	0.06 ± 0.04	0.0 ± 0.0	0.16 ± 0.06
Non-Expert 50	R10	**1.00 ± 0.00**	**0.98 ± 0.04**	0.96 ± 0.08	**0.98 ± 0.04**	0.57 ± 0.02	0.60 ± 0.30	0.84 ± 0.12
	R20	**0.92 ± 0.04**	**0.91 ± 0.14**	0.81 ± 0.14	0.33 ± 0.12	0.10 ± 0.04	0.16 ± 0.16	0.75 ± 0.08

D. Additional Experiments

In this section, we include the following experiments:

1. 2D PointReach Task;
2. Uncertainty and Density;
3. Ablation on the Ensemble Size;
4. Additional Ablations of GOAT;
5. The Effectiveness of Expectile Regression;
6. Ablations of Ensemble and HER for Other GCRL Algorithms;
7. Ablations of GoFAR;
8. Combining Weighted Imitation Learning with Value Underestimation;
9. Discounted Relabeling Weight (DRW);

10. Adaptive Data Selection Weight;
11. MSG+HER with Varying Hyper-parameter;
12. Online Fine-tuning;
13. Training Time;
14. Random Network Distillation as the Uncertainty Measurement;
15. Full Benchmark Experiments.

D.1. 2D PointReach Task

Average Success Rates and Cumulative Returns In Table 5 and Table 6, we provide average success rates and average cumulative returns of different algorithms on "Expert 10", "Non-Expert 10", and "Non-Expert 50" datasets. We also include the performance of our method GOAT for comparison. As demonstrated in the two tables, GOAT achieves the highest average success rates and average returns on all three types of training data, surpassing the strong baseline WGCSL. In addition, we visualize trajectories collected by these agents in Figure 7. The results also show that GOAT performs better than WGCSL with a smaller trajectorie variance on the lower semicircle. Other conclusions keep the same as the didactic example in Section 3.1.

Table 6. Final average returns of different agents trained on three types of PointReach datasets. The results are averaged over 5 random seeds.

		GOAT(ours)	WGCSL (tuned)	WGCSL	GCSL	Goal BC	HER	CQL+HER
Expert 10	R10	**34.69 ± 2.81**	**38.37 ± 2.36**	33.76 ± 3.07	30.23 ± 3.22	28.48 ± 0.75	0.05 ± 0.0	15.22 ± 1.16
	R20	**15.34 ± 5.35**	10.35 ± 3.96	12.99 ± 5.49	1.75 ± 2.24	**14.62 ± 2.77**	0.00 ± 0.0	0.14 ± 0.28
Non-Expert 10	R10	**37.57 ± 3.21**	35.50 ± 4.88	**37.34 ± 3.56**	22.42 ± 4.75	12.62 ± 1.89	0.21 ± 0.32	20.90 ± 1.06
	R20	**20.88 ± 5.84**	**19.27 ± 5.52**	16.19 ± 2.89	2.42 ± 2.09	1.58 ± 0.75	0.00 ± 0.00	3.93 ± 1.20
Non-Expert 50	R10	**39.99 ± 0.1**	**39.22 ± 1.46**	38.23 ± 3.29	32.84 ± 1.75	18.72 ± 0.90	24.70 ± 12.41	32.50 ± 4.33
	R20	**27.89 ± 1.1**	**27.34 ± 4.19**	23.93 ± 4.37	6.53 ± 2.41	3.65 ± 0.78	5.21 ± 4.96	21.09 ± 2.53

Table 7. Final average success rates of CQL+HER agents with different hyperparameter α. The results are averaged over 5 random seeds.

CQL+HER		$\alpha = 5$	$\alpha = 2$	$\alpha = 1$	$\alpha = 0.1$	$\alpha = 0.01$
Expert 10	R10	0.34 ± 0.07	**0.45 ± 0.03**	**0.39 ± 0.05**	0.36 ± 0.07	0.21 ± 0.08
	R20	**0.09 ± 0.08**	**0.08 ± 0.04**	0.01 ± 0.02	0.06 ± 0.06	0.06 ± 0.07
Non-Expert 10	R10	0.28 ± 0.17	**0.53 ± 0.07**	0.52 ± 0.05	**0.60 ± 0.10**	0.10 ± 0.15
	R20	0.07 ± 0.04	**0.19 ± 0.07**	0.16 ± 0.06	**0.32 ± 0.04**	0.02 ± 0.04
Non-Expert 50	R10	0.57 ± 0.04	0.76 ± 0.05	**0.84 ± 0.12**	**0.95 ± 0.10**	0.77 ± 0.21
	R20	0.21 ± 0.12	0.52 ± 0.12	**0.75 ± 0.08**	**0.66 ± 0.16**	0.46 ± 0.24

Hyper-parameter Tuning for CQL+HER It is also interesting to check whether tuning the ratio α of the CQL loss can enable better OOD generalization. Specifically, we tune α in $\{0.01, 0.1, 1, 2, 5\}$. The results are shown in Table 7. The generalization performance of CQL+HER drops as α becomes large, e.g., $\alpha = 5$, and as α becomes small, e.g., $\alpha = 0.01$. When the data coverage is insufficient (i.e., the "Expert 10" setting), CQL+HER cannot generalize on the full circle of radius 20 ("R20"), no matter how α is adjusted. The tuned results of CQL+HER are still incomparable to weighted imitation learning methods such as GOAT and WGCSL.

Additional Tasks In addition to tasks introduced before, we include another three datasets in Figure 8 (a), where we have few trajectories on the lower semicircle. As shown in Figure 8 (b), these tasks are relatively easy compared with those used in our didactic example, achieving higher average success rates. The conclusions are also consistent with Section 3.1.

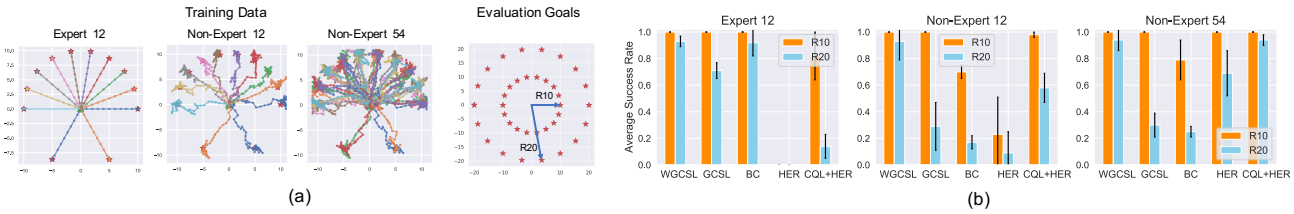


Figure 8. (a) Visualization of three 2D goal-reaching datasets, and two groups of evaluation goals, "R10" and "R20", with a radius of 10 and 20, respectively. (b) Average success rates of different agents over 5 random seeds.

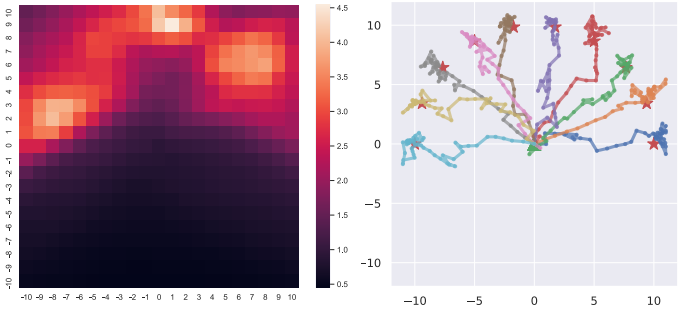


Figure 9. The reciprocal of uncertainty is an estimation of density. The uncertainty is measured by the variance of ensemble value functions for initial state $(0, 0)$ and goals on $[-10, 10] \times [-10, 10]$.

D.2. Uncertainty and Density

To verify if the estimated uncertainty can approximate density, we visualize the value of $\frac{1}{\text{Std}(s_0,g)}$ in Figure 9, where Std is the standard deviation of a group of 5 value networks. To calculate values on the figure, we set the state s_0 as $(0, 0)$ and use the value functions of GOAT to calculate each $\text{Std}(s_0, g)$, where g is set on a $[-10, 10] \times [-10, 10]$ grid with equal intervals of 1. We can observe that positions with more achieved goals (i.e., near the red stars) have larger values $\frac{1}{\text{Std}(s_0,g)}$, which validates the relationship between uncertainty and density.

Table 8. Ablation on ensemble size for GOAT over all 26 tasks.

Success Rate (%)	$N = 2$	$N = 3$	$N = 5$ (default)	$N = 7$
Average IID	90.4	90.4	90.3	90.6
Average OOD	64.0	66.7	67.9	65.3

D.3. Ablation on the Ensemble Size

The ensemble size N of GOAT can affect the value estimation and the uncertainty estimation. As shown in Table 8, though the IID performance is similar, $N = 5$ works better than $N \in \{2, 3, 7\}$ on the average OOD success rate. Note that with different N, the average OOD success rate is still better than that of WGCSL (i.e., 62.1 for OOD tasks).

D.4. Additional Ablations of GOAT

Taking into account additional design considerations, we also revisit two design choices, namely the value function and the exponential weight. Specifically, in GOAT, we adopt the approach proposed by WGCSL (Yang et al., 2022b) to learn Q functions for estimating the advantage value, which is given by $A(s_t, a_t, g) = r(s_t, a_t, g) + \gamma Q(s_{t+1}, \pi(s_{t+1}, g), g) - Q(s_t, \pi(s_t, g), g)$. The difference is that Q values are averaged by an ensemble of Q functions. Alternatively, we can estimate the advantage values by learning V functions $V(s, g)$, which can be expressed as $A(s_t, a_t, g) = r(s_t, a_t, g) + \gamma V(s_{t+1}, g) - V(s_t, g)$. Note that the learned action π is not needed in the value function learning for $V(s, g)$. We use the

Table 9. Additional ablations of GOAT. Average success rates (%) with standard deviation over 5 random seeds.

Task Group	Task	GOAT	GOAT(V)	GOAT(V+χ^2)
Reach Left-Right	Right	100.0±0.0	99.9 ± 0.2	100.0 ± 0.0
	Left	99.0±2.0	94.5 ± 7.5	96.6 ± 2.5
	Average	**99.5**	97.2	98.3
Reach Near-Far	Near	100.0±0.0	99.6 ± 0.8	100.0 ± 0.0
	Far	97.6±1.1	89.8 ± 4.5	90.6 ± 1.0
	Average	**98.8**	94.7	95.3
Push Left-Right	Right2Right	95.9±1.2	92.6 ± 3.4	93.9 ± 1.6
	Right2Left	69.3±6.6	48.9 ± 5.8	52.0 ± 5.8
	Left2Right	76.0±7.4	56.2 ± 5.1	56.6 ± 11.2
	Left2Left	61.1±7.6	39.5 ± 4.0	34.1 ± 5.1
	Average	**75.6**	59.3	59.2
Push Near-Far	Near2Near	92.0±2.6	92.2 ± 2.2	87.5 ± 1.7
	Near2Far	70.3±5.7	59.4 ± 5.1	63.8 ± 3.9
	Far2Near	69.5±3.6	65.0 ± 3.9	63.4 ± 3.7
	Far2Far	50.8±1.8	41.3 ± 2.2	43.5 ± 4.3
	Average	**70.6**	64.5	64.6
Pick Left-Right	Right2Right	97.3±1.2	82.1 ± 3.2	82.5 ± 8.3
	Right2Left	88.6±1.1	52.7 ± 9.2	51.4 ± 8.2
	Left2Right	93.9±1.9	62.4 ± 5.2	64.1 ± 15.7
	Left2Left	88.3±3.7	43.8 ± 13.4	42.6 ± 9.8
	Average	**92.0**	60.3	60.2
Pick Low-High	Low	99.8±0.2	99.1 ± 0.9	98.9 ± 1.0
	High	71.9±6.4	17.5 ± 2.5	24.8 ± 4.9
	Average	**85.8**	58.3	61.8
Slide Left-Right	Right2Right	79.0±5.8	77.5 ± 6.7	71.8 ± 5.0
	Right2Left	41.3±7.1	45.7 ± 6.3	40.5 ± 7.8
	Left2Right	59.0±7.6	52.2 ± 10.9	31.1 ± 7.8
	Left2Left	50.1±9.5	41.9 ± 8.7	30.9 ± 4.9
	Average	**57.4**	54.3	43.6
Slide Near-Far	Near	76.9±3.3	73.2 ± 4.8	72.7 ± 9.0
	Far	29.0±4.5	26.4 ± 4.1	32.0 ± 4.4
	Average	**53.0**	49.8	52.4
HandReach Near-Far	Near	71.9±3.2	74.4 ± 2.9	76.8 ± 2.2
	Far	38.4±4.1	37.3 ± 6.5	43.1 ± 4.2
	Average	55.2	55.9	**60.0**
Average	IID Tasks	**90.3**	87.8	87.1
	OOD Tasks	**67.9**	51.4	50.7

notation "GOAT(V)" to describe this variant of GOAT with V functions. Furthermore, the exponential advantage weight is a special case of maximizing expected value and minimizing the f-divergence. Following (Ma et al., 2022b), we also consider replacing the exponential weight with $\max(A(s,a,g)+c_{\chi^2},0)$, which is derived by considering the χ^2-divergence. We denote GOAT with both V function and χ^2-divergence as "GOAT(V+χ^2)". After the parameter search, we find $c_{\chi^2}=0$ performs well for GOAT(V+χ^2).

The final results are reported in Table 9. Our results indicate that, on average across the 17 out-of-distribution tasks, the use of a V function in GOAT significantly reduces the OOD generalization performance, with negligible impact from the χ^2-divergence. It is worth noting, however, that for the high-dimensional HandReach task, the best performance is achieved by incorporating both a V function and χ^2-divergence into GOAT. The rationale behind this finding lies in the high-dimensional nature of the state-goal and action spaces in the HandReach task. As a result of this high dimensionality, the multimodal problem is more pronounced, rendering the learning of a V function useful for achieving better stability than the Q function. Moreover, the integration of a weighting function induced by the χ^2-divergence serves to eliminate inferior data and also alleviate the multimodal problems arising in such high-dimensional spaces. Therefore, the choice of value function and weighting function also depends on the task characteristics.

D.5. The Effectiveness of Expectile Regression

Kostrikov et al. (2021) proposed IQL to combine expectile regression with weighted imitation learning in offline RL. The difference is that we do not learn additional V function to avoid OOD actions when learning value functions and we validate its effectiveness for improving advantage value estimation of offline GCRL. As shown in Figure 10, expectile regression (ER) improves both WGCSL and GOAT by around 3 points on average OOD success rates. Besides, in Figure 11 we show

Figure 10. Ablation of Expectile Regression

Success Rate (%)	WGCSL	WGCSL+ER	GOAT	GOAT+ ER
Average	71.1	73.6	75.7	77.9
Average OOD	62.1	65.1	67.9	70.9

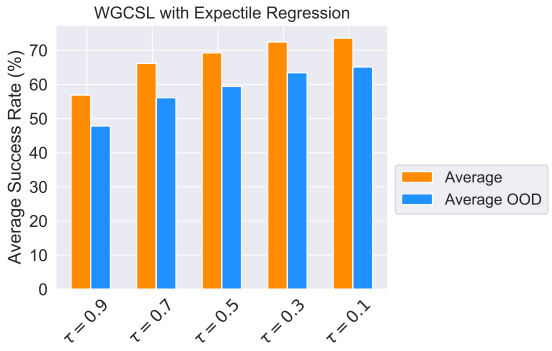
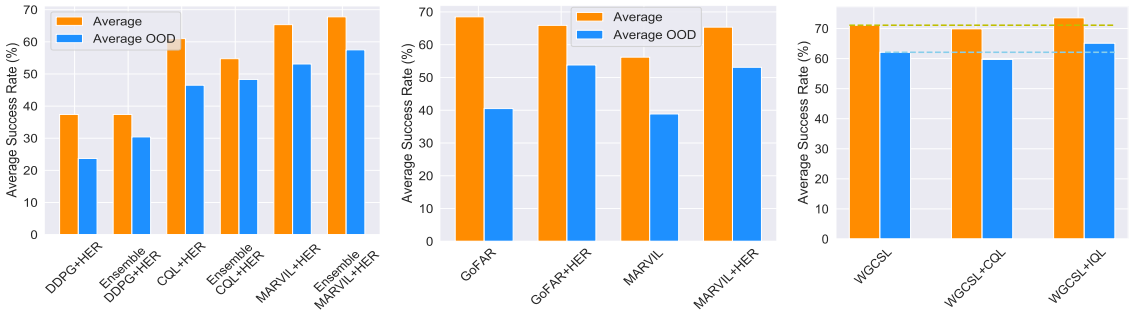


Figure 11. Comparing differnt τ in Expectile Regression for Weighted Imitation Learning.

that the performance of WGCSL+ER improves with the decrease of τ. We conjecture that the smaller τ is, the more accurate the relative relationship of the advantage values are, and thus the better the estimation of the expert policy for imitation.



(a) (b) (c)

Figure 12. (a) Comparison between methods with and without value function ensemble. (b) Comparison of GoFAR and MARVIL with and without HER. (c) Comparison between WGCSL and WGCSL with value underestimation.

D.6. Ablations of Ensemble and HER for Other GCRL Algorithms

In Figure 12(a) and Figure 12(b), we demonstrate that ensemble value functions and HER can improve the performance of different RL algorithms, indicating that they are generally useful techniques for OOD generalization of offline GCRL.

Table 10. Ablations of GoFAR on the HandReach task. Average success rates (%) with standard deviation over 5 random seeds.

Task Group	Task	GoFAR	GoFAR(binary)	GoFAR(binary+Q)	GoFAR(binary+exp)	GoFAR(binary+relabel)	GoFAR(binary+exp+relabel)
HandReach Near-Far	Near	77.4±1.7	78.9±3.3	4.9±5.0	10.8±2.7	62.0±7.9	57.1±7.4
	Far	36.9±3.1	39.2±4.3	3.1±1.8	2.6±1.4	29.3±8.2	24.0±4.2
	Average	57.1	59.0	4.0	6.7	45.7	40.5

D.7. Ablations of GoFAR

GoFAR (Ma et al., 2022b) is a recent offline GCRL method which is also based on weighted imitation learning, but we observe its performance drops significantly on OOD tasks. We conjecture the primary reason is that GoFAR does not use goal relabeling. To validate our conjecture, we compare GoFAR and GoFAR+HER in Figure 12(b). The results show that HER does not improve the overall performance of GoFAR but increases the average OOD success rates by a large margin. In Figure 12(b), the performance increment of GoFAR+HER over GoFAR also matches that of MARVIL+HER over MARVIL.

In our benchmark experiments, we observe that GoFAR has an advantage on the high-dimensional HandReach task over

WGCSL and GOAT. Given this observation, we investigate and analyze the roles played by the key techniques of GoFAR's design, namely the discriminator-based rewards, the advantage estimation, and the weighting function. We subsequently compare the following variants:

- GoFAR: it employs discriminator-based rewards, learns $V(s, g)$ function to estimate the advantage value $A(s_t, a_t, g) = r(s_t, a_t, g) + \gamma V(s_{t+1}, g) - V(s_t, g)$, weights the imitation loss with $\max(A(s_t, a_t, g) + 1, 0)$, and does not use goal relabeling;
- GoFAR(binary): it replaces the discriminator-based rewards with binary rewards $r(s_t, a_t, g) = 1[\|\phi(s_t) - g\|_2^2 \leq \epsilon]$;
- GoFAR(binary+Q): it uses binary rewards to learn $Q(s, a, g)$ functions instead of V functions, then the advantage value are estimated by $A(s_t, a_t, g) = r(s_t, a_t, g) + \gamma Q(s_{t+1}, \pi(s_{t+1}, g), g) - Q(s_t, \pi(s_t, g), g)$;
- GoFAR(binary+exp): it also employs binary rewards to learn V functions, but includes a weighting term of $\exp(A(s, a, g))$ based on the KL divergence;
- GoFAR(binary+relabel): it uses binary rewards and goal relabeling for GoFAR;
- GoFAR(binary+exp+relabel): it utilizes binary rewards and goal relabeling to learn V functions, and the exponential weighting function $\exp(A(s, a, g))$ for weighted imitation learning.

Table 10 presents the findings of this study, which suggest that the value function and weighting function are the most crucial components of GoFAR for the high-dimensional HandReach task. "GoFAR(binary+Q)" and "GoFAR(binary+exp)" both fail on this task. We posit that this maybe due to the following reasons: (1) with high-dimensional state-goal and action spaces, the Q value function has more dimensions as input than the V function and is therefore less stable and harder to train. Additionally, the learned policy π via weighted imitation learning is prone to interpolation into out-of-distribution actions, which causes imprecise advantage value estimation using $Q(s, \pi(s, g), g)$. (2) The weighting function $\max(A(s, a, g)+1, 0)$ also serves the purpose of clearing poor quality data from our weighted imitation learning, whereas the exponential weighting function is more sensitive to the multimodal problem. Furthermore, our results indicate that the discriminator-based rewards play no significant role and may even decrease performance when compared to binary rewards. It is also worth noting that the effectiveness of goal relabeling varies with the type of weighting function. While it diminishes performance for $\max(A(s, a, g) + 1, 0)$, it enhances the performance of weighting with $\exp(A(s, a, g))$.

D.8. Combining Weighted Imitation Learning with Value Underestimation

The value function learning has an impact on the estimation of the expert policy for imitation. It is interesting to see whether simply underestimating values for weighted imitation learning is also helpful for OOD generalization. In Figure 12(c), we consider two methods to be on top of WGCSL, CQL (Kumar et al., 2020), and IQL (Kostrikov et al., 2021) (specifically, the expectile regression technique). We demonstrate that while CQL is not helpful for WGCSL, while the expectile regression in IQL is a good choice for better value function estimation of offline GCRL.

D.9. Discounted Relabeling Weight (DRW)

Yang et al. (2022b) introduced the Discounted Relabeling Weight (DRW) for offline GCRL. For a relabeled transition $(s_t, a_t, \phi(s_i)), i \geq t$, DRW is defined as γ^{i-t}, which has an effect of optimizing a tighter lower bound for offline GCRL. Intuitively, DRW assigns relatively larger weights on closer relabeling goals with smaller i. In Table 11, we find DRW can slightly improve the IID performance but reduce the average OOD performance of WGCSL. It is reasonable because DRW assigns larger weights for closer goals, which contradicts the Uncertainty Weight as discussed in Section 5.2 and has the risk of overfitting simpler goals.

Table 11. Comparison of WGCSL and WGCSL+DRW on 26 tasks.

Success Rate (%)	WGCSL	WGCSL+DRW
Average IID	88.1	88.4
Average OOD	62.1	59.5

D.10. Adaptive Data Selection Weight

Yang et al. (2022b) introduced the Data Selection Weight (DSW) to tackle the multi-modal problem in multi-goal datasets, which also improves the OOD generalization performance through narrowing the expert estimation gap. However, the introduced approach utilizes a global threshold for all (s, g) pairs. **Is it helpful to include an adaptive threshold function for different (s, g) pairs?** This can be done using expectile regression for advantage values, i.e., learning a function $f(s, g)$ to estimate the β expectile value of the distribution of $A(s, g, a)$.

$$\mathcal{L}_f = \mathbb{E}_{(s,a,g)\sim D_{relabel}}[L_2^{\beta}(A(s, a, g) - f(s, g))]$$

β is the hyper-parameter similar to α in the original DSW controlling the quality of data used for weighted imitation learning. Finally, the adaptive data selection weight is $\epsilon(A(s, a, g)) = 1[A(s, a, g) \geq f(s, g)]$. We compare WGCSL and WGCSL with adaptive data selection weight (ADSW) on Push and Pick task groups. As shown in Table 12, WGCSL with ADSW only brings slight improvement over global threshold. For most tasks, WGCSL also obtains top two scores with a simpler data selection method. Considering the extra computation of learning the threshold function f, we do not include it in GOAT. But we believe it is a good start for future research on how to achieve more efficient adaptive data selection for offline goal-conditioned RL.

Table 12. Comparison with Adaptive Data Selection Weight (ADSW). Average success rates (%) with standard deviation over 5 random seeds. Top two scores for each task are highlighted.

Task Group	Task	WGCSL	ADSW, $\beta = 0.8$	ADSW, $\beta = 0.9$	ADSW, $\beta = 0.95$
Push Left-Right	Right2Right	93.2±0.9	95.1±1.4	95.5±0.8	92.8±1.9
	Right2Left	63.3±8.9	65.3±4.7	69.5±6.4	62.5±7.4
	Left2Right	67.6±7.1	74.9±8.6	74.6±6.0	64.3±13.1
	Left2Left	47.7±7.4	59.8±3.7	66.4±5.9	52.1±6.9
	Average	68.0	**73.8**	**76.5**	67.9
Push Near-Far	Near2Near	93.5±1.0	90.7±2.8	93.4±1.0	91.9±1.2
	Near2Far	67.0±5.4	68.1±4.2	69.4±3.8	63.4±5.0
	Far2Near	68.0±2.4	66.1±2.6	67.6±2.3	62.1±4.0
	Far2Far	51.1±4.7	47.4±3.4	53.1±1.9	40.2±6.1
	Average	**69.9**	68.1	**70.9**	64.4
Pick Left-Right	Right2Right	93.8±5.3	96.6±2.6	94.4±4.1	96.0±2.0
	Right2Left	89.4±3.9	83.5±6.4	71.1±9.4	78.9±11.2
	Left2Right	90.0±4.1	89.8±4.8	91.6±4.7	89.3±2.8
	Left2Left	87.0±5.1	83.0±5.7	73.3±8.6	78.6±8.4
	Average	**90.0**	**88.2**	82.6	85.7
Pick Low-High	Low	98.6±1.3	99.8±0.2	99.1±0.6	99.4±0.6
	High	66.6±6.6	63.4±5.1	66.8±13.5	63.5±8.6
	Average	**82.6**	81.6	**83.0**	81.4
Average	IID Tasks	94.8	**95.6**	**95.6**	95.0
	OOD Tasks	69.8	**70.1**	**70.3**	65.5

Table 13. Varying hyper-parameter of MSG over all 26 tasks.

Success Rate (%)	$c = 1$	$c = 3$	$c = 5$	$c = 7$
Average IID	60.1	65.4	68.5	69.9
Average OOD	41.0	42.4	43.1	39.4

D.11. MSG+HER with Varying Hyper-parameter

MSG (Ghasemipour et al., 2022) is a recent SOTA ensemble-based offline RL method, which learns a group of independent Q networks and estimates a Lower Confidence Bound (LCB) objective with the standard deviation (std) of Q networks. One important hyper-parameter for MSG+HER is the weight parameter c for the std (see Appendix C.2). We compare the performance of MSG+HER with varying $c \in \{1, 3, 5, 7\}$ in Table 13. The results demonstrate that $c = 5$ achieves the best OOD generalization results. When c is larger than 5 (i.e., $c = 7$), the agent is too conservative to generalize, leading to improvement on IID tasks but decrease on OOD tasks. GOAT still outperforms MSG+HER by a large margin, which also supports that pessimism-based offline RL method can inhibit generalization.

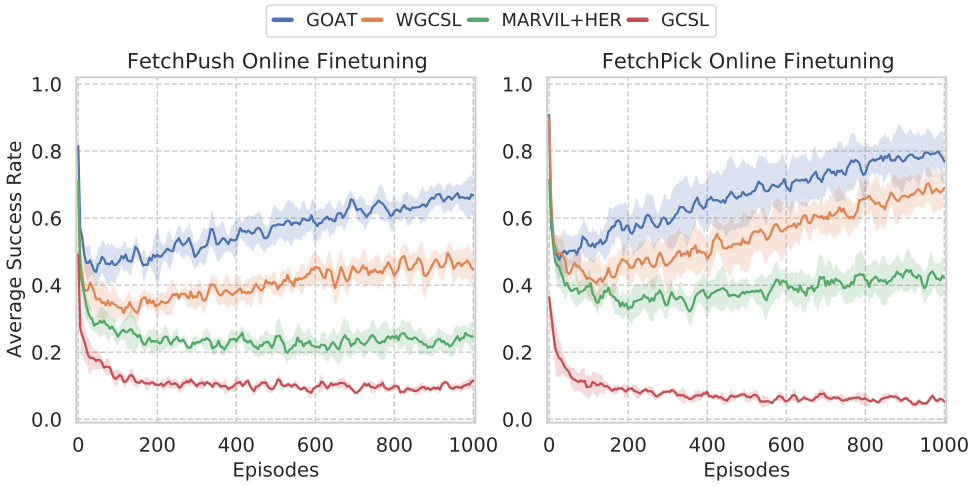


Figure 13. Online fine-tuning of different supervised learning methods in FetchPush Left-Right and FetchPick Left-Right tasks.

D.12. Online Fine-tuning

To understand the effect of the generalization ability of pre-trained agents for online learning, we design an experiment to fine-tune pre-trained agents with online samples. The pre-trained agents are trained on offline datasets with partial coverage (e.g., Right2Right) and evaluated with full coverage goals (Right2Right, Right2Left, Left2Right, Left2Left). In the fine-tuning period, agents explore with additional Gaussian noise (zero mean and 0.2 standard deviation) and random actions (with a probability of 0.3). For all pre-trained agents, we fine-tune the policy and value function for 10 (FetchPick) or 20 (FetchPush) batches after every a trajectory collected. The training batch size is 512, the learning rate is 5×10^{-4}, and the optimizer is Adam. Note that we do not use offline datasets during online fine-tuning, and we fine-tune agents to goals not seen in the pre-training phase, which is different from prior offline-to-online setting (Nair et al., 2020b).

For the first experiment in Figure 6, all pre-trained agents are fine-tuned with DDPG+HER (Andrychowicz et al., 2017), which is a general baseline in the online setting. In addition to DDPG+HER, we apply different supervised learning methods, i.e, GOAT, WGCSL, MARVIL+HER, GCSL for online fine-tuning in Figure 13. The fine-tuning algorithms are the same as their pre-training algorithms. Other settings are kept the same as the above fine-tuning experiments. Comparing Figure 13 with Figure 6, we can also conclude that (1) off-policy method (i.e., DDPG+HER) is more efficient than supervised methods for online fine-tuning, (2) GOAT substantially outperforms other supervised methods such as WGCSL and MARVIL+HER when fine-tuned using their respective pre-training algorithms.

D.13. Training Time

We consider the training time as a measure of computational cost in Figure 14. For our experiments, we use one single GPU (NVIDIA GeForce RTX 2080 Ti 11 GB) and one cpu core (Intel Xeon W-2245 CPU @ 3.90GHz). Among all the algorithms, BC and GCSL require the least training time due to their simplicity, but they suffer to generalize given non-expert datasets. WGCSL, MARVIL+HER and DDPG+HER need more training time because they are equipped with additional Q networks. Besides, MSG leverages an ensemble of Q networks and averages their gradients to the policy, leading to the longest training time. CQL+HER requires the second longest training time because of the OOD action sampling and the logsumexp approximation procedures. Though GoFAR is also a weighted imitation method similar to WGCSL, it is the second slowest method because it uses additional discriminator for reward estimation and it is implemented based on Torch. GOAT introduces ensemble networks, expectile regression and uncertainty estimation on top of WGCSL, thus increasing the computational cost. Thanks to the efficient implementation based on tensorflow, GOAT is still more efficient than CQL+HER, and GoFAR, and requires affordable computational cost.

D.14. Random Network Distillation as the Uncertainty Measurement

The uncertainty weight is an empirical instance to estimate the density under our framework. Random Network Distillation (RND) (Burda et al., 2018) is a potential alternative for density estimation in the continuous state space. We implement

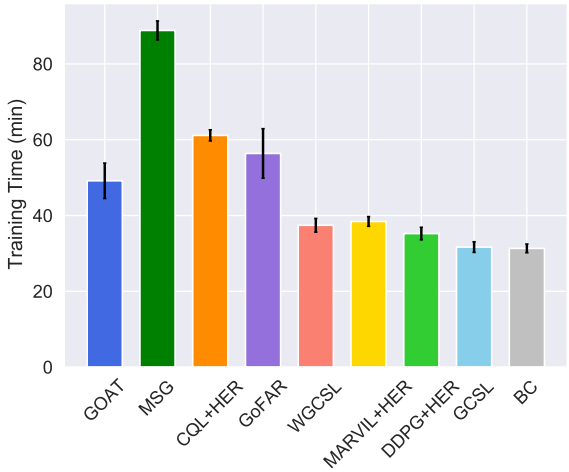


Figure 14. Comparison of training time on FetchPush task.

a RND version of GOAT for a comparison and use the same hyperparameter search range. In Table 14, GOAT(RND) outperforms WGCSL, but is worse than the version with ensemble Q functions, which may be because naive applications of RND do not yield good results for vector-input tasks (Rezaeifar et al., 2022; Nikulin et al., 2023).

Table 14. Average success rates (%) with standard deviation over 5 random seeds.

Tasks	GOAT	GOAT(RND)	WGCSL
Reach Left-Right	99.5	**99.7**	98.9
Reach Near-Far	**98.8**	95.1	94.5
Push Left-Right	75.6	**76.3**	68.0
Push Near-Far	**70.6**	69.2	69.9
Pick Left-Right	**92.0**	91.6	90.0
Pick Low-High	**85.8**	85.1	82.6
Slide Left-Right	**57.4**	53.1	48.3
Slide Near-Far	**53.0**	49.3	45.2
HandReach Near-Far	**55.2**	53.9	50.9
Average IID Tasks	**90.3**	89.4	88.1
Average OOD Tasks	**67.9**	66.0	62.1

D.15. Full Benchmark Experiments

As demonstrated in Table 15 and Table 16, we include more baselines (i.e., IQL+HER and MARVIL+HER) and additional measure (i.e., average cumulative return) for the benchmark experiments. GOAT achieves the highest average success rate and average return on the benchmark. Other conclusions are consistent with Section 5.3.

E. Additional Related Works

In ML community, there are different types of OOD studied by prior works (Nair et al., 2020a; Ma et al., 2022a; Han et al., 2021; Hansen-Estruch et al., 2022; Pitis et al., 2022), e.g., handling spurious feature, assuming Factored MDPs, or learning generalizable representations for different objects or scenes. Different from these works, our work focus on the OOD goal generalization problem, which is essentially a type of covariate shift. In practical applications, more than one type of OOD is generally involved. The work (Hong et al., 2022) studies a similar goal generalization setting of our work, but it is in the online setting with exploration. Instead, we consider the offline setting, where online interaction is prohibited and commonly used pessimism-based method can inhibit OOD generalization.

Table 15. Average success rates (%) with standard deviation over 5 random seeds. Blue lines and purple lines refer to IID and OOD tasks, respectively. Top two scores for each task are highlighted.

Task Group	Task	GOAT(τ)	GOAT	WGCSL	GCSL	BC	GoFAR	MARVIL+HER	IQL+HER	DDPG+HER	CQL+HER	MSG+HER
Reach Left-Right	Right	100.0±0.0	100.0±0.0	100.0±0.0	93.6±4.3	92.0±3.0	100.0±0.0	99.9±0.2	100.0±0.0	99.6±0.6	100.0±0.0	99.4±0.6
	Left	99.9±0.2	99.0±2.0	97.8±4.4	36.3±10.9	30.4±15.2	54.2±9.3	75.9±18.6	89.2±5.1	73.8±27.6	94.5±6.3	85.6±15.7
	Average	**99.9**	**99.5**	98.9	65.0	61.2	77.1	87.9	94.6	86.7	97.2	92.5
Reach Near-Far	Near	100.0±0.0	100.0±0.0	100.0±0.0	79.7±3.0	85.3±4.3	100.0±0.0	99.8±0.4	100.0±0.0	95.9±2.0	100.0±0.0	98.6±2.8
	Far	90.9±1.5	97.6±1.1	89.0±2.1	33.5±5.5	37.9±9.7	85.0±1.9	75.4±3.9	84.2±4.0	66.8±6.9	88.0±2.1	77.8±9.7
	Average	**95.4**	**98.8**	94.5	56.6	61.6	92.5	87.6	92.1	81.4	94.0	88.2
Push Left-Right	Right2Right	96.2±1.2	95.9±1.2	93.2±0.9	82.1±3.7	78.9±3.8	95.9±1.4	95.0±1.8	97.2±1.5	60.1±6.0	83.3±2.7	92.8±0.9
	Right2Left	75.6±3.6	69.3±6.6	63.3±8.9	40.1±6.0	25.6±2.7	43.8±4.7	64.9±8.5	67.7±8.8	28.5±4.3	46.2±7.1	52.9±6.5
	Left2Right	78.8±6.8	76.0±7.4	67.6±7.1	38.8±6.8	33.5±8.1	59.7±4.3	67.1±2.9	71.4±9.1	20.6±11.5	40.4±12.1	59.3±7.7
	Left2Left	75.6±12.1	61.1±7.6	47.7±7.4	35.4±6.6	20.9±3.2	32.5±5.8	57.9±4.9	58.7±4.2	27.0±3.8	34.9±5.9	38.8±7.9
	Average	**81.5**	**75.6**	68.0	49.1	39.7	58.0	71.2	73.8	34.1	51.2	61.0
Push Near-Far	Near2Near	97.2±0.7	92.0±2.6	93.5±1.0	77.6±4.7	67.5±3.6	92.6±2.2	89.4±3.0	96.0±0.9	39.3±22.4	77.7±3.9	84.7±6.1
	Near2Far	78.4±3.5	70.3±5.7	67.0±5.4	43.1±7.2	24.9±5.9	60.9±3.8	42.4±6.4	74.0±2.6	30.5±12.1	60.0±6.2	58.4±2.1
	Far2Near	70.5±2.4	69.5±3.6	68.0±2.4	47.4±3.5	40.2±7.5	65.0±4.8	57.9±2.6	68.8±4.1	25.0±12.8	61.1±4.3	56.5±6.0
	Far2Far	55.1±2.4	50.8±1.8	51.1±4.7	27.9±4.1	15.3±2.7	41.3±3.1	25.4±4.9	48.8±4.1	18.0±7.0	47.1±2.4	41.7±5.4
	Average	**75.3**	70.6	69.9	49.0	37.0	65.0	53.8	**71.9**	28.2	61.5	60.3
Pick Left-Right	Right2Right	96.5±1.1	97.3±1.2	93.8±5.3	53.4±14.1	52.9±7.5	56.9±4.3	91.5±2.6	88.8±5.4	40.4±13.1	91.9±6.8	94.9±2.2
	Right2Left	87.9±5.1	88.6±1.1	89.4±3.9	20.7±6.9	5.6±2.1	9.3±1.8	58.4±8.0	65.2±11.9	52.7±14.9	82.4±12.6	89.3±6.8
	Left2Right	91.4±2.3	93.9±1.9	90.0±4.1	47.0±10.9	37.2±6.4	51.1±6.5	85.2±4.2	80.2±4.0	9.8±5.7	86.4±8.6	60.8±16.5
	Left2Left	87.6±5.7	88.3±3.7	87.0±5.1	24.7±7.8	3.3±1.4	6.0±2.0	50.7±8.8	60.7±9.7	26.4±10.9	83.5±9.1	66.9±7.0
	Average	**90.8**	**92.0**	90.0	36.4	24.8	30.8	71.4	73.7	32.3	86.1	78.0
Pick Low-High	Low	99.3±0.5	99.8±0.2	98.6±1.3	84.4±3.6	72.4±5.4	95.2±1.6	98.9±0.6	98.0±0.5	50.4±23.9	100.0±0.0	97.3±2.2
	High	78.3±6.3	71.9±6.4	66.6±6.6	28.4±6.9	3.0±1.6	7.6±3.1	64.5±9.2	58.0±12.0	17.0±10.2	44.6±9.2	23.3±7.8
	Average	**88.8**	**85.8**	82.6	56.4	37.7	51.4	81.7	78.0	33.7	72.3	60.3
Slide Left-Right	Right2Right	82.0±3.2	79.0±5.8	70.8±13.5	62.2±7.0	60.3±4.7	62.6±8.7	76.5±3.1	76.7±3.6	4.7±1.5	20.3±2.5	20.8±5.0
	Right2Left	45.1±8.8	41.3±7.1	36.2±8.6	11.5±2.0	15.7±6.0	31.6±3.9	43.5±6.2	43.8±4.6	0.3±0.4	8.6±3.0	7.3±4.9
	Left2Right	79.6±2.7	59.0±7.6	50.7±12.7	29.1±4.8	41.8±7.2	51.0±10.5	55.5±5.7	71.6±3.1	0.2±0.2	1.7±0.7	3.6±4.3
	Left2Left	52.5±8.3	50.1±9.5	35.3±11.3	25.5±5.4	33.7±10.6	28.2±2.6	39.3±7.5	43.9±3.8	2.1±1.1	4.3±2.5	7.1±3.3
	Average	**64.8**	57.4	48.3	32.1	37.9	43.4	53.7	**59.0**	1.8	8.7	9.7
Slide Near-Far	Near	77.4±4.5	76.9±3.3	73.1±5.8	28.0±7.1	26.6±8.3	69.3±2.8	73.7±6.2	80.2±3.2	11.3±4.5	43.5±3.3	28.3±9.5
	Far	25.1±3.9	29.0±4.5	17.4±3.2	0.0±0.0	0.0±0.0	24.1±2.9	10.8±3.6	6.8±1.3	4.4±3.7	7.4±3.8	2.6±1.4
	Average	**51.2**	**53.0**	45.2	14.0	13.3	46.7	42.2	43.5	7.8	25.5	15.4
HandReach Near-Far	Near	72.6±5.3	71.9±3.2	70.0±3.6	0.0±0.0	0.0±0.0	77.4±1.7	72.2±4.0	67.2±9.8	0.0±0.0	1.8±3.6	0.0±0.0
	Far	33.1±4.5	38.4±4.1	31.8±3.8	0.1±0.2	0.0±0.0	36.9±3.1	28.3±6.1	27.5±3.8	0.0±0.0	0.0±0.0	0.0±0.0
	Average	52.8	**55.2**	50.9	0.0	0.0	**57.1**	50.3	47.4	0.0	0.9	0.0
Average	IID Tasks	**91.2**	**90.3**	88.1	62.3	59.5	83.3	88.5	89.3	44.6	68.7	68.5
	OOD Tasks	**70.9**	**67.9**	62.1	28.8	21.7	40.5	53.1	60.0	23.7	46.5	43.1

Table 16. Average cumulative return with standard deviation over 5 random seeds. Blue lines refer to IID tasks and purple lines indicate OOD tasks. Top two scores for each task are highlighted.

Task Group	Task	GOAT(τ)	GOAT	WGCSL	GCSL	BC	GoFAR	MARVIL+HER	IQL+HER	DDPG+HER	CQL+HER	MSG+HER
Reach Left-Right	Right	46.5±0.1	46.4±0.0	46.5±0.0	39.8±1.9	40.4±2.0	46.7±0.2	45.1±0.3	45.8±0.1	46.7±0.3	46.1±0.2	46.6±0.3
	Left	46.2±0.2	45.8±0.9	45.1±1.9	16.1±4.3	14.0±6.5	25.5±4.3	33.8±8.4	40.7±2.5	34.8±12.3	42.9±3.0	40.1±7.3
	Average	**46.4**	**46.1**	45.8	28.0	27.2	36.1	39.5	43.3	40.8	44.5	43.3
Reach Near-Far	Near	46.7±0.1	46.7±0.0	46.7±0.1	32.6±1.1	35.9±1.4	47.0±0.2	45.4±0.2	45.9±0.1	45.6±1.2	46.4±0.1	46.7±1.1
	Far	39.4±0.8	43.1±0.8	38.6±1.0	10.5±1.6	14.5±2.8	37.1±1.0	31.2±1.6	35.4±1.9	30.6±2.7	37.8±1.0	35.1±4.2
	Average	**43.0**	**44.9**	42.6	21.5	25.2	42.0	38.3	40.6	38.1	42.1	40.9
Push Left-Right	Right2Right	39.4±0.5	38.9±1.0	38.0±0.3	28.5±1.8	26.8±2.2	39.1±0.9	37.7±0.5	39.2±0.5	25.0±2.1	33.9±1.5	37.8±0.6
	Right2Left	27.2±1.4	24.9±2.4	22.6±3.0	11.5±1.8	7.5±1.0	14.8±1.4	21.2±2.5	23.0±3.4	10.4±2.1	16.9±2.6	19.6±2.7
	Left2Right	25.9±3.0	24.3±2.7	21.3±2.1	10.4±2.0	8.9±2.6	18.2±2.5	20.0±1.6	21.7±3.2	6.8±3.5	12.1±3.8	18.8±3.2
	Left2Left	29.3±5.0	23.0±2.8	18.8±3.2	12.6±1.7	8.5±1.4	12.7±1.8	21.5±1.8	21.4±1.8	11.4±1.4	14.2±2.3	15.8±2.6
	Average	**30.5**	**27.8**	25.2	15.7	12.9	21.2	25.1	26.3	13.4	19.2	23.0
Push Near-Far	Near2Near	37.8±0.6	34.9±1.3	35.9±0.2	25.6±2.1	21.6±1.4	36.1±0.9	35.6±0.8	36.6±0.7	13.9±8.0	29.2±1.9	31.4±3.0
	Near2Far	27.9±1.2	25.3±2.1	24.1±2.2	12.8±2.4	7.2±1.3	20.9±1.6	23.1±1.3	25.5±1.2	11.0±3.8	21.4±2.4	21.0±0.9
	Far2Near	22.9±0.9	23.0±1.4	22.4±1.1	12.7±1.5	10.2±1.9	21.1±1.7	20.5±1.1	22.1±1.3	7.8±4.0	20.3±1.5	18.1±1.9
	Far2Far	17.3±0.5	16.2±0.6	16.4±1.4	7.5±1.5	4.3±0.7	12.8±1.3	14.2±0.9	14.9±1.0	6.3±2.2	15.3±1.0	13.5±2.3
	Average	**26.5**	**24.9**	24.7	14.7	10.8	22.7	23.4	24.8	9.8	21.6	21.0
Pick Left-Right	Right2Right	36.8±0.2	36.7±0.6	36.1±1.0	18.5±4.6	16.6±2.6	24.0±2.1	34.0±0.4	32.3±1.3	15.2±4.4	36.1±1.7	35.9±0.8
	Right2Left	32.6±1.8	32.3±0.7	32.8±1.2	6.6±2.5	1.3±0.5	3.2±0.7	22.8±2.0	23.4±2.9	19.6±5.2	32.0±4.0	32.7±2.3
	Left2Right	32.5±1.6	33.4±0.6	32.6±0.8	14.7±3.1	10.7±1.3	18.7±2.4	28.2±1.3	27.1±1.4	2.9±1.8	30.8±2.7	19.2±5.3
	Left2Left	32.5±2.3	32.3±1.5	31.7±2.1	8.6±2.6	1.2±0.4	2.2±0.9	20.3±2.4	21.9±3.3	8.6±3.3	31.9±2.4	23.1±2.1
	Average	**33.6**	**33.7**	33.3	12.1	7.5	12.0	26.3	26.2	11.6	32.7	27.7
Pick Low-High	Low	40.0±0.1	40.2±0.2	39.8±0.3	24.9±1.3	24.1±1.6	38.8±0.6	38.0±0.3	37.4±0.3	19.4±9.1	40.9±0.2	39.5±1.0
	High	28.2±2.3	26.2±2.4	23.9±2.2	8.4±1.3	0.7±0.3	2.5±0.8	21.9±2.9	20.0±4.4	6.0±3.7	16.1±2.9	8.7±2.8
	Average	**34.1**	**33.2**	31.9	16.7	12.4	20.7	30.0	28.7	12.7	28.5	24.1
Slide Left-Right	Right2Right	23.6±2.2	23.8±1.5	21.8±3.5	13.8±1.3	13.6±1.4	25.5±0.8	22.2±2.4	26.7±0.8	2.4±0.9	8.6±1.2	9.2±1.9
	Right2Left	12.0±2.2	10.1±1.8	11.0±3.8	2.0±0.3	2.9±1.2	10.1±1.2	10.5±2.2	10.8±0.8	0.1±0.2	3.0±1.1	2.5±1.2
	Left2Right	17.8±0.6	14.4±1.4	12.2±2.1	6.5±1.5	8.6±1.6	14.9±2.8	13.1±2.0	19.3±1.1	0.2±0.2	0.6±0.2	1.1±0.8
	Left2Left	13.3±1.4	13.3±2.2	10.9±2.9	5.5±1.0	7.8±2.6	9.1±2.5	10.7±1.1	12.8±0.8	1.2±0.5	2.4±0.8	3.3±1.0
	Average	**16.7**	15.4	14.0	7.0	8.2	14.9	14.1	**17.4**	1.0	3.7	4.0
Slide Near-Far	Near	21.3±1.2	22.2±1.7	21.5±1.6	5.2±1.3	5.4±1.6	22.4±1.3	17.9±2.4	21.6±1.1	5.5±2.0	14.8±2.3	9.4±2.5
	Far	5.0±0.9	5.4±0.8	3.5±1.2	0.0±0.0	0.0±0.0	4.4±0.9	1.6±0.5	1.0±0.1	1.3±0.9	2.5±1.4	0.7±0.4
	Average	13.2	**13.8**	12.5	2.6	2.7	**13.4**	9.8	11.3	3.4	8.6	5.1
HandReach Near-Far	Near	33.6±2.0	32.9±1.0	32.7±1.6	0.1±0.0	0.1±0.1	36.0±0.8	32.4±1.8	30.4±4.0	0.0±0.0	0.8±1.6	0.2±0.1
	Far	15.0±1.8	17.0±1.7	14.2±1.4	0.0±0.1	0.0±0.0	16.7±1.3	12.1±2.5	11.8±1.5	0.0±0.0	0.0±0.0	0.0±0.0
	Average	24.3	**25.0**	23.5	0.0	0.0	**26.3**	22.2	21.1	0.0	0.4	0.1
Average	IID Tasks	**36.2**	**35.9**	35.4	21.0	20.5	35.1	34.3	35.1	19.3	28.5	28.5
	OOD Tasks	**25.0**	**24.1**	22.5	8.6	6.4	14.4	19.2	20.8	9.4	17.7	16.1