GOPlan: Goal-conditioned Offline Reinforcement Learning by Planning with Learned Models

Anonymous authors
Paper under double-blind review

Abstract

Offline Goal-Conditioned RL (GCRL) offers a feasible paradigm for learning general-purpose policies from diverse and multi-task offline datasets. Despite notable recent progress, the predominant offline GCRL methods, mainly model-free, face constraints in handling limited data and generalizing to unseen goals. In this work, we propose *Goal-conditioned Offline Planning* (GOPlan), a novel model-based framework that contains two key phases: (1) pretraining a prior policy capable of capturing multi-modal action distribution within the multi-goal dataset; (2) employing the *reanalysis* method with planning to generate imagined trajectories for funetuning policies. Specifically, we base the prior policy on an advantage-weighted conditioned generative adversarial network, which facilitates distinct mode separation, mitigating the pitfalls of out-of-distribution (OOD) actions. For further policy optimization, the *reanalysis* method generates high-quality imaginary data by planning with learned models for both intra-trajectory and inter-trajectory goals. With thorough experimental evaluations, we demonstrate that GOPlan achieves state-of-the-art performance on various offline multi-goal navigation and manipulation tasks. Moreover, our results highlight the superior ability of GOPlan to handle small data budgets and generalize to OOD goals.

1 Introduction

Offline reinforcement learning (RL) (Fujimoto et al., 2019; Kumar et al., 2020; Levine et al., 2020; Janner et al., 2021) enables learning policies from offline dataset without online interactions with the environment, offering an efficient and safe approach for real-world applications, e.g., robotics, autonomous driving, and healthcare (Levine et al., 2020). Developing a general-purpose policy capable of multiple skills is particularly appealing for RL. Offline goal-conditioned RL (GCRL) (Chebotar et al., 2021; Yang et al., 2022b) offers such a way to learn multiple skills from diverse and multi-goal datasets. So far, prior works in this direction (Chebotar et al., 2021; Yang et al., 2022b; Ma et al., 2022; Mezghani et al., 2022) mainly focus on avoiding out-of-distribution (OOD) actions and learning to reach all in-dataset goals. Despite recent progress, efficiently mastering diverse skills with limited data and achieving OOD goal generalization remain significant challenging.

Model-based RL (Schrittwieser et al., 2020; Hafner et al., 2020; Moerland et al., 2023) is a natural choice to overcome these two challenges for offline GCRL, as it enables efficient learning from limited data (Deisenroth & Rasmussen, 2011; Argenson & Dulac-Arnold, 2021; Schrittwieser et al., 2021) and enjoys better generalization (van Steenkiste et al., 2019; Yu et al., 2020; Lee et al., 2020). The advantages of model-based RL in addressing the aforementioned challenges emanate from its ability to utilize more direct supervision information rather than relying solely on scalar rewards. Additionally, world models used in model-based RL are trained using supervised learning, which is more stable and reliable than bootstrapping (Yu et al., 2020). A recent notable model-based method, *reanalyse* (Schrittwieser et al., 2020; 2021), employs model-based value estimation and planning to generate improved value and policy target for given states, showing advantageous performance for both online and offline RL and holding the potential for handling the limited data and OOD generalization challenges. However, it should be noted that reanalysis is primarily designed for single-task RL and cannot be directly applied for offline GCRL. This is in part due to the challenges presented by multi-goal datasets, which contain trajectories collected for heterogeneous goals. For effective

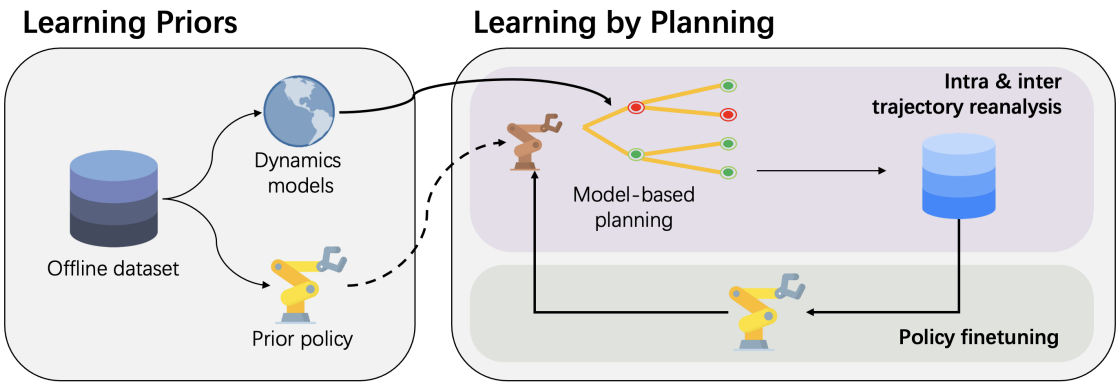

Figure 1: The two-stage framework of GOPlan: pretraining a prior policy and a group of dynamics models, and finetuning policy with imagined trajectories generated by the reanalysis method.

long-term planning in offline GCRL, accurately modeling action modes in these datasets while avoiding OOD actions is crucial. In addition, the question of what goal to plan for, must be addressed in reanalysis for offline GCRL, which is not a concern for single-task RL.

To this end, we introduce *Goal-conditioned Offline Planning* (GOPlan), a novel model-based algorithm designed for offline GCRL. A diagram of GOPlan is shown in Figure 1. GOPlan consists of two main stages, a pretraining stage and a reanalysis stage. During the pretraining stage, GOPlan trains a policy using advantage-weighted Conditioned Generative Adversarial Network (CGAN) to capture the multi-modal action distribution in the heterogeneous multi-goals dataset. The pretrained policy exhibits notable mode separation to avoid OOD actions and is improved towards high-reward actions, making it suitable for offline planning. A group of dynamics models is also learned during this stage and will be used for planning and uncertainty quantification. During the reanalysis stage, GOPlan finetunes the policy with imagined trajectories for further policy optimization. Specifically, GOPlan generates a reanalysis buffer by planning using the policy and learned models for both intra-trajectory and inter-trajectory goals. Given a state in a trajectory, intra-trajectory goals are along the same trajectory as the state, and inter-trajectory goals lie in different trajectories. We quantify the uncertainty of the planned trajectories based on the disagreement of the models (Pathak et al., 2019; Yu et al., 2020) to avoid going excessively outside the dynamics model's support. The imagined data with small uncertainty can be high-quality demonstrations that can enhance the agent's ability to achieve both in-dataset and out-of-dataset goals. By iteratively planning to generate better data and finetune the policy with advantage-weighted CGAN, the reanalysis stage significantly improves the policy's performance while also reducing the requirement for a large offline dataset. Furthermore, GOPlan can leverage the generalization of dynamics models to plan for OOD goals, which is a particularly promising feature for offline GCRL.

In the experiments, we evaluate GOPlan across multiple multi-goal navigation and manipulation tasks, and demonstrate that GOPlan outperforms prior state-of-the-art (SOTA) offline GCRL algorithms. Moreover, we extend the evaluation to the small data budget and the OOD goal settings, in which GOPlan exhibits superior performance compared to other algorithms by a substantial margin. To summarize, our main contributions are three folds:

1. We propose GOPlan, a novel model-based offline GCRL algorithm that can address challenging settings with limited data and OOD goals.

2. GOPlan consists of two stages, pretraining a prior policy via advantage-weighted CGAN and finetuning the policy with high-quality imagined trajectories generated by planning.

3. Experiments validate GOPlan's efficacy in benchmarks as well as two challenging settings.

2 Related Work

Offline RL. Addressing OOD actions for offline RL is essential to ensure safe policies during deployment. Several works propose to avoid OOD actions by enforcing constraints on the policy (Wang et al., 2018; Wu et al., 2019; Nair et al., 2021) or penalizing the Q values of OOD state-action pairs (Kumar et al., 2020; An et al., 2021; Bai et al., 2022; Yang et al., 2022a; Sun et al., 2022). However, these methods generally use a uni-modal Gaussian policy, which cannot capture the multi-modal action distribution of the heterogeneous offline dataset. To address the multi-modality issue, PLAS (Zhou et al., 2020) decodes variables sampled from a variational auto-encoder (VAE) latent space, while LAPO (Chen et al., 2022) leverages advantage-weighted latent space to further optimize the policy. A recent work (Yang et al., 2022c) demonstrates GAN's ability to capture multiple action modes. Based on GAN, DASCO (Vuong et al., 2022) proposes a dual-generator algorithm to maximize the expected return. Distinctively, we propose to capture the multi-modal distribution in multi-goal dataset via advantage-weighted CGAN.

Offline GCRL. In offline GCRL, agents learn goal-conditioned policies from a static dataset. One direction for offline GCRL is through goal-conditioned supervised learning (GCSL), which directly performs imitation learning on the relabelled data (Ghosh et al., 2021; Emmons et al., 2022). When the data is suboptimal or noisy, weighted GCSL (Yang et al., 2022b; Wang et al., 2024; Yang et al., 2023) with multiple weighting criteria is a more powerful solution. Other directions include optimizing the state occupancy on the targeted goal distribution (Ma et al., 2022) and contrastive RL (Eysenbach et al., 2022). Additionally, Mezghani et al. (2022) design a self-supervised dense reward learning method to solve offline GCRL with long-term planning. Different from prior works, our work alleviates the multi-modality problem raised from the multi-goal dataset via advantage-weighted CGAN, and efficiently utilizes dynamics models for reanalysis and policy improvement.

Model-based RL. The ability to reach any in-dataset goals requires an agent to grasp the invariance underlying different tasks. The invariance can be represented by the dynamics of the environment that can facilitate RL (Schrittwieser et al., 2020; Hafner et al., 2020; Nagabandi et al., 2020; Hansen et al., 2022; Rigter et al., 2022) and GCRL (Charlesworth & Montana, 2020; Yang et al., 2021). In the offline setting, an ensemble of dynamics models is used to construct a pessimistic MDP (Kidambi et al., 2020; Yu et al., 2020); Reanalysis (Schrittwieser et al., 2021) uses Monte-Carlo tree search (Coulom, 2007) with learned model to generate new training targets for a given state. These dynamics models could be deterministic models (Yang et al., 2021), Gaussian models (Nagabandi et al., 2020), or recurrent models with latent space (Hafner et al., 2019; 2020). GOPlan chooses the deterministic models due to its simplicity, and with the recurrent models it can be extended to high-dimensional state space. Imagined trajectories during offline planning may contain high uncertainty, resulting in inferior performance. MOPP (Zhan et al., 2022) measures the uncertainty by the disagreement of the dynamics models and prunes uncertain trajectories. However, prior model-based offline RL works are not designed for the multi-goal setting. Instead, we perform model-based planning for both intra-trajectory and inter-trajectory goals, iteratively finetuning the GAN-based policy with high-quality and low-uncertainty imaginary data.

3 Preliminaries

Goal-conditioned RL. Goal-conditioned RL is generally formulated as Goal-Conditioned Markov Decision Processes (GCMDP), denoted by a tuple $< \mathcal{S}, \mathcal{A}, \mathcal{G}, P, \gamma, r >$ where $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{G}$ are the state, action and goal spaces, respectively. $P(s_{t+1} \mid s_t, a_t)$ is the transition dynamics, r is the reward function, and γ is the discount factor. An agent learns a goal-conditioned policy $\pi : \mathcal{S} \times \mathcal{G} \to \mathcal{A}$ to maximize the expected discounted cumulative return:

$$J(\pi) = \mathbb{E}_{\substack{g \sim P_g, s_0 \sim P_0(s_0), \\ a_t \sim \pi(\cdot|s_t, g), \\ s_{t+1} \sim P(\cdot|s_t, a_t)}} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t, g) \right], \tag{1}$$

where P_g and $P_0(s_0)$ are the distribution of the goals and initial states, and T corresponds to the length of an episode. The expected value of a state-goal pair is defined as: $V^\pi(s,g) = \mathbb{E}_\pi\left[\sum_{t=0}^{T}\gamma^t r_t \mid s_0 = s\right]$. A sparse reward is non-zero only when the goal is reached, i.e., $r(s,a,g) = \mathbb{1}[\|\phi(s) - g\|_2^2 \leq \epsilon]$, where $\phi : \mathcal{S} \to \mathcal{G}$ is the state-to-goal mapping (Andrychowicz et al., 2017) and ϵ is a threshold. This sparse reward function is accessible to the agent during the learning process. For offline GCRL, the agent can only learn from an offline dataset $\mathcal{B}$ without interacting with the environment.

Reanalysis. MuZero Reanalyse (Schrittwieser et al., 2020; 2021) introduces reanalysis to perform iterative model-based value and policy improvement for states in the dataset, resulting in an ongoing cycle of refinement through updated predictions and improved generated training data. Specifically, MuZero Reanalyse updates its parameters θ for K-step model rollouts via supervised learning:

$$l_t(\theta) = \sum_{k=0}^{K} l^p(\pi_{t+k}, p_t^k) + \sum_{k=0}^{K} l^v(z_{t+k}, v_t^k) + \sum_{k=0}^{K} l^r(u_{t+k}, r_t^k), \tag{2}$$

where p_t^k, v_t^k, r_t^k are action, value, and reward predictions generated by the model θ. Their training targets π_{t+k}, z_{t+k} are generated via search tree, and u_{t+k} is the true reward. In continuous control settings, the loss functions $l^p(\cdot,\cdot)$, $l^v(\cdot,\cdot)$, $l^r(\cdot,\cdot)$ are mean square error (MSE) between the inputs.

4 Methodology

This section introduces the two-stage GOPlan algorithm, specially designed for offline GCRL. During the pretraining stage, we implement an advantage-weighted CGAN to establish a proficient prior policy for capturing multi-modal action distribution in offline datasets, which is suitable for subsequent model-based planning. In the reanalysis stage, we enhance the performance of the agent by enabling planning with learned models and multiple goals, resulting in a significant policy improvement. The overall framework is depicted in Figure 1.

4.1 Learning Priors from Offline Data

Learning Prior Policy. The initial step involves learning a policy that can generate in-distribution and high-reward actions from multi-goal offline data. Due to the nature of collecting data for multiple heterogeneous goals, these datasets can be highly multi-modal (Lynch et al., 2020; Yang et al., 2022b), meaning that a state can have multiple valid action labels. The potential conflict between these actions poses a significant learning challenge. Unlike prior works using Gaussian (Yang et al., 2022b; Ma et al., 2022), conditional Variational Auto-encoder (CVAE) (Zhou et al., 2020; Chen et al., 2022) and Conditioned Generative Adversarial Network (CGAN) (Yang et al., 2022c), we employ Weighted CGAN as the prior policy. In Figure 2, we compare six different models : Gaussian, Weighted Gaussian, CVAE, Weighted CVAE, CGAN, and Weighted CGAN on a multi-modal dataset with imbalanced rewards, where high rewards have less frequency, as illustrated in Figure 2(a-2). The weighting scheme for Weight CVAE and Weighted CGAN is based on advantage re-weighting (Yang et al., 2022b) and we directly use rewards as weights in this example.

In Figure 2, Weighted CGAN outperforms other models by exhibiting a more distinct mode separation. As a result, the policy generates fewer OOD actions by reducing the number of interpolations between modes. In contrast, other models all suffer from interpolating between modes. Even though VAE models perform better than Gaussian models, they are still prone to interpolation due to the regularization of the Euclidean norm on the Jacobian of the VAE decoder (Salmona et al., 2022). Furthermore, without employing advantage-weighting, both the CVAE and CGAN models mainly capture the denser regions of the action distribution, but fail to consider their importance relative to the rewards associated with each mode.

Based on the empirical results, the utilization of the advantage-weighted CGAN model for modeling the prior policy from multi-modal offline data demonstrates notable advantages for offline GCRL. In this framework, the discriminator is responsible for distinguishing high-quality actions in the offline dataset from those generated by the policy, while the generative policy is designed to generate actions that outsmart the discriminator

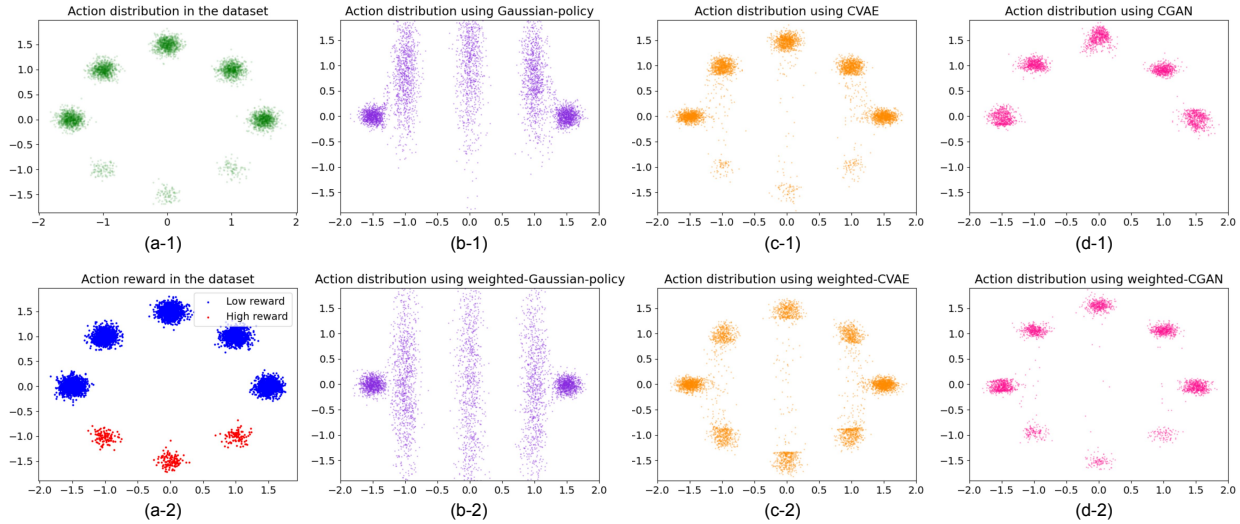

Figure 2: An example about modeling the multi-modal behavior policy while maximizing average rewards. The x-axis represents the state, and the y-axis represents the multi-modal action. (a-1) shows the action distribution of the offline dataset. (a-2) shows the corresponding reward distribution. (b-1) and (b-2) illustrate the action distributions generated by Gaussian and Weighted Gaussian. (c-1) and (c-2) illustrate action distributions from CVAE and Weighted CVAE. (d-1) and (d-2) illustrate action distributions from CGAN and Weighted CGAN.

in an adversarial process. This mechanism encourages the policy to produce actions that closely resemble high-quality actions from the offline dataset. In Section 4.3, we will elaborate on the specific definition of the advantage-weighted CGAN model.

Learning Dynamics Models. We also train a group of N dynamic models $\{M_{\psi_i}\}_{i=1}^{N}$ from the offline data to predict the residual between current states and next states. Each model M_{ψ_i} minimizes the following loss:

$$\mathcal{L}(\psi_i) = \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{B}}\left[||M_{\psi_i}(s_t,a_t) - (s_{t+1} - s_t)||_2^2\right]. \tag{3}$$

Then the predicted next state is $\overline{s}_{t+1} = s_t + M_{\psi_i}(s_t, a_t)$. As the dynamics models are trained through supervised learning, making them more stable and better equipped with generalization abilities compared to bootstrapping (Yu et al., 2020). In the subsequent section, we will illustrate how the learned models can be utilized to enhance the training efficiency and generalization ability of offline GCRL.

4.2 Learning by Planning with Learned Models

Based on the learned prior policy and dynamics models, we reanalyse and finetune the policy in this learning procedure. As the dynamics information encapsulated in the offline dataset remains invariant across different goals, we utilize this property to equip our policy with the capacity to achieve a broad spectrum of goals by reanalysing the current policy for both intra-trajectory and inter-trajectory goals. The reanalysis procedure is shown in Figure 3, where we apply model-based planning method to generate trajectories for selected goals and save potential trajectories that can help improve the policy and have small uncertainty. By integrating this imagined data to update the policy, we repeat the aforementioned process to iteratively improve the policy.

Model-based Planning. We first introduce our planning method that serves intra-trajectory and inter-trajectory reanalysis. Our planner is similar to prior model-predictive approaches (Nagabandi et al., 2020; Argenson & Dulac-Arnold, 2021; Charlesworth & Montana, 2020; Schrittwieser et al., 2021), where the planner employs a model to simulate multiple imaginary trajectories, assigns scores to the initial actions,

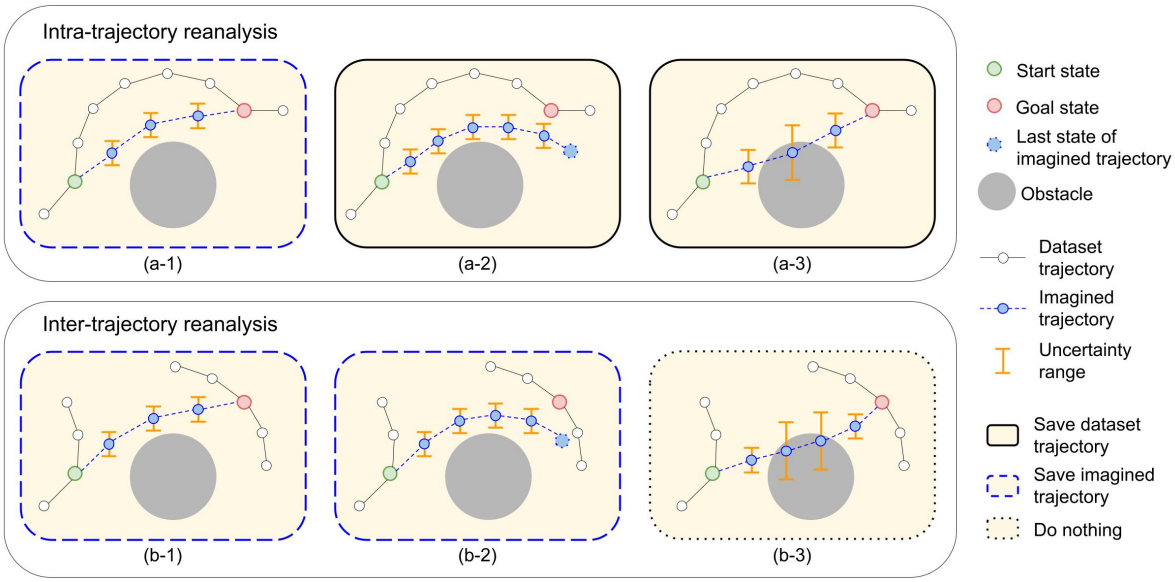

Figure 3: Illustration of intra-trajectory and inter-trajectory reanalysis. There are six scenarios: (a-1) the imagined trajectory is valid and better than the original trajectory; (a-2) the imagined trajectory fails to reach the goal within the same number of steps as the original trajectory; (b-1) a valid imagined trajectory connects the state to an inter-trajectory goal; (b-2) a valid imagined trajectory that does not achieve the desired goal; (a-3) (b-3) invalid imagined trajectories with large uncertainty.

and generates actions based on these scores. Following prior work (Charlesworth & Montana, 2020), given the current state s_t and the selected goal g, our planner firstly samples C initial actions $\{a_t^c\}_{c=1}^C$ from the prior generative policy π and predicts C next states $\{\overline{s}_{t+1}^c\}_{c=1}^C$ based on a randomly selected dynamics model M_{ψ_i}. To score each initial action, the planner duplicates every next state H times and generates H imagined trajectories of K steps with the policy and a randomly chosen dynamics model at each step. For each initial action a_t^c, the planner averages and normalizes all cumulative returns commencing from that initial action as $R(a_t^c)$, where the reward for every generated step is computed by the sparse reward function. Finally, the planner's output action for s_t, g is $a_t = \frac{\sum_{c=1}^C e^{\kappa R(a_t^c)} a_t^c}{\sum_{c=1}^C e^{\kappa}}$, where κ is a hyper-parameter tuning the exponentially weights for actions. It is worth noting that advantage-weighted CGAN effectively provides diverse in-dataset actions with higher rewards, enabling efficient and safe action planning for offline GCRL. The planning algorithm is outlined in Algorithm 2 provided in Appendix A.

Intra-trajectory and Inter-trajectory Reanalysis. Offline goal-conditioned planning involves the challenges of choosing planning goals and selecting appropriate imaginary data for policy training. In this paper, we present a novel approach that provides solutions to these challenges by incorporating intra-trajectory and inter-trajectory reanalysis, as illustrated in Figure 3. Specifically, our approach includes designed mechanisms for assigning goals and selecting generated trajectory to be stored in a reanalysis buffer $\mathcal{B}_{re}$ for subsequent policy improvement.

Since the primary objective of offline GCRL is to reach all in-dataset goals, utilizing all the states in the dataset as goals appears to be a natural choice. These goals can be classified into two categories: ***intra-trajectory goals***, which are along the same trajectory as the starting state, and ***inter-trajectory goals***, which lie in different trajectories. The division criterion is determined by the existence of a path between the state and the goal. Intra-trajectory goals refer to those that already have a path that links the state to them and hence, they can be reached with high probability because the successful demonstration has been provided. In contrast, inter-trajectory goals may not have a valid path to be reached from the given state. Therefore, different criterion is needed for the two situations.

The main idea of the two types of reanalysis is to store improved trajectories than the dataset and remove invalid trajectories with large uncertainty. The criterion of removing unrealistic imagined trajectories with large uncertainty is shared by both intra-trajectory and inter-trajectory reanalysis, because imitating these trajectories can lead the policy excessively outside the support of the dynamics model. To achieve this, we measure the the uncertainty of a trajectory by accessing the disagreement of a group of learned dynamics models. Specifically, given a trajectory $\tau = \{s_0, a_0, s_1, a_1, ..., s_H\}$, the uncertainty is defined as the maximal disagreement on the transition within the trajectory:

$$U(\tau) = \max_{0 \leq k \leq H} \frac{1}{N} \sum_{i=1}^{N} ||M_{\psi_i}(s_k, a_k) - \bar{s}_{k+1}||_2^2, \tag{4}$$

where M_{ψ_i} is the i-th model and $\bar{s}_{k+1}$ is the mean prediction over N models: $\bar{s}_{k+1} = \frac{1}{N}\sum_{i=1}^{N} M_{\psi_i}(s_k, a_k)$.

Intra-trajectory reanalysis. randomly samples a trajectory τ and two states $s_t, s_{t+k} \in \tau$, where $k > 0$. After that, the model-based planning approach mentioned earlier is then employed to generate an action at each step from s_t with the aim of reaching $\phi(s_{t+k})$, transiting according to the mean prediction of $M_{\psi_i}, i \in [1, N]$. There are mainly three cases: (a-1) if the planner reaches $\phi(s_{t+k})$ with fewer than k steps and the uncertainty of the imagined trajectory is less than a threshold value u, the generated virtual trajectory $\overline{\tau}$ is stored into a reanalysis buffer $\mathcal{B}_{re}$. In cases where the (a-2) policy fails to reach $\phi(s_{t+k})$ within k steps or (a-3) the generated virtual trajectory has larger uncertainty than u, the original trajectory $\tau_{t:t+k}$ is included in $\mathcal{B}_{re}$ because we can enhance learning the original successful demonstrations during fine-tuning.

Inter-trajectory reanalysis. involves randomly selecting a state s_t in a trajectory τ_1 and another state s_g in a different trajectory τ_2, followed by model-based planning from s_t to reach $\phi(s_g)$. There are also three cases: (b-1) if the planner successfully reaches $\phi(s_g)$ with an uncertainty less than u, we include the generated virtual trajectory $\overline{\tau}$ into the reanalysis buffer $\mathcal{B}_{re}$. (b-2) Moreover, in situations where the policy fails to attain the intended goal state $\phi(s_g)$ but the associated trajectory exhibits low levels of uncertainty, the virtually achieved goals along the generate trajectory are labeled as the intended goal for the trajectory. This approach enables the utilization of failed virtual trajectories to enhance the diversity of the reanalysis buffer $\mathcal{B}_{re}$. (b-3) Additionally, if the imagined trajectory has high uncertainty, we do not save such invalid trajectory.

4.3 Policy Optimization

In GOPlan, the policy is updated during both the pretraining stage and the finetuning stage. We train the policy π via advantage-weighted CGAN, which can be formulated as the following objective:

$$\begin{aligned} \max_{D} \min_{\pi} & \mathbb{E}_{(s_t,a_t,g)\sim\mathcal{B}} \left[w(s_t, a_t, g) \log D(s_t, a_t, g) \right] \\ & + \mathbb{E}_{(s_t,g)\sim\mathcal{B}, a'\sim\pi} \left[\log(1 - D(s_t, a', g)) \right] \end{aligned} \tag{5}$$

Here, D is the discriminator weighted by w. $w(s, a, g) = \exp(A(s, a, g) + N(s, g))$ is the exponential advantage weight, where $N(s, g)$ serves as a normalizing factor to ensure that $\sum_{a\in\mathcal{A}} w(s, a, g)\pi_b(a \mid s, g) = 1$. π_b is the behavior policy underlying the relabeled offline dataset $\mathcal{B}$. The advantage function can be estimated by a learned value function V_{θ_v}: $A(s_t, a_t, g) = r(s_t, a_t, g) + \gamma V_{\theta_v}(s_{t+1}, g) - V_{\theta_v}(s_t, g)$. The value function $V_{\theta_v}(s_t, g)$ is updated to minimize the TD loss:

$$\mathcal{L}(\theta_v) = \mathbb{E}_{(s_t,r_t,s_{t+1},g)\sim\mathcal{B}} \left[(V_{\theta_v}(s_t, g) - y_t)^2 \right], \tag{6}$$

where the target $y_t = r_t + \gamma V_{\theta_v^-}(s_{t+1}, g)$ and the target network $V_{\theta_v^-}$ is slowly updated to improve training stability.

To optimize the minmax objective in Eq.equation 5, we train a discriminator D and a generative policy π separately. The discriminator D, with parameter θ_d, learns to minimize the following loss function:

$$\begin{aligned} \mathcal{L}(\theta_d) = & - \mathbb{E}_{(s_t,a_t,g)\sim\mathcal{B}} \left[w(s_t, a_t, g) \log D_{\theta_d}(s_t, a_t, g) \right] \\ & - \mathbb{E}_{\substack{(s_t,g)\sim\mathcal{B}, z\sim P(z), \\ a'\sim\pi_{\theta_\pi}(s_t,g,z)}} \left[\log(1 - D_{\theta_d}(s_t, a', g)) \right], \end{aligned} \tag{7}$$

Algorithm 1 Goal-conditioned Offline Planning (GOPlan).

Initialise: N dynamics models $\{\psi_i\}_{i=1}^N$, a discriminator θ_d, a policy θ_π, a goal-conditioned value function θ_v; an offline dataset $\mathcal{B}$ and a reanalysis buffer $\mathcal{B}_{re}$; the state-to-goal mapping ϕ.

# Pre-train
while *not converges* **do**
  Update $\{\psi_i\}_{i=1}^N$ using $\mathcal{B}$.
  Update θ_v using $\mathcal{B}$. ▷ Eq. 6
  Update θ_d using $\mathcal{B}$. ▷ Eq. 7
  Update θ_π using $\mathcal{B}$. ▷ Eq. 8
end while

# Finetune
for $i = 1, ..., I$ **do**
  for $j = 1, ..., I_{intra}$ **do**
    $\tau =$ Intra_traj()
    $\mathcal{B}_{re} = \mathcal{B}_{re} \cup \tau$
  end for
  for $j = 1, ..., I_{inter}$ **do**
    $\tau =$ Inter_traj()
    $\mathcal{B}_{re} = \mathcal{B}_{re} \cup \tau$
  end for
  Finetune θ_v using $\mathcal{B}_{re}$. ▷ Eq. 6
  Finetune θ_d using $\mathcal{B}_{re}$. ▷ Eq. 7
  Finetune θ_π using $\mathcal{B}_{re}$. ▷ Eq. 8
end for

def Intra_traj():
$(s_t, s_{t+1}, ..., s_{t+K}, g) \sim \mathcal{B}$, $\hat{s}_t = s_t$
for $k = 0, ..., K$ **do**
  $a_{t+k} = \texttt{Plan}(\hat{s}_{t+k}, \phi(s_{t+K}))$
  $\hat{s}_{t+k+1} = M_{\psi_{i,i\sim\{1,...,N\}}}(\hat{s}_{t+k}, a_{t+k})$
  if $U(\hat{s}_{t+k}, a_{t+k}) > u$ **then**
    return $\{s_t, s_{t+1}, ..., s_{t+K}, g\}$.
  end if
  if $\hat{s}_{t+k+1}$ *achieves* $\phi(s_{t+K})$ **then**
    return $\{s_t, \hat{s}_{t+1}, ..., \hat{s}_{t+k+1}, \phi(s_{t+K})\}$.
  end if
end for
return $\{s_t, s_{t+1}, ..., s_{t+K}, g\}$.

def Inter_traj():
$s_0 \sim \mathcal{B}$, $s_g \sim \mathcal{B}$, $\hat{s}_0 = s_0$
for $t = 0, ..., T$ **do**
  $a_t = \texttt{Plan}(\hat{s}_t, \phi(s_g))$
  $\hat{s}_{t+1} = M_{\psi_{i,i\sim\{1,...,N\}}}(\hat{s}_t, a_t)$
  if $U(\hat{s}_t, a_t) > u$ **then**
    return $\{\emptyset\}$
  end if
  if $\hat{s}_{t+1}$ *achieves* $\phi(s_g)$ **then**
    return $\{s_0, \hat{s}_1, ..., \hat{s}_{t+1}, \phi(s_g)\}$
  end if
end for
return $\{s_0, \hat{s}_1, ..., \hat{s}_T, \phi(\hat{s}_T)\}$

where $P(z)$ constitutes random noise that follows a diagonal Gaussian distribution. The discriminator's output is passed through a sigmoid function, thereby constraining the output to lie within $(0, 1)$. To deceive the discriminator, the policy network π, with parameter θ_π, is trained to minimize the following loss function:

$$\mathcal{L}(\theta_\pi) = \mathbb{E}_{\substack{(s_t,a_t,g)\sim\mathcal{B}, z\sim P(z), \\ a'\sim\pi_{\theta_\pi}(s_t,g,z)}} \left[\log(1 - D_{\theta_d}(s_t, a', g))\right]. \tag{8}$$

Through this training process, the policy network is capable of producing actions that closely resemble high-quality actions in the dataset and can enjoy policy improvement similar to prior advantage-weighted works (Wang et al., 2018; Peng et al., 2019).

4.4 Overall Algorithm

In the pretraining stage, we train the dynamics models $M_{\psi_i}, i \in [1, N]$, the discriminator D_{θ_d}, the value function V_{θ_v} and the policy π_{θ_π} until convergence. In the subsequent reanalysis stage, we generate new trajectories through intra-trajectory and inter-trajectory reanalysis, save them into the reanalysis buffer, and then employ the reanalysis buffer to finetune the value function, the discriminator, and the policy. The process of reanalysis and fine-tuning is repeated over I iterations to improve the policy performance. Algorithm 1 outlines the details of the two main stages of the overall algorithm.

5 Experiments

We conduct a comprehensive evaluation of GOPlan across a collection of continuous control tasks (Plappert et al., 2018; Yang et al., 2022b) with multiple goals and sparse rewards. In addition, we also assess the efficacy of GOPlan in settings with limited data budgets and OOD goal generalization.

Table 1: Average return with standard deviation on the offline goal-conditioned benchmark and the small dataset setting.

Task	GOPlan	BC	GCSL	WGCSL	GEAW	AM	CRL	g-TD3-BC	g-TT	g-DT
PointReach	**46.09**$_{\pm 0.1}$	39.36$_{\pm 0.4}$	39.27$_{\pm 0.4}$	44.40$_{\pm 0.1}$	42.95$_{\pm 0.1}$	43.56$_{\pm 0.6}$	44.53$_{\pm 0.2}$	43.69$_{\pm 0.2}$	45.39$_{\pm 0.5}$	45.85$_{\pm 0.5}$
PointRooms	**43.57**$_{\pm 1.8}$	33.17$_{\pm 0.5}$	33.05$_{\pm 0.5}$	36.15$_{\pm 0.8}$	36.02$_{\pm 0.5}$	33.45$_{\pm 1.9}$	40.12$_{\pm 1.0}$	42.32$_{\pm 0.8}$	42.46$_{\pm 0.8}$	43.20$_{\pm 1.2}$
Reacher	**40.67**$_{\pm 0.3}$	35.72$_{\pm 0.3}$	36.42$_{\pm 0.3}$	40.57$_{\pm 0.2}$	38.89$_{\pm 0.1}$	37.48$_{\pm 4.1}$	37.79$_{\pm 0.2}$	37.39$_{\pm 0.3}$	38.79$_{\pm 0.3}$	38.65$_{\pm 0.4}$
SawyerReach	40.43$_{\pm 0.3}$	32.91$_{\pm 0.3}$	33.65$_{\pm 0.3}$	40.12$_{\pm 0.2}$	37.42$_{\pm 0.3}$	40.91$_{\pm 0.2}$	36.73$_{\pm 0.3}$	30.96$_{\pm 1.7}$	40.25$_{\pm 0.3}$	**41.10**$_{\pm 0.3}$
SawyerDoor	**44.42**$_{\pm 0.3}$	35.03$_{\pm 0.2}$	35.67$_{\pm 0.1}$	42.81$_{\pm 0.2}$	40.03$_{\pm 0.1}$	42.49$_{\pm 0.5}$	38.58$_{\pm 0.4}$	35.90$_{\pm 0.5}$	44.32$_{\pm 0.4}$	44.23$_{\pm 0.3}$
FetchReach	**47.33**$_{\pm 0.2}$	42.03$_{\pm 0.2}$	41.72$_{\pm 0.3}$	46.33$_{\pm 0.0}$	45.01$_{\pm 0.1}$	46.50$_{\pm 0.1}$	46.10$_{\pm 0.1}$	45.51$_{\pm 0.3}$	47.15$_{\pm 0.1}$	47.03$_{\pm 0.2}$
FetchPush	**39.15**$_{\pm 0.6}$	31.56$_{\pm 0.6}$	28.56$_{\pm 0.9}$	39.11$_{\pm 0.1}$	37.42$_{\pm 0.2}$	30.49$_{\pm 2.1}$	36.52$_{\pm 0.6}$	30.83$_{\pm 0.6}$	38.90$_{\pm 0.8}$	38.63$_{\pm 0.7}$
FetchPick	**37.01**$_{\pm 1.1}$	31.75$_{\pm 1.2}$	25.22$_{\pm 0.8}$	34.37$_{\pm 0.5}$	34.56$_{\pm 0.5}$	34.07$_{\pm 0.6}$	35.77$_{\pm 0.2}$	36.51$_{\pm 0.5}$	36.65$_{\pm 0.8}$	35.24$_{\pm 1.0}$
FetchSlide	10.08$_{\pm 0.8}$	0.84$_{\pm 0.3}$	3.05$_{\pm 0.6}$	**10.73**$_{\pm 1.0}$	4.55$_{\pm 1.7}$	6.92$_{\pm 1.2}$	9.91$_{\pm 0.2}$	5.88$_{\pm 0.6}$	9.61$_{\pm 1.0}$	8.20$_{\pm 1.7}$
HandReach	**28.28**$_{\pm 5.3}$	0.06$_{\pm 0.1}$	0.57$_{\pm 0.6}$	26.73$_{\pm 1.2}$	0.81$_{\pm 1.5}$	0.02$_{\pm 0.0}$	6.46$_{\pm 2.0}$	5.21$_{\pm 1.6}$	24.12$_{\pm 1.4}$	22.80$_{\pm 1.5}$
Average	**37.70**	28.24	27.71	36.13	31.76	31.58	33.25	31.42	36.76	36.49
FetchPush-s	**37.31**$_{\pm 0.5}$	25.54$_{\pm 1.0}$	26.30$_{\pm 0.7}$	32.35$_{\pm 0.9}$	33.68$_{\pm 1.9}$	32.93$_{\pm 0.6}$	31.72$_{\pm 1.5}$	30.92$_{\pm 0.6}$	37.02$_{\pm 1.1}$	37.10$_{\pm 0.9}$
FetchPick-s	**32.85**$_{\pm 0.3}$	23.05$_{\pm 1.0}$	23.71$_{\pm 1.4}$	29.12$_{\pm 0.2}$	30.92$_{\pm 0.5}$	25.56$_{\pm 3.5}$	32.27$_{\pm 0.8}$	29.06$_{\pm 4.0}$	32.53$_{\pm 1.5}$	31.23$_{\pm 1.2}$
FetchSlide-s	**5.04**$_{\pm 0.4}$	0.31$_{\pm 0.1}$	0.98$_{\pm 0.4}$	0.22$_{\pm 0.1}$	0.30$_{\pm 0.1}$	1.97$_{\pm 2.7}$	4.74$_{\pm 0.6}$	0.16$_{\pm 0.2}$	2.39$_{\pm 1.1}$	1.60$_{\pm 1.4}$
HandReach-s	**10.11**$_{\pm 1.4}$	0.16$_{\pm 0.1}$	0.13$_{\pm 0.1}$	0.12$_{\pm 0.1}$	0.03$_{\pm 0.0}$	0.08$_{\pm 0.1}$	0.45$_{\pm 0.3}$	1.6$_{\pm 2.3}$	4.22$_{\pm 0.6}$	5.69$_{\pm 1.2}$
FetchPush-es	**18.91**$_{\pm 1.2}$	10.57$_{\pm 1.9}$	5.96$_{\pm 2.1}$	13.25$_{\pm 2.8}$	10.35$_{\pm 2.3}$	11.28$_{1.6}$	16.13$_{\pm 1.8}$	6.27$_{\pm 0.9}$	15.76$_{\pm 1.2}$	16.56$_{\pm 1.4}$
FetchPick-es	**14.19**$_{\pm 0.9}$	4.16$_{\pm 2.0}$	1.67$_{\pm 0.8}$	6.49$_{\pm 1.7}$	3.17$_{\pm 0.3}$	4.16$_{1.3}$	13.12$_{\pm 0.8}$	5.96$_{\pm 2.2}$	12.28$_{\pm 1.4}$	11.82$_{\pm 1.4}$
Average	**19.73**	10.63	9.79	13.59	13.07	12.66	16.40	12.33	17.37	17.33

Environments and Datasets. To conduct benchmark experiments, we utilize offline datasets from (Yang et al., 2022b). The dataset contains 1×10^5 transitions for low-dimensional tasks and 2×10^6 for four high-dimensional tasks (FetchPush, FetchPick, FetchSlide and HandReach). Furthermore, to demonstrate the ability to handle small data budgets, we also integrate an additional group of small and extra small datasets, referred to with a suffix "-s" and "-es", containing only 10% and 1% of the number of transitions. To assess GOPlan's ability to generalize to OOD goals, we leverage four task groups from (Yang et al., 2023): FetchPush Left-Right, FetchPush Near-Far, FetchPick Left-Right, and FetchPick Low-High, each consisting of a dataset and multiple tasks. For instance, the dataset of FetchPush Left-Right contains trajectories where both the initial object and achieved goals are on the right side of the table. As such, the independent identically distributed (IID) task assesses agents handling object and goals on the right side (i.e., Right2Right), while the other tasks in the group assess OOD goals or starting positions, such as Right2Left, Left2Right, and Left2Left. For further information regarding the environments and datasets used in our evaluation, we refer readers to Appendix B.

Experimental Setup. We compare GOPlan against SOTA offline GCRL algorithms, WGCSL (Yang et al., 2022b), contrastive RL (CRL) (Eysenbach et al., 2022), actionable models (AM) (Chebotar et al., 2021), GCSL (Ghosh et al., 2021), and modified offline RL methods, such as TD3-BC (g-TD3-BC) (Fujimoto & Gu, 2021), exponential advantage weighting (GEAW) (Wang et al., 2018), Trajectory Transformer (TT) (Janner et al., 2021), Decision Transformer (DT) (Chen et al., 2021). We denote a variant approach of GOPlan as **GOPlan2**, which employs testing-time model-based planning with candidate actions from the GOPlan policy. The testing-time planning method aligns with the model-based planning approach described in Section 4.2. For all experiments, we report the average returns with standard deviation across 5 different random seeds. Implementation details are provided in Appendix C.

5.1 Benchmarks and Small Dataset Results

Table 1 demonstrates the performance of GOPlan and baselines in the benchmark tasks. As shown in the table, GOPlan shows improved performance over current SOTA model-free algorithms, achieving the highest average return on 8 out 10 tasks. Notably, unlike prior online model-based GCRL approaches (Charlesworth & Montana, 2020; Yang et al., 2021) that fail to handle high dimensional tasks like HandReach, GOPlan works well for these tasks. This perhaps can be attributed to the fact that GOPlan incorporates both the strengths of model-free advantage-weighted policy optimization and model-based reanalysis techniques.

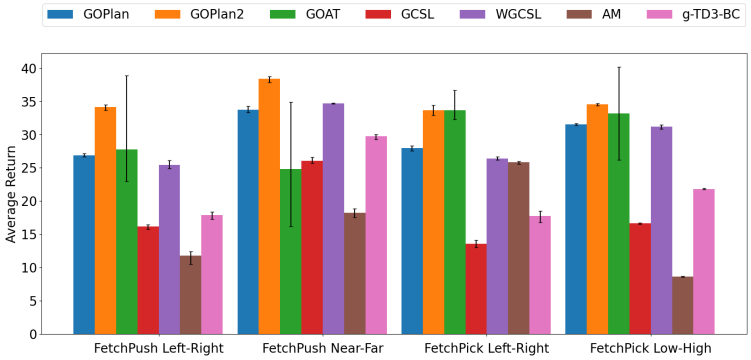

Figure 4: Average performance on OOD generalization tasks over 5 random seeds. The error bars depict the upper and lower bounds of the returns within each task group.

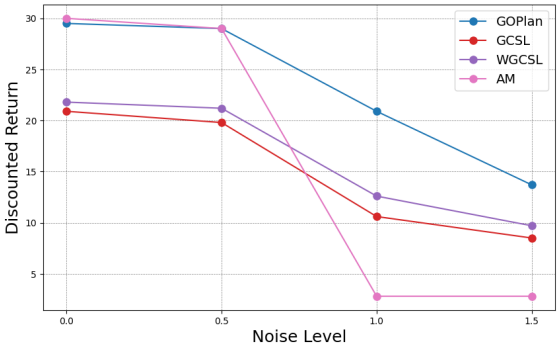

Figure 5: Stochastic environment evaluation.

Apart from the benchmark results, we conduct experiments under a small data setting that only has $\frac{1}{10}$th of the dataset used for benchmark experiments, and we observe that GOPlan surpasses other baselines by a significant margin. Specifically, GOPlan delivers an average improvement of 23% compared to the best-performing baseline CRL, and 38% compared to WGCSL. It is noteworthy that GOPlan alone achieves an average return over 10 on the challenging HandReach-s task, while other baselines fail on the HandReach-s task with small data budget and high-dimensional states.

5.2 Generalization to OOD Goals

We also conducted experiments on the four OOD generalization task groups in Figure 4. The results demonstrate that GOPlan outperforms other baselines. Specifically, by planning actions for OOD goals, GOPlan2 showcases significant improved performance, outperforming WGCSL by nearly 20% on average and enjoying less variance across IID and OOD tasks than recent SOTA method GOAT (Yang et al., 2023) for OOD goal generalization. The full results of this experiment can be found in Appendix D.1. The results obtained from our experiments provide empirical evidence in support of the effectiveness of using dynamics models for enhancing generalization towards achieving OOD goals.

5.3 Robustness to Noise

In this experimental setup, we investigated the robustness of various algorithms in modified FetchReach environments, which are subjected to different levels of Gaussian action noise applied to the policy action outputs. The datasets for these environments were obtained from Ma et al. (2022). As shown in Figure 5, GOPlan exhibits robust performance in this setup even under a noise level of 1.5.

5.4 Finetuning with Different Models

In offline GCRL, the reward function is always accessible for the agent, so we do not need to learn a reward model. However, it is feasible to learn a world model, including the transitions and the rewards. We design two variants: *GOPlan with reward models*, in which we do not use the environment-defined reward function, and instead train a reward model using the offline dataset with a binary cross entropy loss to predict the sparse reward; *GOPlan with RSSM*, in which we replace the environment-defined reward function and our learned transition models with the RSSM world model used in Dreamer (Hafner et al., 2019; 2020). Since the inference of the RSSM model consists of a stochastic latent code, we directly indicate the uncertainty by its standard deviation. In Figure 2, we evaluate the two variants with GOPlan in FetchPush, FetchPick and SawyerDoor. We found that GOPlan with reward models shows a similar performance as the vanilla GOPlan. However, GOPlan with RSSM shows slight performance decline. This decline might be attributed to the design of RSSM, which is tailored for processing high-dimensional image inputs, while our experiments primarily utilize low-dimensional state inputs.

	FetchPush	FetchPick	SawyerDoor
GOPlan	$39.15_{\pm 0.6}$	$37.01_{\pm 1.1}$	$44.42_{\pm 0.3}$
GOPlan with reward models	$38.75_{\pm 0.5}$	$37.10_{\pm 1.0}$	$44.32_{\pm 0.3}$
GOPlan with RSSM	$38.20_{\pm 0.4}$	$36.40_{\pm 0.8}$	$43.20_{\pm 0.5}$

Table 2: Comparsions between GOPlan, GOPlan with learned reward models, and GOPlan with RSSM.

5.5 Ablations

This section investigates the impact of different components on the performance of GOPlan.

Different prior policy for planning. Initially, we compare different models as the prior policy for planning. We denote planning with model "X" as "$X-$plan", and advantage-weighted CGAN as "ACGAN". Based on the results in Table 3, GOPlan2 and ACGAN-plan exhibit superior performance over Gaussian and VAE models, owing to the performant OOD action avoidance capabilities, which are advantageous for long-term planning. Furthermore, policies that incorporate advantage weighting, including ACGAN, Weighted CVAE, and Weighted Gaussian, have shown superior performance compared to their unweighted counterparts. This underscores the efficacy of the advantage-weighted policy approach and its potential for application in planning scenarios.

(a) FetchPick pretrain (b) SawyerDoor pretrain (c) FetchPick finetune (d) SawyerDoor finetune

Figure 6: Ablation studies of GOPlan. The static lines in (a) and (b) are the convergent performance of GOPlan, i.e., ACGAN after finetuning.

Ablation on the two stages of GOPlan. Next, we compare the performance of three different variants of GOPlan to assess the significance of the pretraining and finetuning stages in GOPlan. The variants include: (1) ACGAN (i.e., GOPlan without finetuning), (2) CGAN with $w(s, a, g) = 1$ in Eq.equation 7, (3) ACGAN-plan with model-based planning with action candidates from ACGAN. Results presented in Figures 6 (a)(b) demonstrate that ACGAN consistently outperforms CGAN in terms of average return.

Table 3: Comparison of different prior policies for planning.

Algorithm	FetchPick	FetchPush	SawyerDoor
GOPlan2	**38.20**$_{\pm 0.8}$	**39.75**$_{\pm 0.8}$	**45.13**$_{\pm 0.3}$
ACGAN-plan	**37.01**$_{\pm 1.1}$	38.42$_{\pm 0.9}$	**44.42**$_{\pm 0.3}$
CGAN-plan	33.95$_{\pm 2.3}$	36.51$_{\pm 2.6}$	42.04$_{\pm 1.0}$
weighted CVAE-plan	35.02$_{\pm 0.9}$	**39.11**$_{\pm 0.9}$	42.88$_{\pm 0.7}$
CVAE-plan	30.32$_{\pm 1.2}$	34.36$_{\pm 1.1}$	39.73$_{\pm 0.7}$
weighted Gaussian-plan	36.41$_{\pm 0.6}$	38.62$_{\pm 0.9}$	43.71$_{\pm 0.8}$
Gaussian-plan	29.95$_{\pm 1.0}$	30.97$_{\pm 1.5}$	36.61$_{\pm 0.5}$

Furthermore, ACGAN-plan is stable and efficient, but it imposes excessive computation for online interaction (30 Hz on GTX 3090). After finetuning ACGAN with our reanalysis methods, GOPlan achieves comparable performance as ACGAN-plan, and offers faster interaction (500 Hz).

Inter-trajectory and intra-trajectory reanalysis. During finetuning, both inter-trajectory and intra-trajectory reanalysis techniques are used in GOPlan. To assess the effectiveness of each approach, we compare the individual impacts and examine their joint influence on performance, as illustrated in Figures 6 (c)(d). Specifically, GOPlan generates 50% trajectories for each reanalysis, while "inter-trajectory"(or "intra-trajectory") performs 100% inter-trajectory (or intra-trajectory) reanalysis. Our findings are threefold: (1) inter-trajectory reanalysis can introduce a distribution shift that initially decreases performance, while intra-trajectory reanalysis has no such effect. (2) Both intra and inter-trajectory reanalysis lead to considerable improvement over the pretrained policy after sufficient finetuning steps. (3) The two reanalysis methods can work synergistically to yield the best performance.

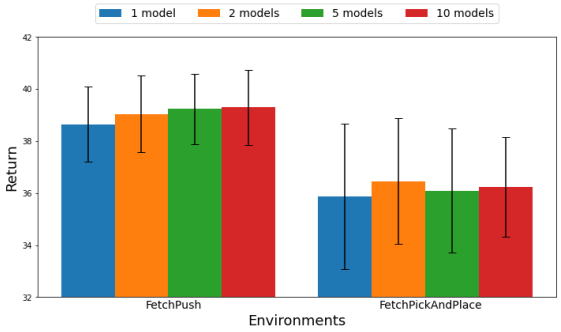

Figure 7: Ablation studies of GOPlan on different ensemble sizes.

Ensemble sizes and uncertainty truncation. We investigates the influence of different ensemble sizes of the dynamics models on the performance of GOPlan. We use $1, 2, 5, 10$ dynamics models in the finetuning stage to generate data for reanalysis. The results, presented in Figure 7, demonstrate the robustness of GOPlan to varying ensemble sizes, with all ensemble sizes exceeding the performance of a single dynamics model. This observation underscores the significance of uncertainty qualification through ensembles for GOPlan's performance. Notably, when only one model is employed, the inability to estimate uncertainty for truncating trajectories with high uncertainty leads to a degradation in performance. Furthermore, in the FetchPush task, where the variance in performance is relatively small, the average performance exhibits a stable improvement as the number of models increases.

6 Conclusions and Discussions

This paper proposes GOPlan, a novel model-based offline GCRL algorithm designed to effectively learn general goal-reaching policies from diverse and multi-goal offline datasets. GOPlan comprises two components:

pretraining a prior policy using an advantage-weighted CGAN and finetuning the policy with reanalysis. The advantage-weighted CGAN exhibits distinct mode separation and enhances the action distribution based on advantage values, thereby mitigating the OOD action issue and promoting long-term planning for offline GCRL. In addition, the reanalysis method generates high-performing and low-uncertainty imaginary samples by planning with learned models towards both intra-trajectory and inter-trajectory goals. Experimental results demonstrate that GOPlan achieves SOTA performance on offline GCRL benchmark tasks. Importantly, GOPlan yields greater improvements for challenging settings with limited data and OOD goal generalization, highlighting its potential advantages for practical scenarios.

Despite its promising features, GOPlan could be enhanced to efficiently learn from human demonstrations, which exhibit more diverse patterns than our testing benchmarks (Shafiullah et al., 2022; Lynch et al., 2019). Future research directions also include extending GOPlan to high-dimensional setting, which may require planning with advanced world models as proposed in (Janner et al., 2021; Wang et al., 2023; Janner et al., 2022), as well as planning in latent spaces as studied in (Hafner et al., 2019; Nguyen et al., 2021). In addition, researchers should consider its potential ethical issues when applying the technique to real-world applications, such as the misuse of technology and the safety in scenarios where mistakes could be hazardous.

References

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In *Advances in Neural Information Processing Systems*, 2021.

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

Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. In *International Conference on Learning Representations*, 2021.

Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhi-Hong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. In *International Conference on Learning Representations*, 2022.

Henry Charlesworth and Giovanni Montana. PlanGAN: Model-based planning with sparse rewards and multiple goals. In *Advances in Neural Information Processing Systems*, 2020.

Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jacob Varley, Alex Irpan, Benjamin Eysenbach, Ryan C Julian, Chelsea Finn, and Sergey Levine. Actionable models: Unsupervised offline reinforcement learning of robotic skills. In *International Conference on Machine Learning*, 2021.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, 2021.

Xi Chen, Ali Ghadirzadeh, Tianhe Yu, Jianhao Wang, Alex Yuan Gao, Wenzhe Li, Liang Bin, Chelsea Finn, and Chongjie Zhang. LAPO: Latent-variable advantage-weighted policy optimization for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. (Jeroen) Donkers (eds.), *Computers and Games*, pp. 72–83, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-75538-8.

Marc Deisenroth and Carl E Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning*, 2011.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. RvS: What is essential for offline RL via supervised learning? In *International Conference on Learning Representations*, 2022.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Scott Fujimoto and Shixiang Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, 2019.

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

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020.

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, 2022.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, 2021.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis, 2022.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *International Conference on Machine Learning*, 2020.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on Robot Learning*, 2019.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Annual Conference on Robot Learning*, 2020.

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

Lina Mezghani, Sainbayar Sukhbaatar, Piotr Bojanowski, Alessandro Lazaric, and Karteek Alahari. Learning goal-conditioned policies offline with self-supervised reward shaping. In *Annual Conference on Robot Learning*, 2022.

Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Annual Conference on Robot Learning*, 2020.

Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. AWAC: Accelerating online reinforcement learning with offline datasets, 2021.

Tung D Nguyen, Rui Shu, Tuan Pham, Hung Bui, and Stefano Ermon. Temporal predictive coding for model-based planning in latent space. In *International Conference on Machine Learning*, 2021.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *International Conference on Machine Learning*, 2019.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019.

Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, Vikash Kumar, and Wojciech Zaremba. Multi-goal reinforcement learning: Challenging robotics environments and request for research, 2018.

Marc Rigter, Bruno Lacerda, and Nick Hawes. RAMBO-RL: Robust adversarial model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

Antoine Salmona, Valentin de Bortoli, Julie Delon, and Agnès Desolneux. Can push-forward generative models fit multimodal distributions?, 2022.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, December 2020.

Julian Schrittwieser, Thomas K Hubert, Amol Mandhane, Mohammadamin Barekatain, Ioannis Antonoglou, and David Silver. Online and offline reinforcement learning by planning with a learned model. In *Advances in Neural Information Processing Systems*, 2021.

Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=agTr-vRQsa`.

Hao Sun, Lei Han, Rui Yang, Xiaoteng Ma, Jian Guo, and Bolei Zhou. Exploit reward shifting in value-based deep-rl: Optimistic curiosity-based exploration and conservative exploitation via linear reward shaping. *Advances in Neural Information Processing Systems*, 35:37719–37734, 2022.

Sjoerd van Steenkiste, Klaus Greff, and Jürgen Schmidhuber. A perspective on objects and systematic generalization in model-based RL, 2019.

Quan Vuong, Aviral Kumar, Sergey Levine, and Yevgen Chebotar. Dual generator offline reinforcement learning, 2022.

Mianchu Wang, Yue Jin, and Giovanni Montana. Goal-conditioned offline reinforcement learning through state space partitioning. *Machine Learning*, Feb 2024.

Pengqin Wang, Meixin Zhu, and Shaojie Shen. Entropy: Environment transformer and offline policy optimization, 2023.

Qing Wang, Jiechao Xiong, Lei Han, peng sun, Han Liu, and Tong Zhang. Exponentially weighted imitation learning for batched historical data. In *Advances in Neural Information Processing Systems*, 2018.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning, 2019.

Rui Yang, Meng Fang, Lei Han, Yali Du, Feng Luo, and Xiu Li. MHER: Model-based hindsight experience replay. In *Deep RL Workshop NeurIPS 2021*, 2021.

Rui Yang, Chenjia Bai, Xiaoteng Ma, Zhaoran Wang, Chongjie Zhang, and Lei Han. RORL: Robust offline reinforcement learning via conservative smoothing. *Advances in Neural Information Processing Systems*, 35:23851–23866, 2022a.

Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking goal-conditioned supervised learning and its connection to offline RL. In *International Conference on Learning Representations*, 2022b.

Rui Yang, Yong Lin, Xiaoteng Ma, Hao Hu, Chongjie Zhang, and Tong Zhang. What is essential for unseen goal generalization of offline goal-conditioned RL? In *International Conference on Machine Learning*, 2023.

Shentao Yang, Zhendong Wang, Huangjie Zheng, Yihao Feng, and Mingyuan Zhou. A behavior regularized implicit policy for offline reinforcement learning, 2022c.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, 2020.

Xianyuan Zhan, Xiangyu Zhu, and Haoran Xu. Model-based offline planning with trajectory pruning. In *International Joint Conference on Artificial Intelligence*, 2022.

Wenxuan Zhou, Sujay Bajracharya, and David Held. PLAS: Latent action space for offline reinforcement learning, 2020.