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

Mianchu Wang[*]
University of Warwick
Mianchu.Wang@warwick.ac.uk

Rui Yang[*]
HKUST
ryangam@connect.ust.hk

Xi Chen
Tsinghua University
pcchenxi@tsinghua.edu.cn

Meng Fang
University of Liverpool
Meng.Fang@liverpool.ac.uk

1 Introduction

Model-based RL [3, 20, 11, 16] is a natural choice to address limited data budget [6, 2, 21] and achieve generalization [22, 29, 12] in offline reinforcement learning (RL). A recent notable technique, *reanalysis* [20, 21], has shown superior performance for both online and offline RL and has advanced model-based methods to overcome the two challenges. However, it is primarily designed for single-task RL and cannot be directly applied for offline goal-conditioned RL (GCRL) [27, 13], since

1. the presence of **multi-modal actions in multi-goal datasets** presents challenges in avoiding out-of-distribution (OOD) actions during long-term offline planning,
2. it remains unexplored to determine **the selection of goals for multi-goal reanalysis**, which can generate improved targets for the value function and policy to enhance performance.

We introduce *Goal-conditioned Offline Planning* (GOPlan), a novel model-based algorithm designed to address the limited data and the OOD generalization challenges in offline GCRL. GOPlan consists of two stages, a *pretraining stage* and a *reanalysis stage*:

1. **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-goal 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. Besides, a group of dynamics models is also learned during this stage.
2. **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**. We quantify the uncertainty of the planned trajectories based on the disagreement of the models [18, 29] to avoid going excessively outside the dynamics model's support. Data with small uncertainty from the reanalysis buffer are high-quality demonstrations that can enhance the agent's ability to achieve both in-dataset and out-of-dataset goals.

GOPlan iteratively executes planning to generate better data and finetunes the policy with advantage-weighted CGAN. The framework is shown in Figure 1. The experimental evaluations demonstrate its state-of-the-art (SOTA) performance on various offline GCRL tasks, its superior ability to handle small data budgets, and its generalization capability to out-of-distribution goals.

[*]Equal contribution.

Goal-conditioned Reinforcement Learning Workshop at NeurIPS, 2023.

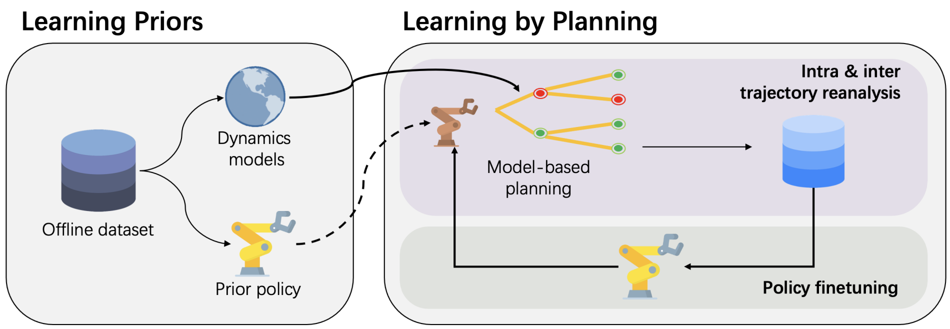


Figure 1: The two-stage framework of GOPlan: pretrain a prior policy and a group of dynamics models, and finetune policy with imagined trajectories generated by our multi-goal reanalysis method.

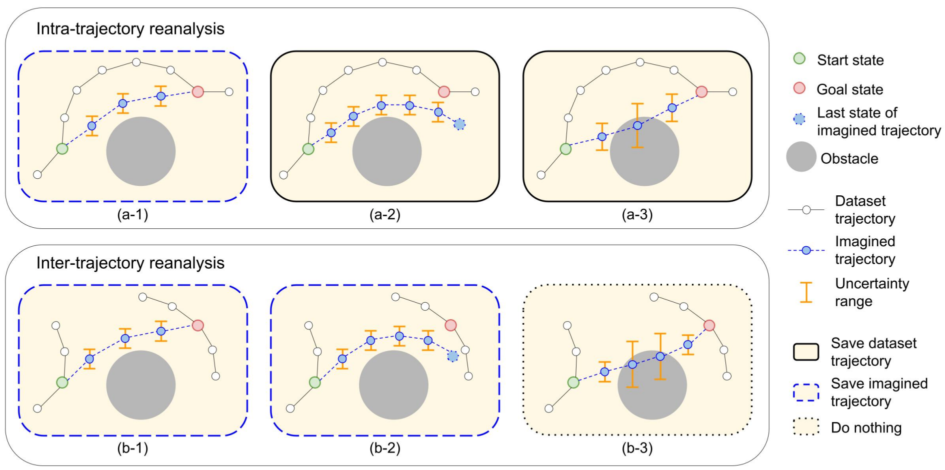


Figure 2: Illustration of intra-trajectory and inter-trajectory reanalysis. There are six scenarios: (a-1) the imagined trajectory is valid and better than the original trajectory; (a-2) the imagined trajectory fails to reach the goal within the same number of steps as the original trajectory; (b-1) 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.

2 Goal-conditioned Offline Planning

In this section, we present the two-stage GOPlan algorithm for offline GCRL. The pretraining stage introduces the advantage-weighted CGAN as an efficient 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.

2.1 Pretraining Stage

Due to the nature of collecting data for multiple heterogeneous goals, the multi-goal datasets can be highly multi-modal [14, 27], meaning that a state can have multiple valid action labels. Unlike prior works using Gaussian [27, 15] as a mean-seeking uni-modal policy, we propose exponentially advantage weighted CGAN, which can capture the multi-modal action distribution and produce in-distribution actions which are potential to maximise the expected return. A motivation example is shown in Appendix A. The training objective corresponds to the adversarial training objective [10]:

$$\max_D \min_\pi \mathbb{E}_{(s_t,a_t,g)\sim\mathcal{B}}\left[w(s_t,a_t,g)\log D(s_t,a_t,g)\right] + \mathbb{E}_{(s_t,g)\sim\mathcal{B},a'\sim\pi}\left[\log(1-D(s_t,a',g))\right] \quad (1)$$

Dataset	**Task**	**GOPlan**	**BC**	**GCSL** [9]	**WGCSL** [27]	**GEAW** [24]	**AM** [4]	**CRL** [7]	**g-TD3-BC** [8]
Normal (2×10^6 transitions)	FetchPush	$\mathbf{39.15}_{\pm 0.6}$	$31.56_{\pm 0.6}$	$28.56_{\pm 0.9}$	$39.11_{\pm 0.1}$	$37.42_{\pm 0.2}$	$30.49_{\pm 2.1}$	$36.52_{\pm 0.6}$	$30.83_{\pm 0.6}$
	FetchPick	$\mathbf{37.01}_{\pm 1.1}$	$31.75_{\pm 1.2}$	$25.22_{\pm 0.8}$	$34.37_{\pm 0.5}$	$34.56_{\pm 0.5}$	$34.07_{\pm 0.6}$	$35.77_{\pm 0.2}$	$36.51_{\pm 0.5}$
	FetchSlide	$10.08_{\pm 0.8}$	$0.84_{\pm 0.3}$	$3.05_{\pm 0.6}$	$\mathbf{10.73}_{\pm 1.0}$	$4.55_{\pm 1.7}$	$6.92_{\pm 1.2}$	$9.91_{\pm 0.2}$	$5.88_{\pm 0.6}$
	HandReach	$\mathbf{28.28}_{\pm 5.3}$	$0.06_{\pm 0.1}$	$0.57_{\pm 0.6}$	$26.73_{\pm 1.2}$	$0.81_{\pm 1.5}$	$0.02_{\pm 0.0}$	$6.46_{\pm 2.0}$	$5.21_{\pm 1.6}$
Small (2×10^5 transitions)	FetchPush	$\mathbf{37.31}_{\pm 0.5}$	$25.54_{\pm 1.0}$	$26.30_{\pm 0.7}$	$32.35_{\pm 0.9}$	$33.68_{\pm 1.9}$	$32.93_{\pm 0.6}$	$31.72_{\pm 1.5}$	$30.92_{\pm 0.6}$
	FetchPick	$\mathbf{32.85}_{\pm 0.3}$	$23.05_{\pm 1.0}$	$23.71_{\pm 1.4}$	$29.12_{\pm 0.2}$	$30.92_{\pm 0.5}$	$25.56_{\pm 3.5}$	$32.27_{\pm 0.8}$	$29.06_{\pm 4.0}$
	FetchSlide	$\mathbf{5.04}_{\pm 0.4}$	$0.31_{\pm 0.1}$	$0.98_{\pm 0.4}$	$0.22_{\pm 0.1}$	$0.30_{\pm 0.1}$	$1.97_{\pm 2.7}$	$4.74_{\pm 0.6}$	$0.16_{\pm 0.2}$
	HandReach	$\mathbf{10.11}_{\pm 1.4}$	$0.16_{\pm 0.1}$	$0.13_{\pm 0.1}$	$0.12_{\pm 0.1}$	$0.03_{\pm 0.0}$	$0.08_{\pm 0.1}$	$0.45_{\pm 0.3}$	$1.6_{\pm 2.3}$

Table 1: Average return with standard deviation on the benchmark with normal/small size dataset.

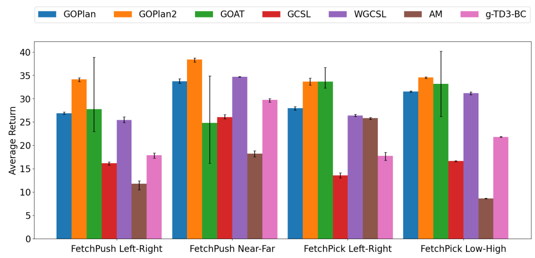


Figure 3: Average performance on OOD generalization tasks. The error bars depict the upper and lower bounds of the returns within each task group. GOPlan2 is a variant approach of GOPlan, which employs testing-time model-based planning with candidate actions from GOPlan policy.

Here, D is the discriminator, $w(s_t, a_t, g) = \exp(A^{\pi_b}(s_t, a_t, g))$ is the exponential advantage weight, π_b is the behavior policy underlying the relabeled offline dataset $\mathcal{B}$ [1]. The advantage A^{π_b} is estimated by a learned value function [27, 23]. In addition, a group of dynamics model $\{M_i\}_{i=1}^N$ is learned to predict the state residual [25, 17] for model-based planning and uncertainty quantification in the next stage.

2.2 Reanalysis Stage

With the learned prior policy and dynamics models, we reanalyse and finetune the policy in this stage. 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 diverse goals by reanalysing the current policy for both *intra-trajectory goals (goals along the same trajectory as the state) and inter-trajectory goals (goals lying in different trajectories)*. We formally define them here:

- **Intra-trajectory goals**: for any trajectory $\tau = \{s_0, a_0, \dots, s_T, g\} \in \mathcal{B}$ and any state $s_t \in \tau$, the intra-trajectory goals are $\{\phi(s_k) \mid T \geq k \geq t\}$, where ϕ is a state-to-goal mapping [13];
- **Inter-trajectory goals**: for any trajectories $\tau_1, \tau_2 \in \mathcal{B}$ and any state $s_t \in \tau_1$, the inter-trajectory goals for s_t are $\{\phi(s) \mid s \in \tau_2\}$.

Given the selected goals, we apply model-based planning method [3] to generate trajectories to reach them and fill a reanalysis buffer $\mathcal{B}_{re}$ with potential trajectories that can help improve the policy and have a small uncertainty less than u. The uncertainty of a generated trajectory τ is measured by the disagreement of the dynamics models: $U(\tau) = \max_{0 \leq t < T} \frac{1}{N} \sum_{i=1}^N ||M_i(s_t, a_t) - \bar{s}_{t+1}||_2^2$, where $\bar{s}_{t+1} = \frac{1}{N} \sum_{i=1}^N M_i(s_t, a_t)$. Figure 2 shows 6 scenarios of the imagined trajectories and presents the criteria to save them. After saving a number of imagined trajectories, we finetune the policy with the samples from $\mathcal{B}_{re}$. We repeat the processes to iteratively improve the policy. More details about GOPlan and the planning algorithm can be found in Appendix B.

3 Experiments

Through extensive experiments, we demonstrate GOPlan's SOTA performance on standard offline tasks, its superior ability to handle small data budgets, and its generalization ability to OOD goals. The benchmarks are detailed in Appendix C.

1. Table 1 demonstrates that GOPlan outperforms competitive baselines in the benchmark tasks [27] with both normal size ('Normal') and limited data budgets ('Small').
2. Figure 3 shows GOPlan and GOPlan2 outperforms other baselines and enjoys less variance in four OOD generalization task groups [26], where the testing goal may be not included in the dataset.

References

[1] 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.

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

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

[4] 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.

[5] 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.

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

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

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

[9] 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.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

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

[12] 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.

[13] Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. In *International Joint Conference on Artificial Intelligence*, 2022.

[14] 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*, 2020.

[15] 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.

[16] 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.

[17] Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *International Conference on Robotics and Automation*, 2018.

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

[19] Antoine Salmona, Valentin de Bortoli, Julie Delon, and Agnès Desolneux. Can push-forward generative models fit multimodal distributions?, 2022. In *Advances in Neural Information Processing Systems*, 2022.

[20] 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.

[21] 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.

[22] Sjoerd van Steenkiste, Klaus Greff, and Jürgen Schmidhuber. A perspective on objects and systematic generalization in model-based RL, 2019. In *arXiv preprint arXiv:1906.01035*, 2019.

[23] Mianchu Wang, Yue Jin, and Giovanni Montana. Goal-conditioned offline reinforcement learning through state space partitioning, 2023. In *arXiv preprint arXiv:2303.09367*, 2023.

[24] 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.

[25] 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.

[26] 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.

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

[28] Shentao Yang, Zhendong Wang, Huangjie Zheng, Yihao Feng, and Mingyuan Zhou. A behavior regularized implicit policy for offline reinforcement learning, 2022. In *arXiv preprint arXiv:2202.09673*, 2022.

[29] 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.

[30] Wenxuan Zhou, Sujay Bajracharya, and David Held. PLAS: Latent action space for offline reinforcement learning, 2020. In *Conference on Robot Learning*, 2021.

Appendix

A Conditioned GAN

The initial step of this section is to learn a policy capable of generating both 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 [14, 27], meaning that a state can have multiple valid action labels. These actions may even conflict with each other and make learning challenging. Unlike prior works using Gaussian [27, 15], conditional Variational Auto-encoder (CVAE) [30, 5] and Conditioned Generative Adversarial Network (CGAN) [28], we employ Weighted CGAN as the prior policy. In Figure 4, 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 4(a-2). The weighting scheme for Weight CVAE and Weighted CGAN is based on advantage re-weighting [27] and we directly use rewards as weights in this example.

In Figure 4, 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 [19]. 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 in an adversarial process. This mechanism encourages the policy to produce actions that closely resemble high-quality actions from the offline dataset.

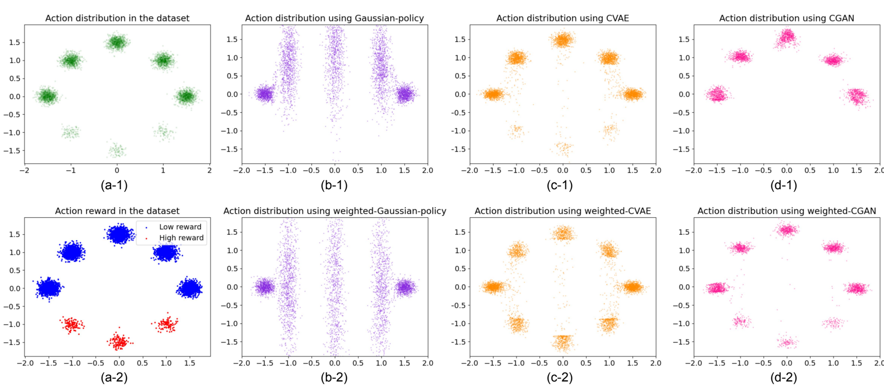


Figure 4: An example about modeling the multi-modal behavior policy while maximizing average rewards. The x-axis and the y-axis respectively represent the state and the multi-modal action. (a-1) and (a-2) show the action distribution and the the reward distribution of the offline dataset. (b-1)(b-2), (c-1)(c-2) and (d-1)(d-2) illustrate the action distributions generated by Gaussian and Weighted Gaussian, CVAE and Weighted CVAE, CGAN and Weighted CGAN, respectively.

B Algorithms

In this section, we present the pseudocode for GOPlan (Algorithm 1) and the model-based planning (Algorithm 2) used in GOPlan.

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 $\mathcal{B}$. ▷ TD-learning
    Update θ_d using $\mathcal{B}$. ▷ *max* in Eq. 1
    Update θ_π using $\mathcal{B}$. ▷ *min* in Eq. 1
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 $\mathcal{B}_{re}$. ▷ TD-learning
    Finetune θ_d using $\mathcal{B}_{re}$. ▷ *max* in Eq. 1
    Finetune θ_π using $\mathcal{B}_{re}$. ▷ *min* in Eq. 1
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 $\{s_0, \hat{s}_1, ..., \hat{s}_T, \phi(\hat{s}_T)\}$

Algorithm 2 Model-based Planning.

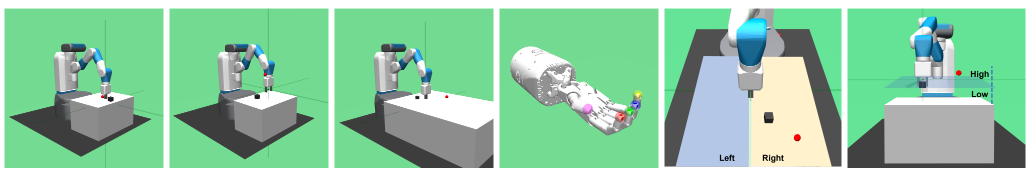
Initialise: N dynamics models $\{M_i\}_{i=1}^N$, policy π; current state s_0, goal g, reward function r.

for $c = 1, ..., C$ **do**
    $z \sim \mathcal{N}(0, 1)$
    $a_0^c = \pi(s_0, g, z)$ ▷ Sample C initial actions $\{a_0^c\}_{c=1}^C$.
    $i \sim$ Uniform$(1, \ldots, N)$
    $\hat{s}_1^c = M_i(s_0, a_0^c)$ ▷ Predict C next states $\{\hat{s}_1^c\}_{c=1}^C$.
    for $h = 1, ..., H$ **do**
        $\hat{s}_{h,1}^c = \hat{s}_1^c$ ▷ Duplicate every next state H times.
        for $k = 1, ..., K$ **do**
            $z \sim \mathcal{N}(0, 1)$
            $a_{h,k}^c = \pi(\hat{s}_{h,k}^c, g, z)$
            $i \sim$ Uniform$(1, \ldots, M)$
            $\hat{s}_{h,k+1}^c = M_i(\hat{s}_{h,k}^c, a_{h,k}^c)$ ▷ Generate H trajectories of K steps.
        end for
        $R_{c,h} = \sum_{k=0}^K r(\hat{s}_{h,k}^c, a_{h,k}^c, g)$
    end for
    $R_c = \frac{1}{H} \sum_{h=1}^H R_{c,h}$ ▷ Average all cumulative returns.
    $R_c = \frac{R_c}{\sum_{c=1}^C R_c}$ ▷ Normalize all cumulative returns.
end for
$a^* = \frac{\sum_{c=1}^C e^{\kappa R_c} \cdot a_0^c}{\sum_{c=1}^C e^{\kappa}}$ ▷ Exponentially weight the actions.
return a^*

C Environments and Datasets

We utilize the offline datasets from [27] to conduct benchmark experiments, as illustrated in (a-d) of Figure 5. The offline datasets, including 2×10^6 transitions, are collected by a pre-trained policy using DDPG and hindsight relabelling [1], where the actions from the policy are perturbed by adding Gaussian noise with zero mean and 0.2 standard deviation to increase the diversity and multi-modality of the dataset. Detailed information about the environments can be found in Appendix F of [27]. Furthermore, to demonstrate the ability to handle small data budgets, we integrate an additional group of small datasets, each containing only $\frac{1}{10}$ of the number of transitions.

To assess GOPlan's ability to generalize to OOD goals, we leverage four task groups from [26]: FetchPush Left-Right, FetchPush Near-Far, FetchPick Left-Right, and FetchPick Low-High, each consisting of a dataset of $5,000$ transitions. 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. In Figure 3, for the FetchPush Left-Right task group, we report the mean, the lowest, and the highest returns within Right2Right, Right2Left, Left2Right, and Left2Left. The behaviour policy used to collect the dataset is the same as the aforementioned policy except with a 30% probability of taking random actions. For further information regarding the ODD benchmark, we refer readers to Appendix C in [26].



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

Figure 5: Goal-conditioned tasks. (a) FetchPush, (b) FetchPick, (c) FetchSlide, (d) HandReach, (e) Push Left-Right, and (f) Pick Low-High.