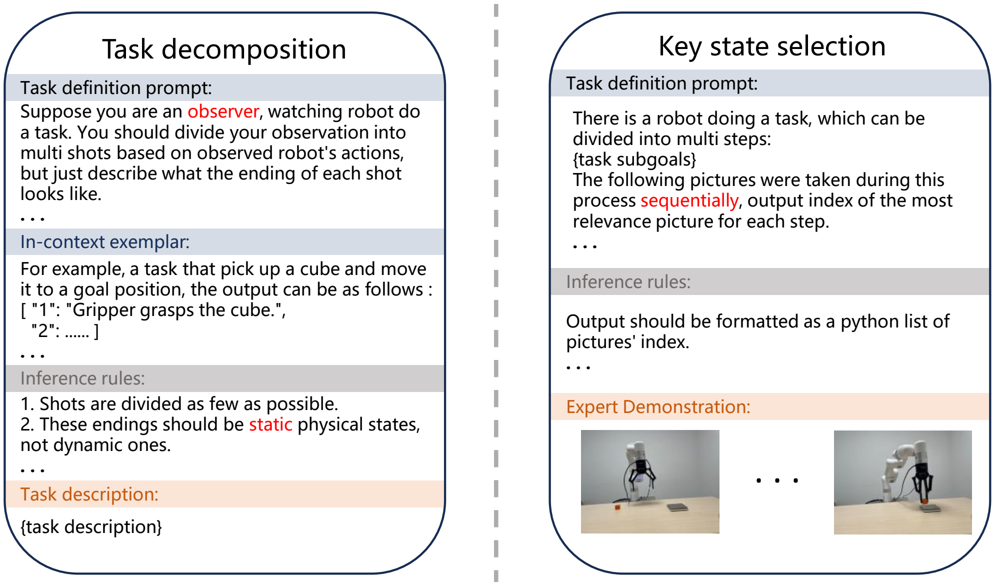
KOI: Accelerating Online Imitation Learning via Hybrid Key-state Guidance (Supplementary Materials)

A Details of Semantic Decomposition Module



Task decomposition

Task definition prompt:

Suppose you are an observer, watching robot do a task. You should divide your observation into multi shots based on observed robot's actions, but just describe what the ending of each shot looks like.

. . .

In-context exemplar:

For example, a task that pick up a cube and move it to a goal position, the output can be as follows :
["1": "Gripper grasps the cube.",
"2": ......]

. . .

Inference rules:

1. Shots are divided as few as possible.
2. These endings should be static physical states, not dynamic ones.

. . .

Task description:

{task description}

Key state selection

Task definition prompt:

There is a robot doing a task, which can be divided into multi steps:
{task subgoals}
The following pictures were taken during this process sequentially, output index of the most relevance picture for each step.

. . .

Inference rules:

Output should be formatted as a python list of pictures' index.

. . .

Expert Demonstration:

. . .

Figure 1: The pre-defined prompt in our Semantic Decomposition Module. Some keywords can enhance the quality of decomposition and selection.

In the Semantic Decomposition Module (SDM), we employ GPT-4v to extract the semantic key states from expert demonstrations, whose rich world knowledge and strong generalization capabilities have been proven in various visual-language tasks [1, 2]. The process of extraction includes two stages: task decomposition and key state selection.

The first stage takes the description of the task as input, decomposing the manipulation process into multiple subgoals. Concretely, we use the name of the task as the task description. These subgoals will be the criteria for selecting key states from expert demonstrations.

The key state selection stage would take the observation queries and subgoal descriptions into GPT-4v, and get the key state indexs. The prompts for the two stages are shown in Figure 1. Due to the API and context length restrictions, we select states from an expert demonstration every T step to form the query expert observation sets. In our experiment, T is assigned as 10. Empirically, we found that there will be chronological confusion if only one subgoal is passed in each query, *e.g.*, selecting the fifth frame for the first subgoal while the third frame for the second subgoal. Passing all subgoals in one go solves this problem effectively.

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

B Additional Experimental Results in Meta-World

In addition to the results provided in Section 4.2, Figure 2 shows the performance of KOI on 3 tasks from the Meta-World suite [3]. On all tasks, KOI has shown its sample efficiency compared to other reward estimation methods [4, 5, 6]. However, the “door unlock” task, as shown in Figure 4, requires the robot arm to rotate the handle to unlock the door. However, during online exploration, the agent could complete this task by using its body instead of its gripper to rotate the handle, due to the imprecise physical simulation. The tricky policy can complete the task without imitating the expert’s trajectory, resulting in the degradation of performance during online finetuning with any reward estimation method as compared to the pretrained policy.

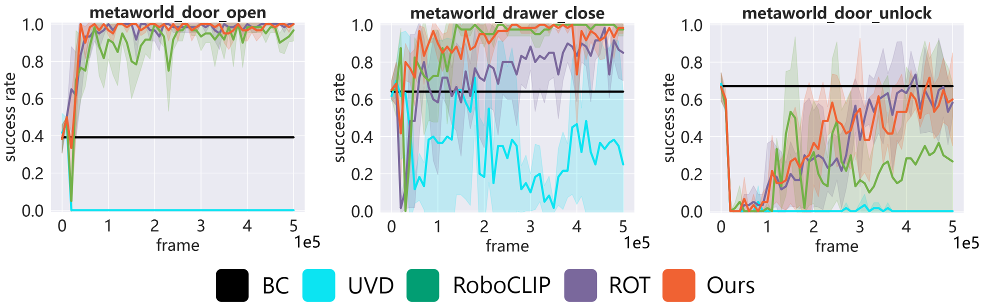


Figure 2: The experiment results on 3 tasks from Meta-World environments. The shaded region represents ± 1 standard deviation across 3 seeds. We notice that KOI method excels in exploration sample efficiency compared to prior work.

C Experiment Details

C.1 Hyperparameters

The complete list of hyperparameters is provided in Table 1. As shown, all the methods differ only in reward estimation, *i.e.*, components like encoder and RL backbone are the same. When modeling the importance weight of expert demonstration, we apply distinct weights and standard deviations for semantic and motion key states, respectively. The weight assigned to the ultimate goal of the task is set higher than that of the subgoals, thereby enhancing the agent’s awareness of task completion.

C.2 Environments

Considering the complex scenarios, large action spaces and long task sequences of LIBERO suite [7], proprioceptive information is utilized as input. Besides, 50 expert demonstrations are borrowed from their open-source release for BC training, while the estimation of the agent’s exploration reward is only based on 5 of them, which reduces the time cost of computation reward. In addition, when the task is finished, each state of the successful exploration will gain a task-finish reward, to promote policy learning. More details can be found in Table 2. For a fair comparison to ROT [4], we train the policy using a stack of 3 consecutive RGB frames in Meta-World suite, and each action in the environment is repeated 2 times.

D Real-World Experiments

In this work, we conduct 3 real-world robotic manipulation tasks, as shown in Figure 4.

D.1 Implementaion Details

For each task, we gather five human expert demonstrations to train the BC policy and estimate online imitation rewards. To facilitate efficient policy learning, we limit the robot’s action space to four dimensions (x, y, z, gripper) and utilize both image and proprioceptive inputs for the policy. To ensure the safety during real-world exploration, we restrict the end-effector's position to a predefined target area.

As mentioned in the limitation, the mismatch between pretrained BC policy and initial critic would negatively impact early exploration learning. Consequently, to ensure both safety and efficiency during real-world exploration, we substitute the adaptive function λ in Equation 1 with a constant value of 0.9 and provide a sparse reward to indicate whether the task has been completed.

$$\pi_e = \arg\max_{\pi} \left[(1 - \lambda(\pi))\mathbb{E}_{(o_e,a_e)\sim T_e}[Q_\theta(o_e, a_e)] - \alpha\lambda(\pi_e)\mathbb{E}_{(o_d,a_d)\sim T_d} \|a_d - \pi(s_d)\| \right]. \quad (1)$$

In real-world scenarios, we first deploy the BC policy to evaluate the performance and collect a few interaction trajectories as an initial replay buffer. As shown in Figure 3(a), the BC policy fails to complete the task when objects are placed in different initial positions.

Further, we start online imitation learning using our hybrid key-state-guided imitation reward. During exploration, the agent explores action spaces not encountered in the expert trajectories. In such cases, our key-state-guided reward effectively corrects these deviations and guides the agent for task-aware exploration.

After 10 interaction trajectories, totaling nearly 4,000 timesteps, our agent can successfully complete the task even when objects are placed in different positions.

Please refer to the one-take video in the supplementary video for specific details.

Method	**Parameter**	**Value**
Common	Replay buffer size	150000
	Learning rate	$1e^{-4}$
	Discount γ	0.99
	n-step returns	3
	Mini-batch size	256
	Agent update frequency	2
	Critic soft-update rate	0.01
	Feature dim	50
	Hidden dim	1024
	Optimizer	Adam
	Exploration steps	0
	DDPG exploration schedule	0.1
	Target feature processor update frequency(steps)	20000
	Reward scale factor	10
	Fixed weight α	0.03
	Linear decay schedule for $\lambda(\pi)$	linear(1,0.1,20000)
KOI	Weight of semantic key states A_i^s $(i < N_s - 1)$	0.15
	Weight of the last semantic key state $A_{N_s}^s$	0.35
	Weight of motion key states A^m	0.05
	Standard deviation of semantic key states σ_1	10
	Standard deviation of motion key states σ_2	25

Table 1: List of hyperparameters.

Suite	**Parameter**	**Value**
Meta-World	Use proprioceptive	False
	Image size	84 × 84
	Action shape	4
	Frame stack	3
	Action repeat	2
	Seed frames	12000
	Task finish reward	0
	Demonstration(s) for BC	1
	Demonstration(s) for online finetuning	1
LIBERO	Use proprioceptive	True
	Image size	128 × 128
	Action shape	7
	Frame stack	1
	Action repeat	1
	Seed frames	24000
	Task finish reward	5
	Demonstration(s) for BC	50
	Demonstration(s) for online finetuning	5

Table 2: Details of environment settings.

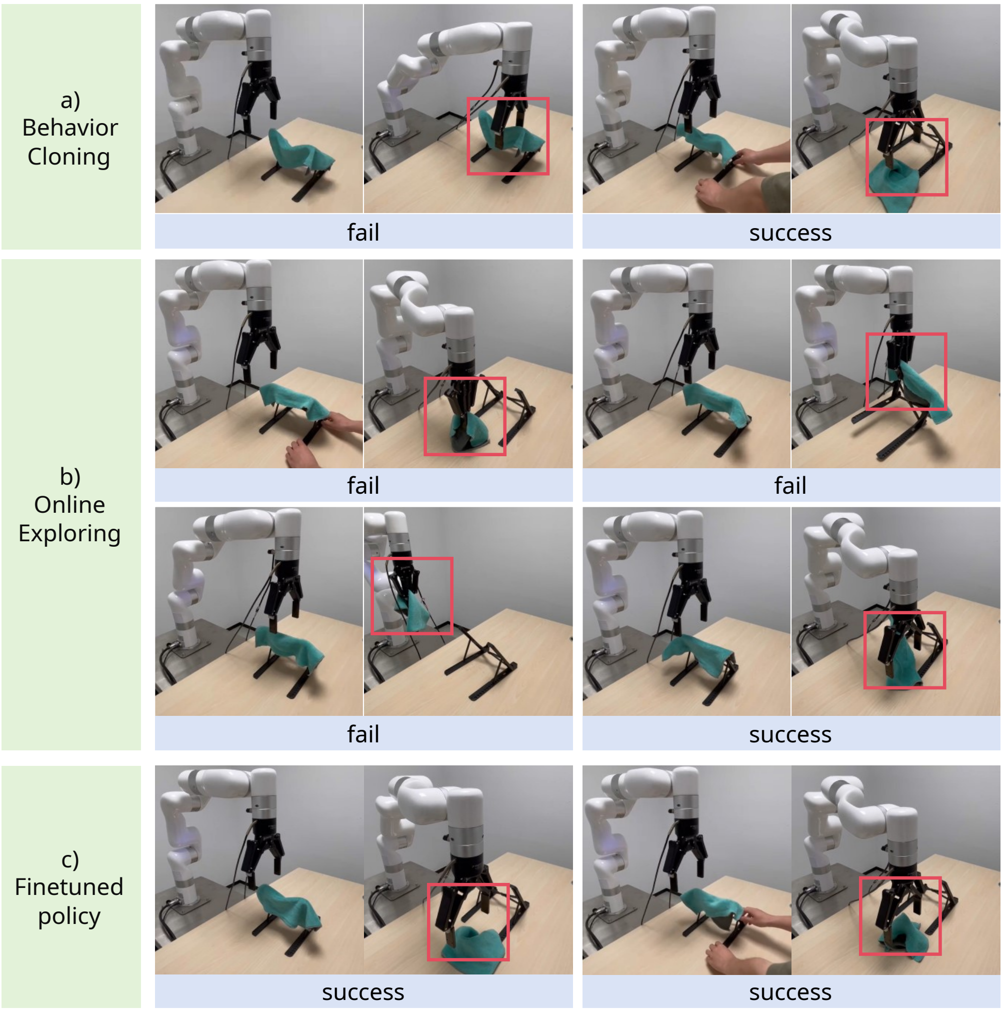


Figure 3: Overview of our real-world experiment on "take the rag off" task. KOI significantly accelerates the online finetuning, guiding the agent to complete the task even when objects are placed in different positions.

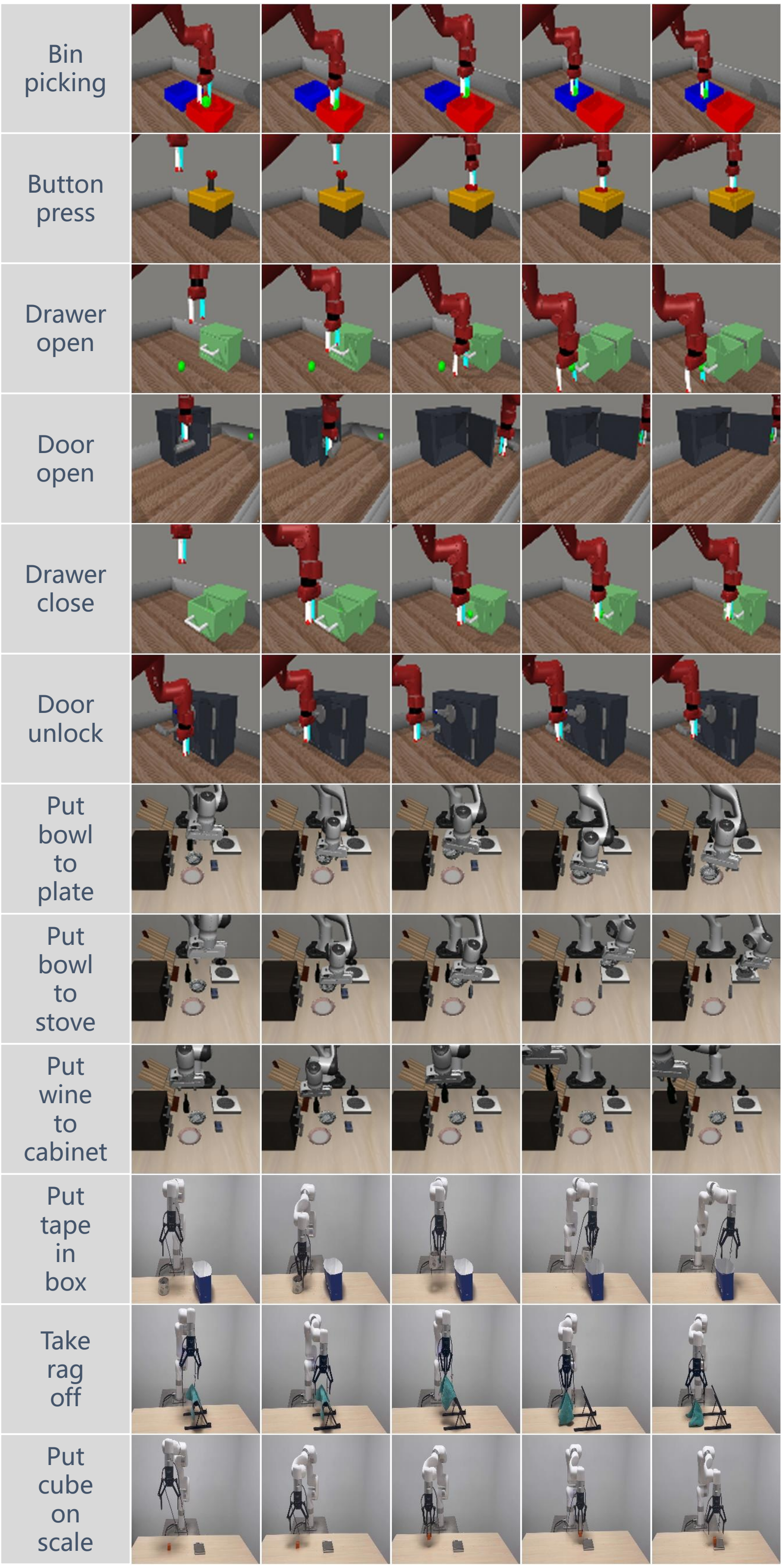


Figure 4: Example trajectories for 6 tasks from Meta-World suite, 3 tasks from LIBERO suite, and 3 real robot tasks, sequentially.

References

[1] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023.

[2] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik, et al. Language to rewards for robotic skill synthesis. *arXiv preprint arXiv:2306.08647*, 2023.

[3] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[4] S. Haldar, V. Mathur, D. Yarats, and L. Pinto. Watch and match: Supercharging imitation with regularized optimal transport. In *Conference on Robot Learning*, pages 32–43. PMLR, 2023.

[5] Z. Zhang, Y. Li, O. Bastani, A. Gupta, D. Jayaraman, Y. J. Ma, and L. Weihs. Universal visual decomposer: Long-horizon manipulation made easy. *arXiv preprint arXiv:2310.08581*, 2023.

[6] S. Sontakke, J. Zhang, S. Arnold, K. Pertsch, E. Bıyık, D. Sadigh, C. Finn, and L. Itti. Roboclip: One demonstration is enough to learn robot policies. *Advances in Neural Information Processing Systems*, 36, 2024.

[7] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36, 2024.