# Efficient Multi-Task Reinforcement Learning with Cross-Task Policy Guidance

## The 38th Conference on Neural Information Processing Systems

**Jinmin He, Kai Li*, Yifan Zang, Haobo Fu, Qiang Fu, Junliang Xing*, Jian Cheng**

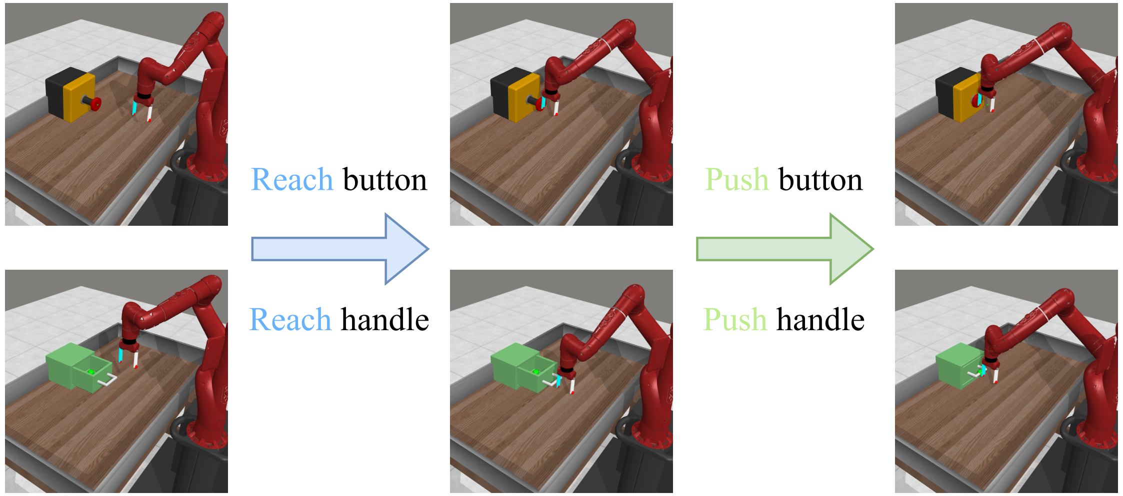
{hejinmin2021, kai.li, zangyifan2019, jian.cheng}@ia.ac.cn,

{haobofu, leonfu}@tencent.com, jlxing@tsinghua.edu.cn

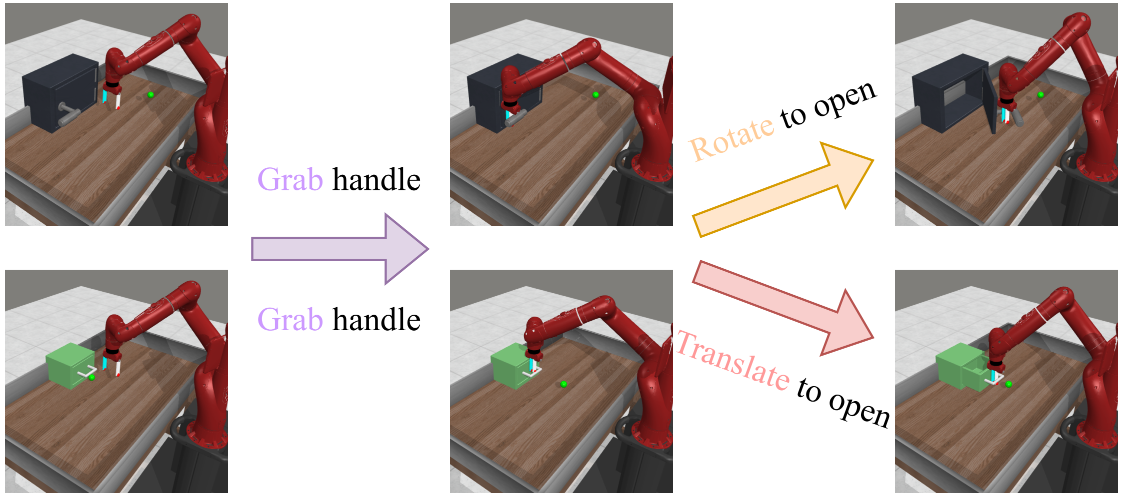
Institute of Automation, Chinese Academy of Sciences

# Motivation

CASIA




(a) *Button-Press* v.s. *Drawer-Close*

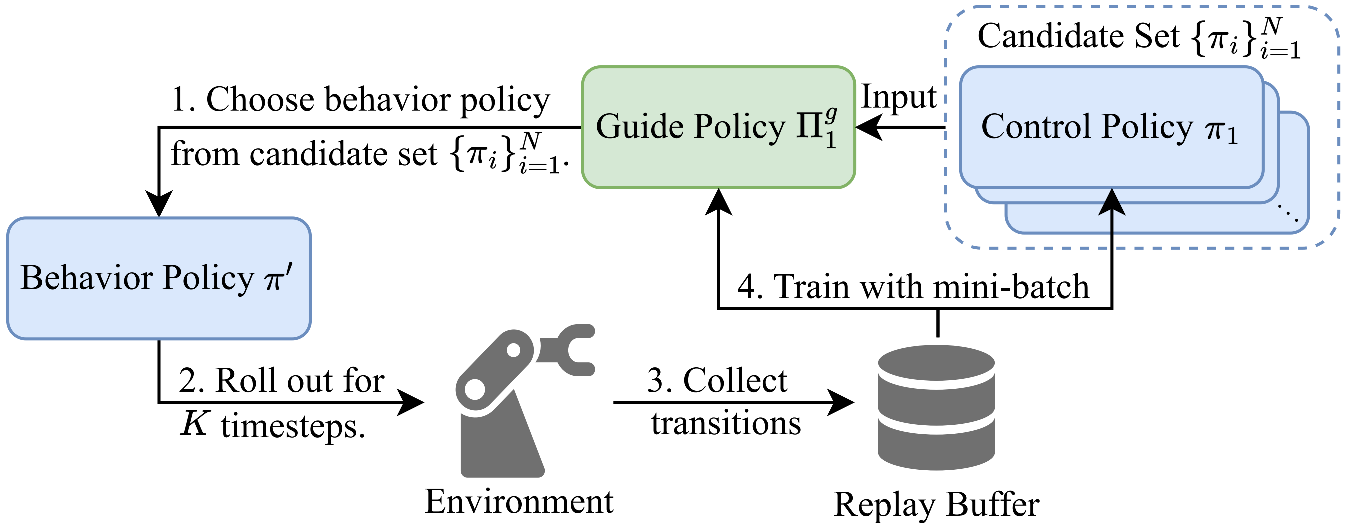



(b) *Door-Open* v.s. *Drawer-Open*

- MTRL enhances generalization by leveraging the information inherent in potentially related tasks.
- In addition to information sharing via network parameters, agents can also share via explicit policies.
- For humans, someone who can ride a bicycle can quickly learn to ride a motorcycle by referring to related skills, such as operating controls, maintaining balance, and executing turns.
- Similarly, full or partial policy sharing is also evident in robotic arm manipulation tasks.

# Cross-Task Policy Guidance

CASIA




- Instead of each task generating trajectories constantly by its corresponding control policy, we consider using control policies of other tasks to generate training data for the current task when appropriate.
- For task 1, its guide policy $\boldsymbol{\Pi_1^g}$ selects a policy $\boldsymbol{\pi'}$ from the candidate set $\{\boldsymbol{\pi_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{N}}$ every K timesteps. It then uses $\boldsymbol{\pi'}$ as the behavior policy to interact with the environment and collect data for next K timesteps.
- CTPG alters only the data collection process, without explicitly changing the training process.

# Guide Policy

CASIA

- Guide policy $\boldsymbol{\Pi_i^g}(\boldsymbol{j_t}|\boldsymbol{s_t})$ of task $\boldsymbol{i}$ outputs a task index $\boldsymbol{j_t} \in \boldsymbol{T}$, meaning using $\boldsymbol{\pi_{j_t}}$ as the behavior policy.
- The guide Q-value function is $\boldsymbol{Q_i^g}(\boldsymbol{s_t}, \boldsymbol{j_t})$ with its Bellman equation defined as:

$$\mathcal{B}^{\Pi_i^g} Q_i^g(s_t, j_t) \triangleq R_i^g(s_t, j_t) + \gamma^K \mathbb{E}_{j_{t+K}\sim\Pi_i^g, s_{t+K}\sim P_i}\left[Q_i^g(s_{t+K}, j_{t+K})\right]$$

- Reward function $\boldsymbol{R_i^g}$ is defined as the expected cumulative discount rewards:

$$R_i^g(s_t, j_t) = \mathbb{E}_{a_{t'}\sim\pi_{j_t}, s_{t'+1}\sim P_i}\left[\sum_{t'=t}^{t+K-1} \gamma^{t'-t} R_i(s_{t'}, a_{t'})\right]$$

- The trajectory generation process can be summarized as:

$$j_t \sim \Pi_i^g(\cdot|s_t), \qquad a_{t'} \sim \pi_{j_t}(\cdot|s_{t'}), \qquad \text{where } t' \in \{t, t+1, \ldots, t+K-1\}$$

- ***Hindsight Off-Policy Correction.*** The guide policy faces a non-stationary challenge during off-policy training. We reassign the action $\boldsymbol{j_t}$ sampled by the past guide policy to a new one $\boldsymbol{j_t'}$, whose control policy $\boldsymbol{\pi_{j_t'}}$ is more likely to output the historical action sequence $\{\boldsymbol{a_{t'}}\}_{\boldsymbol{t'=t}}^{\boldsymbol{t+K-1}}$.

$$j_t' = \arg\max_j \prod_{t'=t}^{t+K-1} \pi_j(a_{t'}|s_{t'}) = \arg\max_j \sum_{t'=t}^{t+K-1} \log \pi_j(a_{t'}|s_{t'}).$$

# Not All Policies Are Beneficial for Guidance

CASIA

- Some control policies perform even worse than the current task's own control policy $\boldsymbol{\pi_i}$.
- The trajectory generation solely using $\boldsymbol{\pi_i}$ can be regarded as equipped with a special guide policy $\boldsymbol{\Pi_i^{\tilde{g}}}$ that exclusively selects $\boldsymbol{\pi_i}$ as the behavior policy.

$$
\begin{aligned}
Q_i^{\tilde{g}}(s_t, i) &= R_i^g(s_t, i) + \gamma^K \mathbb{E}_{s_{t+K}\sim P_i}\left[Q_i^{\tilde{g}}(s_{t+K}, i)\right] \\
&= \mathbb{E}_{a_{t'}\sim\pi_i, s_{t'+1}\sim P_i}\left[\sum_{t'=t}^{t+K-1} \gamma^{t'-t} R_i(s_{t'}, a_{t'}) + \gamma^K Q_i^{\tilde{g}}(s_{t+K}, i)\right] \\
&= \cdots \\
&= \mathbb{E}_{a_{t'}\sim\pi_i, s_{t'+1}\sim P_i}\left[\sum_{t'=t}^{\infty} \gamma^{t'-t} R_i(s_{t'}, a_{t'})\right] \\
&= V_i(s_t),
\end{aligned}
$$

- We design a **Policy-Filter Gate** serving as a mask vector $\boldsymbol{m(s_t)}$

$$
m_j(s_t) = \begin{cases} 1, & Q_i^g(s_t, j) \geq V_i(s_t), \\ 0, & Q_i^g(s_t, j) < V_i(s_t), \end{cases} \quad \text{for } j \in \{1, 2, \ldots, N\},
$$

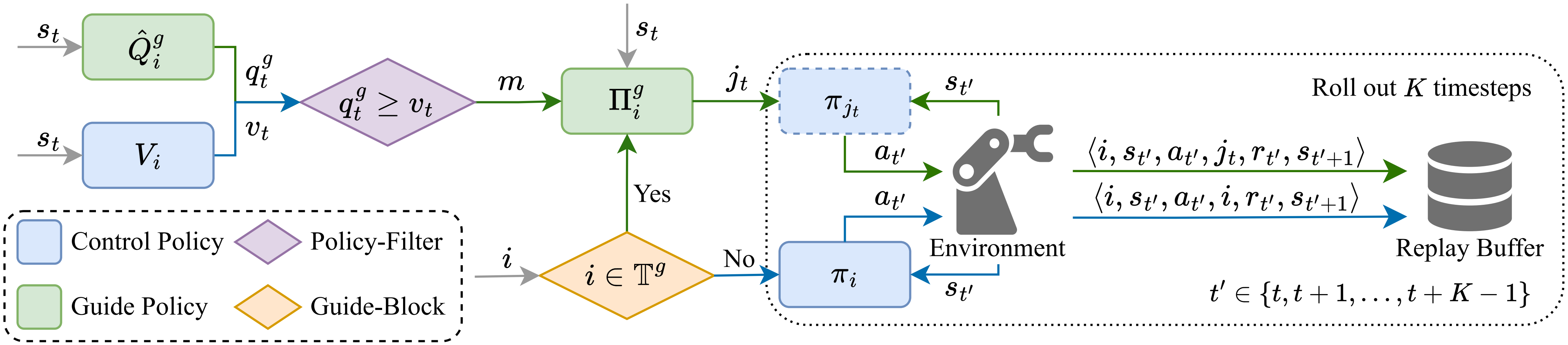
# Not All Tasks Need Guidance

- The easy tasks allow for the quick acquisition of some effective skills, which is helpful in exploring other tasks. However, they do not need additional guidance; instead, they focus on solidifying these skills.
- We design **Guide-Block Gate** to prevent guide policy from engaging in tasks that do not necessitate guidance. We form the tasks that require guidance into a subset $\mathbb{T}^g$ with SAC's temperature $\boldsymbol{\alpha_i}$.

$$\mathbb{T}^g = \left\{ i \middle| \log \alpha_i \leq \frac{1}{N} \sum_{j=1}^{N} \log \alpha_j \right\}$$

- For difficult tasks $i_{\text{diff}}$, their control policy entropies $H(\pi_{i_{\text{diff}}}(\cdot\,|s_t))$ tend to be high, and the corresponding temperature parameters $\alpha_{i_{\text{diff}}}$ decrease according to SAC's automatic temperature adjustment. Conversely, the temperature parameters $\alpha_{i_{\text{easy}}}$ increase for easy tasks $i_{\text{easy}}$. Therefore, $\alpha_i$ is a metric reflecting the relative difficulty and mastery of different tasks.
- We also considered using task success rate directly as a metric, and compared it in our experiments.

# Cross-Task Policy Guidance



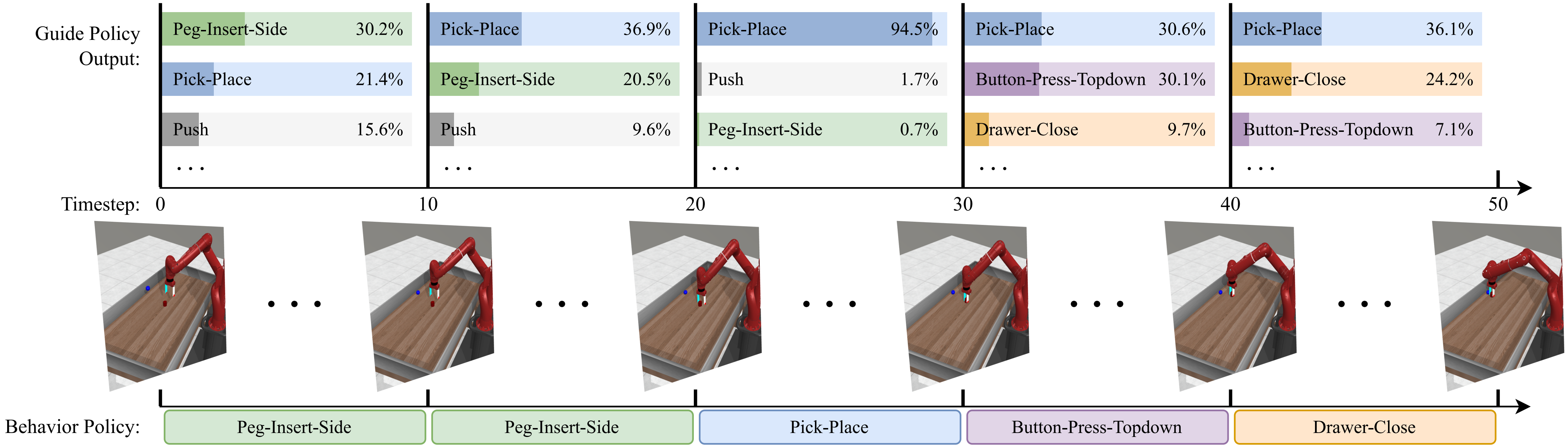

Here is the illustration of the comprehensive CTPG framework. Initially, the guide-block gate selectively provides guidance on tasks $i \in \mathbb{T}^g$. Subsequently, the policy-filter gate generates a mask $m$ to sift through the beneficial policies. Finally, the policy chosen by the guide policy or the control policy of the current task itself interacts with the environment over $K$ timesteps to collect training data.

# Experiments: Quantitative Result

CASIA

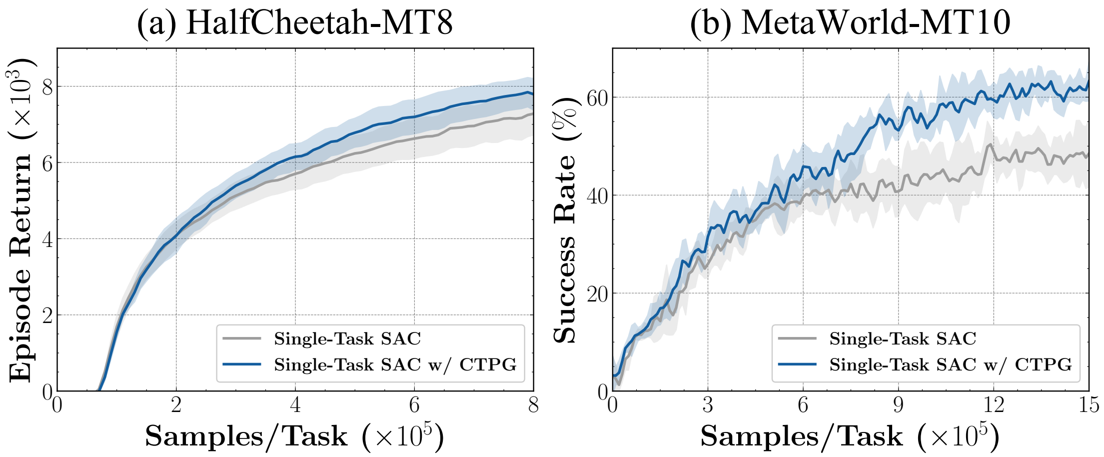
| Environment | Method | MTSAC | MHSAC | PCGrad | SM | PaCo |
|---|---|---|---|---|---|---|
| HalfCheetah MT5 ($\times$ 1e3) | Base | $9.16 \pm 0.42$ | $8.68 \pm 0.55$ | $9.57 \pm 0.73$ | $9.57 \pm 0.21$ | $7.18 \pm 0.44$ |
| | *w/* QMP | $8.81 \pm 0.22$ | $9.09 \pm 0.64$ | $9.46 \pm 0.57$ | $10.09 \pm 0.53$ | $7.83 \pm 0.28$ |
| | *w/* CTPG | $\mathbf{9.59 \pm 0.40}$ | $\mathbf{9.25 \pm 0.12}$ | $\mathbf{10.27 \pm 0.40}$ | $\boxed{\mathbf{10.47 \pm 0.34}}$ | $\mathbf{7.95 \pm 0.47}$ |
| HalfCheetah MT8 ($\times$ 1e3) | Base | $9.00 \pm 0.88$ | $8.90 \pm 0.60$ | $10.17 \pm 1.06$ | $10.05 \pm 0.55$ | $8.44 \pm 0.56$ |
| | *w/* QMP | $10.00 \pm 0.47$ | $9.61 \pm 0.54$ | $10.65 \pm 0.43$ | $10.41 \pm 0.61$ | $\mathbf{9.28 \pm 0.48}$ |
| | *w/* CTPG | $\mathbf{10.17 \pm 0.31}$ | $\mathbf{9.82 \pm 0.40}$ | $\boxed{\mathbf{11.09 \pm 0.50}}$ | $\mathbf{10.81 \pm 0.51}$ | $9.02 \pm 0.48$ |
| MetaWorld MT10 (%) | Base | $62.72 \pm 6.19$ | $63.51 \pm 2.97$ | $69.62 \pm 4.04$ | $74.52 \pm 2.29$ | $69.77 \pm 7.28$ |
| | *w/* QMP | $64.91 \pm 8.82$ | $65.87 \pm 3.05$ | $67.53 \pm 2.93$ | $69.78 \pm 7.50$ | $69.84 \pm 3.49$ |
| | *w/* CTPG | $\mathbf{75.76 \pm 3.82}$ | $\mathbf{74.94 \pm 2.97}$ | $\mathbf{73.31 \pm 3.66}$ | $\boxed{\mathbf{78.97 \pm 2.41}}$ | $\mathbf{70.40 \pm 3.62}$ |
| MetaWorld MT50 (%) | Base | $47.51 \pm 1.95$ | $52.04 \pm 2.78$ | $52.85 \pm 4.12$ | $55.04 \pm 2.84$ | $59.46 \pm 5.14$ |
| | *w/* QMP | $47.82 \pm 1.62$ | $51.79 \pm 4.83$ | $54.05 \pm 1.39$ | $55.91 \pm 5.08$ | $53.81 \pm 2.00$ |
| | *w/* CTPG | $\mathbf{55.97 \pm 2.56}$ | $\mathbf{56.91 \pm 2.57}$ | $\mathbf{58.91 \pm 2.10}$ | $\mathbf{66.24 \pm 3.37}$ | $\boxed{\mathbf{68.10 \pm 3.44}}$ |

CASIA

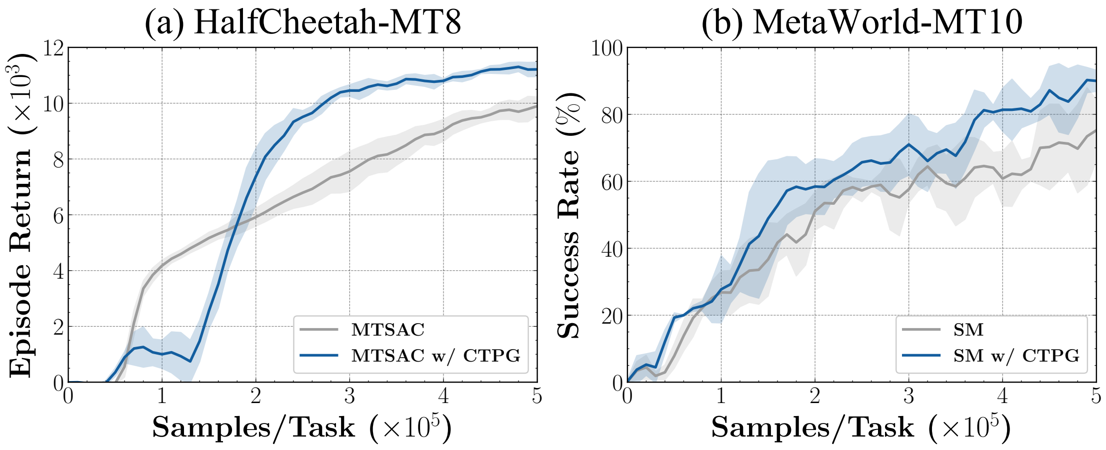



- We visualize one of the sampled trajectories of Task *Pick-Place* on *MetaWorld-MT10.*
- *Pick-Place* and *Peg-Insert-Side* employ a shared policy directing the robotic arm to target object.
- *Button-Press-Topdown* raises the gripper and then Drawer-Close moves forward.
- In the middle 10 timesteps, the probability of *Pick-Place* is notably high due to the absence of alternative shared policies at this stage.

# Other Experiments




CTPG without Implicit Knowledge Sharing




Exploration of New Tasks with CTPG

- CTPG improves performance without implicit knowledge sharing methods.
- We split the original task set in half, pre-training expert policies on the one half. While learning the other half, CTPG with expert policies can expedite the exploration of new tasks effectively.

# Thank You For Your Interest In Our Work

**The 38th Conference on Neural Information Processing Systems**