EMNLP 2025
Suzhou, China | 中国苏州

Yu Xia yux078@ucsd.edu

SAND: Boosting LLM Agents with Self-Taught Action Deliberation

Paper Link

Yu Xia[1] Yiran Shen[1] Junda Wu[1] Tong Yu[2] Sungchul Kim[2]
Ryan A. Rossi[2] Lina Yao[3,4] Julian McAuley[1]
[1]University of California San Diego [2]Adobe Research
[3]University of New South Wales [4]CSIRO's Data61

Motivation

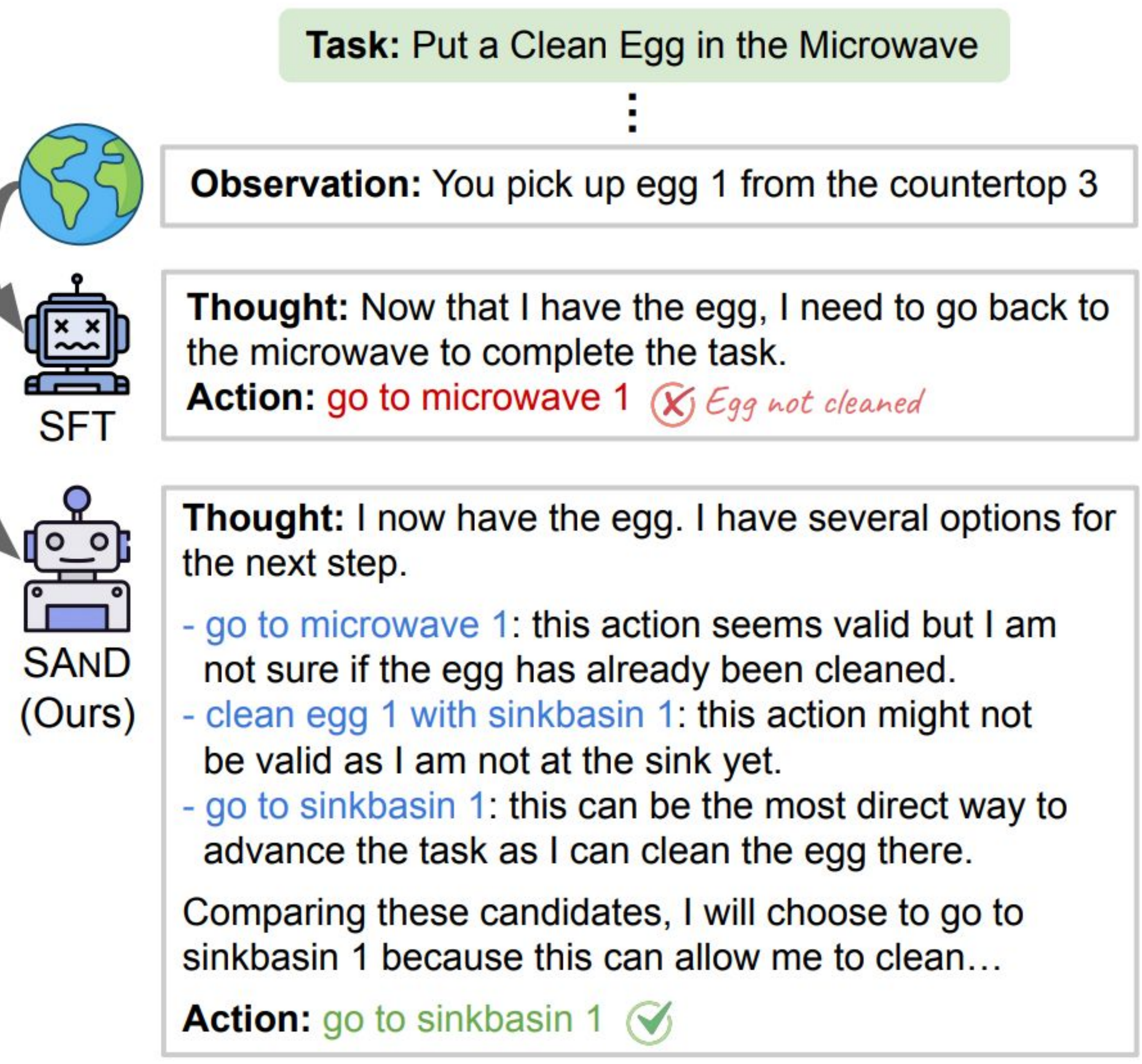
LLM agents are commonly **finetuned** with SFT on ReAct expert trajectories or preference optimization over pairwise rollouts.

Existing methods:

- *focus on **imitating** specific expert behaviors*
- *promote **chosen** reasoning actions **over rejected** ones*
- *may **over-commit** towards seemingly plausible but suboptimal actions due to limited action space exploration*

Our method:

- *enables LLM agents to explicitly **deliberate over candidate actions** before committing to one*
- *finetunes LLM agents with **self-synthesized** deliberation thoughts in an iterative manner*



Methodology

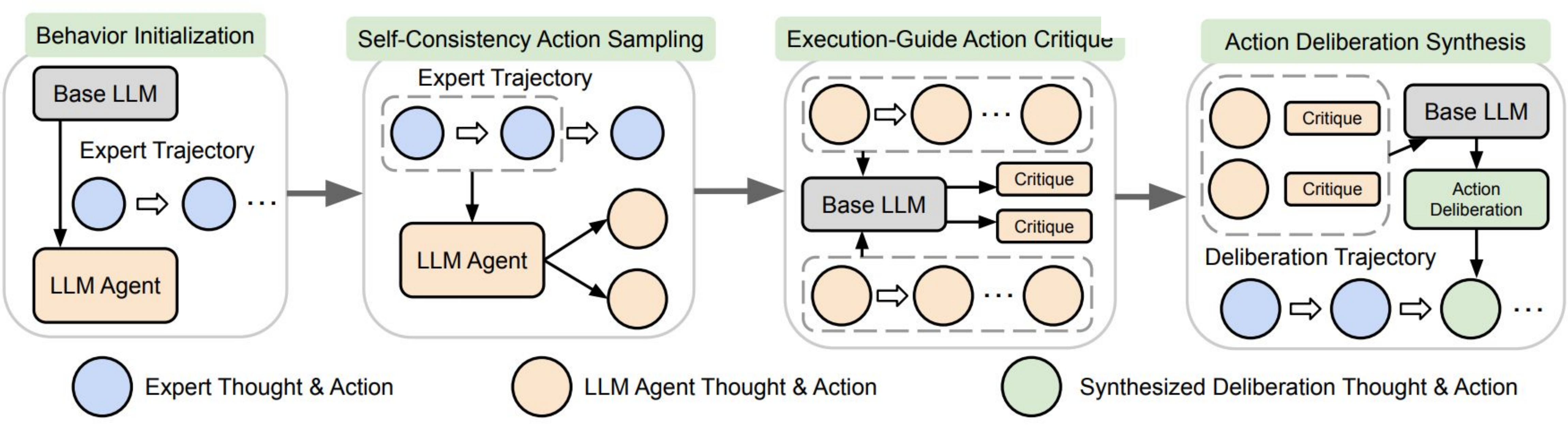


Figure 2: An illustration of our SAND framework for synthesizing one step of action deliberation thoughts.

Algorithm 1: Self-Taught Action Deliberation (SAND)

Input: $\mathcal{D}_{\text{exp}} = \{(u, z_1, a_1, o_1, \ldots, o_{L-1}, z_L, a_L)^{(i)}\}$: expert trajectories, I: number of self-taught iterations, N: number of sampled actions, π_{base}: base LLM, $\pi_\theta = \pi_{\text{base}}$: trainable LLM.
Output: Final LLM agent π_θ
Finetune π_θ on $\mathcal{D}_{\text{exp}}$: $\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{e\sim\mathcal{D}_{\text{exp}}}[\log \pi_\theta(e \mid u)]$
for $k = 1$ **to** I **do**
  $\pi_k \leftarrow \pi_\theta$, $\mathcal{D}_{\text{delib}} \leftarrow \emptyset$
  foreach $e = (u, z_1, a_1, o_1, \ldots, z_L, a_L) \in \mathcal{D}_{exp}$ **do**
    Initialize history $h_0 \leftarrow u$ and self-taught deliberation trajectory $\tilde{e} = (u)$
    for $t = 1$ **to** L **do**
      Sample N actions: $\{\hat{z}_t^{(n)}, \hat{a}_t^{(n)}\}_{n=1}^{N} \sim \pi_k(\cdot \mid h_{t-1})$
      if $|\{\hat{a}_t^{(1)}, \ldots, \hat{a}_t^{(N)}, a_t\}| = 0$ **then continue**
      Rollout each action: $\{\hat{e}_t, r_t\} \sim \pi_k(\cdot \mid h_{t-1}, \hat{z}_t, \hat{a}_t)$
      Generate critique for each action: $c_t \sim \pi_{\text{base}}(\cdot \mid \hat{a}_t, \hat{e}_t, r_t, \texttt{Prompt}_c)$,
      Synthesize action deliberation thought: $\tilde{z}_t \sim \pi_{\text{base}}(\cdot \mid \{(\hat{a}_t^{(n)}, c_t^{(n)})\}_{n=1}^{N+1}, \texttt{Prompt}_d)$
      $\tilde{e} \leftarrow \tilde{e} \cup (\tilde{z}_t, a_t, o_t)$; $h_t \leftarrow (h_{t-1}, z_t, a_t, o_t)$
    $\mathcal{D}_{\text{delib}} \leftarrow \mathcal{D}_{\text{delib}} \cup \{\tilde{e}\}$
  Finetune π_θ on $\mathcal{D}_{\text{delib}}$: $\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{\tilde{e}\sim\mathcal{D}_{\text{delib}}}[\log \pi_\theta(\tilde{e} \mid u)]$
  Set $\mathcal{D}_{\text{exp}} \leftarrow \mathcal{D}_{\text{delib}}$ for the next iteration
return π_θ

Experiments

- SAND **outperforms** existing agent tuning methods on SciWorld and ALFWorld (Table 2).
- Action deliberation **improves** LLM agents at **step-level** across iterations (Figure 3).
- LLM agents finetuned with SAND **learn when to deliberate** (Figure 4).
- For more results please refer to our paper.

Model	Single Agent	ScienceWorld		ALFWorld		Average
		Seen	Unseen	Seen	Unseen	
Agents w/ Training						
Qwen2.5-7B-Instruct + SFT (Zeng et al., 2024)	✓	69.2	60.8	72.1	75.4	69.4
Llama-3.1-8B-Instruct + SFT (Zeng et al., 2024)	✓	75.6	65.1	79.3	71.6	72.9
Llama-3.1-8B-Instruct + ETO (Song et al., 2024b)	✓	81.3	74.1	77.1	76.4	77.2
Llama-3.1-8B-Instruct + KnowAgent (Zhu et al., 2025)	✓	81.7	69.6	80.0	74.9	76.6
Llama-3.1-8B-Instruct + WKM (Qiao et al., 2024)	✗	82.1	76.5	77.1	78.2	78.5
Llama-3.1-8B-Instruct + ETO&MPO (Xiong et al., 2025)	✗	83.4	**80.8**	85.0	79.1	82.1
Qwen2.5-7B-Instruct + SAND (Iteration 1)	✓	80.9	67.2	85.7	85.0	79.7
Qwen2.5-7B-Instruct + SAND (Iteration 2)	✓	83.2	69.9	85.0	89.6	81.9
Qwen2.5-7B-Instruct + SAND (Iteration 3)	✓	84.0	69.0	90.7	94.8	84.6
Llama-3.1-8B-Instruct + SAND (Iteration 1)	✓	86.6	77.5	92.9	91.8	86.0
Llama-3.1-8B-Instruct + SAND (Iteration 2)	✓	**88.7**	78.2	**94.3**	94.0	88.8
Llama-3.1-8B-Instruct + SAND (Iteration 3)	✓	85.7	79.1	**94.3**	**96.3**	**88.9**

Table 2: Average rewards of all compared methods on two datasets. SAND significantly improves LLM agents across different model backbones, outperforming proprietary LLMs as well as state-of-the-art multi-agent approaches.

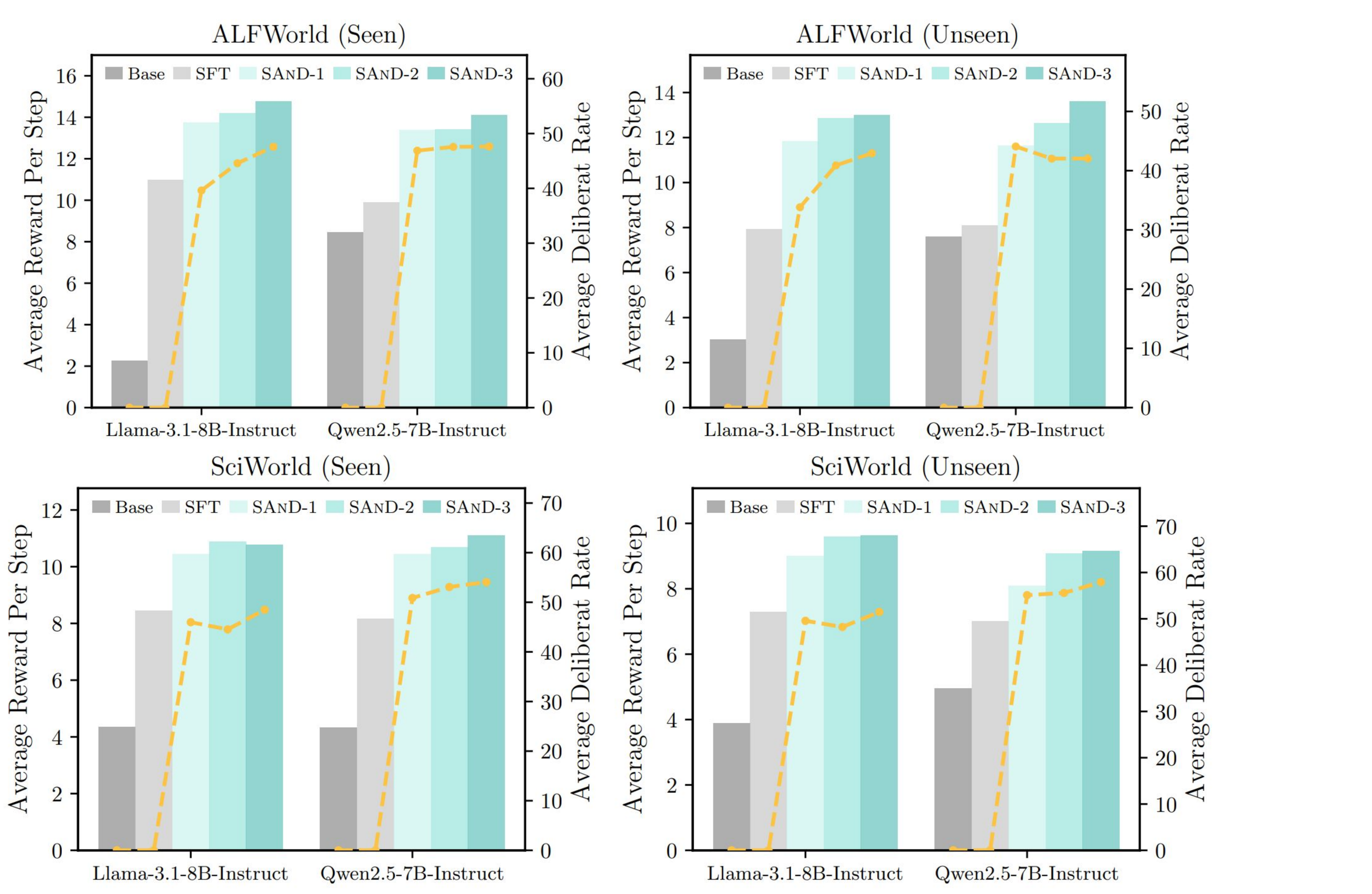


Figure 3: Average reward per step (bars) and average action deliberation rate per step (lines)

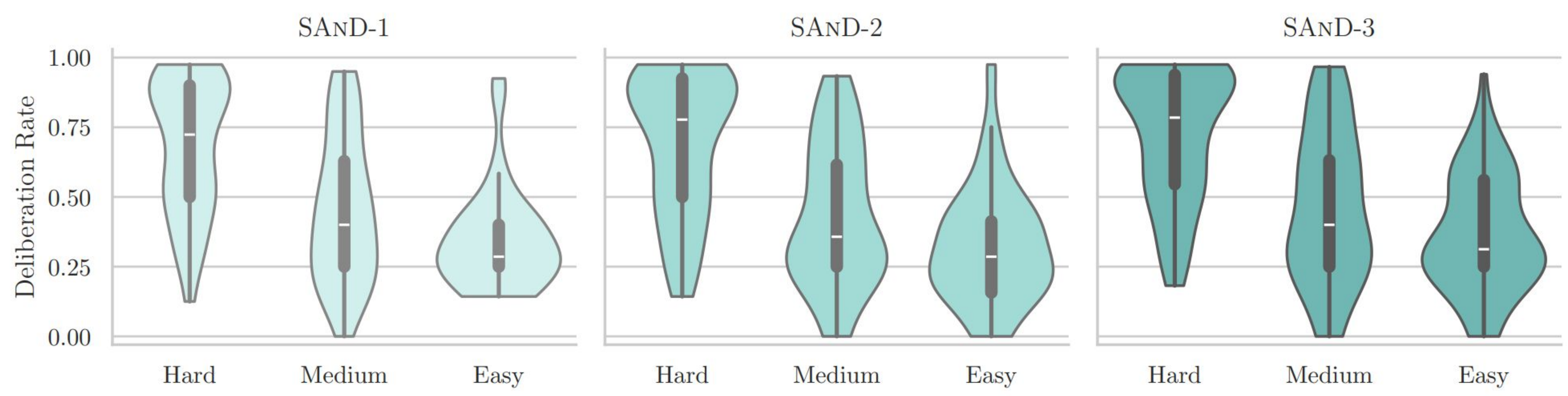


Figure 4: Action deliberation rate distribution across three difficulty bands in unseen test set on ScienceWorld. Each panel corresponds to a SAND iteration starting from Llama-3.1-8B-Instruct. The difficulty bands *Hard*, *Medium*, *Easy* are determined based on the tertiles of reward distribution from the base Llama-3.1-8B-Instruct. The results show that more SAND iterations teach LLM agents to deliberate more on hard tasks and less on easy tasks.