ICLR: In-Context Imitation Learning with Visual Reasoning

Toan Nguyen[1] Weiduo Yuan[1] Songlin Wei[1] Hui Li[2]
Daniel Seita[1,†] Yue Wang[1,†]
[1]University of Southern California [2]Autodesk Research

https://toannguyen1904.github.io/ICLR

***Abstract*— In-context imitation learning enables robots to adapt to new tasks from a small number of demonstrations without additional training. However, existing approaches typically condition only on state–action trajectories and lack explicit representations of task intent. This limitation hinders performance in complex and ambiguous task settings where the same actions may be consistent with different objectives. To address this, we present In-Context Imitation Learning with Visual Reasoning (ICLR), a novel framework that augments demonstration prompts with structured visual reasoning traces representing anticipated future robot trajectories in image space. ICLR also jointly learns to generate reasoning traces and low-level actions within a unified autoregressive transformer, enabling the model to mimic not only action prediction but also the reasoning process that leads to those actions. We extensively evaluate ICLR in both simulation and real-world manipulation tasks and demonstrate consistent improvements in success rates and generalization to unseen tasks and novel object configurations compared to other in-context imitation learning methods. These results suggest that incorporating embodied visual reasoning represents a promising direction for enhancing the robustness and generalization of robotic in-context learning systems.**

I. INTRODUCTION

A long-standing and significant challenge in robotics is data scarcity [1]. Collecting large-scale demonstration data for robotic manipulation in the real world is labor-intensive, time-consuming, and can pose several safety risks [2]. This has motivated the development of robot learning methods that can quickly acquire new skills from a limited number of robot demonstrations [3], [4], [5], [6], [7], [8]. One promising direction that has recently attracted significant attention is in-context imitation learning [9], [5], [10], [11]. In this learning paradigm, the robot learning model is trained to condition its behavior on a set of "in-context," or "prompt," demonstrations composed of state-action pairs. During inference, the robot can execute a previously unseen task by inferring the demonstrator's intent from only a few prompt demonstrations, without requiring any additional training.

Despite recent progress, existing robotic in-context imitation learning methods rely on state–action trajectories alone. By conditioning only on robot states (i.e., proprioceptive information and camera observations) and low-level actions, these approaches lack access to the underlying reasoning process that motivates the demonstrator's decisions. This omission becomes particularly problematic in complex and ambiguous task settings, such as environments with many objects and multiple plausible task objectives, where the same actions may be consistent with different intents. In such scenarios, we hypothesize that explicit reasoning is crucial for conveying high-level task intent and guiding the learning process beyond surface-level in-context action imitation.

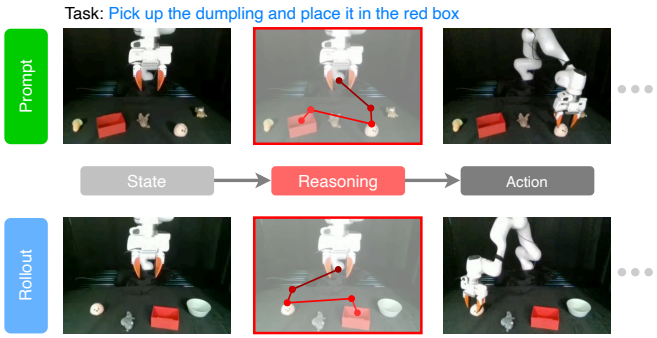

Fig. 1: **General framework overview.** Our method augments prompt demos with keypoint-based visual reasoning traces in the image space, shown above with the overlaid polyline in the middle column. During inference, the model also performs visual reasoning before predicting the subsequent low-level robot action. The task's language description is included for clarity.

In this work, taking inspiration from advances in chain-of-thought prompting for large language models (LLMs) and large vision-language models (VLMs) [12], [13], [14], we propose **I**n-**C**ontext imitation **L**earning with visual **R**easoning (ICLR), a transformer-based method incorporating embodied visual reasoning into robotic in-context imitation learning. Specifically, our approach augments prompt demonstrations with explicit visual reasoning traces, in addition to the robot's states and actions. These reasoning traces represent envisioned future robot trajectories in image space, capturing high-level task intent and providing structured guidance for action prediction. To execute a target task, conditioned on the augmented prompt demos, our method also generates the high-level visual reasoning traces before predicting the low-level control actions in an autoregressive manner. By learning to mimic not only robot actions but also the reasoning process underlying them, ICLR effectively grounds action prediction in structured task intent, leading to reliable adaptation to unseen tasks and strong generalization across visually complex scenarios, as validated via extensive experiments in both simulation and real-world settings. Figure 1 presents an overview of our method on the unseen task of putting the dumpling in the red box, given a single reasoning-augmented prompt demonstration.

† Equal advising

In summary, our main contributions include:

- We introduce ICLR, a novel in-context imitation learning method incorporating explicit embodied visual reasoning into demonstration prompts and policy inference.
- We evaluate our method through extensive experiments in both simulation and real-world robotic settings, demonstrating consistent performance improvements over competitive baselines and ablations.

II. RELATED WORK

A. In-Context Imitation Learning

In-context imitation learning has recently emerged as a promising approach for enabling robots to adapt to new tasks from a small number of test-time demonstrations without further training [15], [9], [10], [16], [5], [17]. A common recipe in existing in-context imitation learning methods is to construct prompt demos using only states or state-action pairs, and to directly predict actions for a target task. For instance, ICRT [5] treats in-context robot learning as a next-token prediction problem and introduces an autoregressive framework for robotic in-context learning conditioning on teleoperated prompt demos consisting of the robot's proprioception, camera images, and robot actions. Another example is Vid2Robot [16], which presents an encoder-decoder transformer that generates robot actions conditioned on the encoded representation of a human demo. While straightforward, this state-action formulation limits the robot's adaptability in cluttered and ambiguous environments, where the same actions may be consistent with different underlying task objectives and successful execution requires reasoning about task intent rather than direct action matching.

B. Robotic Embodied Reasoning

Recent work has explored incorporating reasoning processes into policy learning to improve robot navigation and manipulation in complex environments. Inspired by chain-of-thought reasoning in LLMs and VLMs [12], [13], [14], several approaches have investigated decomposing robotic tasks into intermediate reasoning steps, such as subgoals, plans, or structured representations, before executing low-level actions [18], [19], [20], [21], [22], [23]. Among different embodied reasoning representations, visual reasoning, such as predicted future end-effector trajectories in image space [21], [24], [25], [26], offers a promising alternative to other language-based representations, which can be ambiguous and less compatible with continuous robot actions. Applying embodied reasoning to in-context imitation learning, however, remains an under-explored research problem. In this work, alongside proposing a novel in-context imitation learning method with embodied visual reasoning, we systematically benchmark multiple strategies for embodied reasoning integration. The results show that our proposed method achieves the strongest performance, leading to substantially improved adaptation to unseen and complex manipulation tasks across both simulation and real-world environments.

III. PROBLEM STATEMENT

Our in-context imitation learning setting follows [5]. In particular, we consider a single-arm robot equipped with a standard gripper. There are two cameras: a third-view camera and a wrist-mounted camera. The goal is to train an in-context imitation learning method that can perform unseen tasks in novel environment configurations by conditioning on a few prompt demonstrations *without any further training*. Although the rollout environments allow the intended tasks demonstrated in the prompts, their configurations differ from those in the prompt demos, preventing the robot from naively copying the actions. Moreover, each testing configuration permits multiple possible tasks in addition to the desired one, requiring the model to infer the correct task objective from the prompt demos rather than relying on environmental cues alone. A prompt demo typically consists of RGB observations captured from two cameras, robot proprioception, and robot action. In our method and other baseline methods that use visual reasoning traces, we augment the prompt demos by incorporating visual reasoning traces generated from third-view camera images as detailed in Section IV-B.

IV. METHOD

In this section, we first outline the data formulation of in-context imitation learning. We then present the model architecture and implementation details of our ICLR method.

A. Training Data Formulation

For training in-context imitation learning policies, we consider a dataset $\mathcal{T}$ of visuomotor robot trajectories. In this dataset, each trajectory $\mathbf{T}^{\{i\}} = \{\mathbf{s}_1, \mathbf{a}_1, ..., \mathbf{s}_{t_i}, \mathbf{a}_{t_i}\}$ is a t_i-step sequence of states $\mathbf{s}$ (including multi-view camera observations and the robot proprioceptive information), and robot actions $\mathbf{a}$. We use the absolute end-effector pose as the robot's proprioception and the delta end-effector pose between consecutive time steps as the control action. To facilitate in-context learning, the dataset $\mathcal{T}$ is split into K disjoint subsets $\mathcal{T} = \bigcup_{k=1}^{K} \mathcal{S}_k$, $\mathcal{S}_k \cap \mathcal{S}_l = \emptyset$ for $k \neq l$. Each $\mathcal{S}_k$ is a subset of trajectories of a single task, as determined by the semantic labels associated with the trajectories. These labels are typically textual descriptions of the performed tasks, such as *"Poke the lion."* A training sequence is the concatenation of trajectories of the same subset, where the first randomly-selected n trajectories serve as prompt demos and the remaining trajectories are treated as target episodes.

B. Visual Reasoning Trace Generation

To incorporate visual reasoning into in-context imitation learning, we first augment each trajectory $\mathbf{T}^{\{i\}}$ of $\mathcal{T}$ with visual reasoning traces $\mathbf{r}$ generated from the third-view RGB observations in $\mathbf{o}$, forming the reasoning-augmented dataset $\mathcal{T}_{\text{aug}}$ of trajectories $\mathbf{T}^{\{i\}}_{\text{aug}} = \{\mathbf{s}_1, \mathbf{r}_1, \mathbf{a}_1, ..., \mathbf{s}_{t_i}, \mathbf{r}_{t_i}, \mathbf{a}_{t_i}\}$. The format of our visual reasoning traces follows MolmoAct [21]. In particular, at each time step in $\mathbf{T}^{\{i\}}_{\text{aug}}$, the visual reasoning trace is a polyline of 5 points corresponding to the robot gripper's positions in the pixel space of third-view observations, sampled evenly from the future horizon of the episode

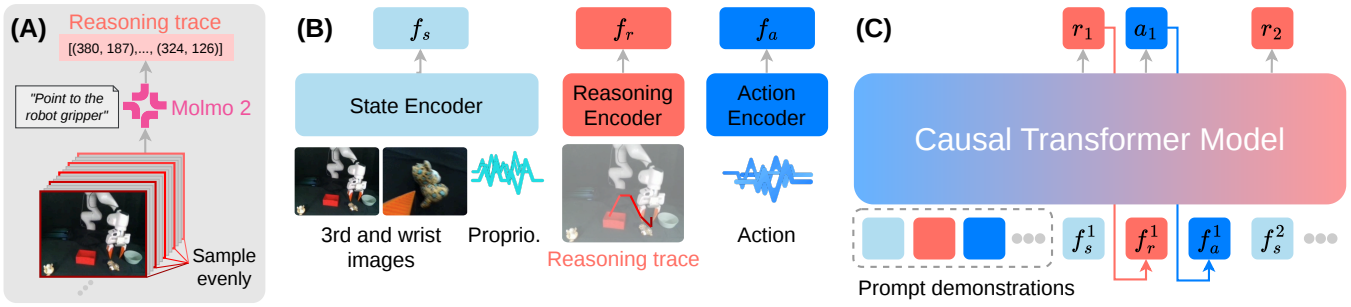

Fig. 2: **Method overview. (A)** To generate the visual reasoning trace at a given time step, we uniformly sample five third-view images from that time step to the end of the trajectory and use Molmo2 to predict the gripper's pixel location in each image. **(B)** Multi-view camera observations and proprioceptive states are encoded by a state encoder to produce state tokens f_s. Visual reasoning traces are embedded by a reasoning encoder to produce reasoning tokens f_r, and actions are embedded by an action encoder to produce action tokens f_a. **(C)** These modality-specific tokens are interleaved and fed into a causal transformer, which autoregressively predicts the next reasoning trace followed by the corresponding action. During training, teacher forcing is applied over reasoning and action tokens. In inference, the model first generates a reasoning trace and then produces the action in a closed-loop manner.

between the current and the terminal time step. We choose five points as a practical balance between granularity and efficiency for our experiments, which largely involve pick-and-place tasks (see Section V). In this setting, the five points naturally align with the four key stages of the behavior: moving to the target object, grasping it, transporting it to the receptacle, and placing the object. For longer-horizon tasks, the visual trace format can be designed to include more points to provide finer temporal resolution. Note that, in contrast to MolmoAct, where visual traces are represented in textual form, we represent visual traces as numerical vectors. To determine the gripper's position in an image, in simulation, we utilize the gripper's 3D position (inferred from the robot's proprioceptive state) and the known camera parameters. In the real world, since the camera parameters are usually unavailable, we employ the Molmo2 VLM [27] and prompt it with the command *"Point to the robot gripper."* In our work, we found that the gripper positions detected by Molmo2 are of high precision and facilitate our imitation learning with visual reasoning. Nevertheless, the generation of visual reasoning traces following this approach is not tied to a specific model and can be implemented using a wide range of VLMs [28], [29], segmentation models [30], [31], or detection models [32], [33]. In our work, we choose Molmo2 because it is open-source and has demonstrated its state-of-the-art performance on pointing tasks and usefulness for several downstream robotic applications [21], [34], [35].

C. In-Context Imitation Learning with Visual Reasoning

In the following, we describe the implementation details, training, and inference processes of our ICLR method.

Model Architecture. We adopt a Llama2-style [36] causal transformer architecture similar to [5], with modality-specific encoders for states, reasoning traces, and actions. Robot states are encoded using a state encoder. In particular, visual observations from the third-view and wrist cameras are first encoded using a pretrained vision transformer [5], while proprioceptive information is embedded by an MLP. We then employ attention pooling [37] to aggregate visual patch tokens and proprioceptive features to form the state token f_s. Visual traces, which are represented as ordered sets of keypoints, are flattened and encoded using an MLP reasoning encoder, producing reasoning tokens f_r. Similar to proprioception, the action tokens f_a are encoded by another MLP action encoder. All modality embeddings are interleaved into a single token sequence and processed by the transformer using next-token prediction. The overview of our model architecture is illustrated in Figure 2. Compared to ICRT [5], the key architectural difference lies in the inclusion of visual reasoning tokens f_r in the input and output sequence, enabling the transformer to jointly model reasoning and action generation within a unified autoregressive framework.

Loss Functions. We employ the standard next-token prediction objective, with losses applied only to predictions after the prompt demonstrations to preserve the in-context learning behavior. The loss is computed over both reasoning trace prediction and action prediction for target episodes. In particular, the combined loss is computed as

$$\mathcal{L} = \mathcal{L}_{\text{action}} + 0.3 \times \mathcal{L}_{\text{reasoning}}, \tag{1}$$

where $\mathcal{L}_{\text{action}}$ and $\mathcal{L}_{\text{reasoning}}$ are L1 losses for action prediction and reasoning trace prediction, respectively. We set the reasoning loss weight to 0.3, which empirically achieves a balanced trade-off between action and reasoning learning.

Training. During training, we freeze the pretrained vision transformer encoding camera images. We also apply action chunking [38], where the model is trained to predict the next 16 actions instead of a single one. In addition, we randomly mask a subset of the visual reasoning trace tokens in the target trajectories, while preserving all reasoning trace tokens in the prompt demonstrations. In particular, for each training sequence, we sample a random masking ratio from 0% to 100% and uniformly mask the corresponding number of reasoning tokens in the target portion of the sequence. This masking acts as a regularization technique that prevents the model's action prediction from over-relying on its generated reasoning traces, which are highly correlated with the actions but may be imperfect in challenging settings. As a result, the model learns to remain robust when reasoning traces are noisy or partially missing. This design, therefore, also naturally enables an efficient inference-time variant, which we refer to as reasoning dropout, inspired by similar

Method	LIBERO-Object	LIBERO-90			Avg.
		Kitchen	Living	Study	
ICRT [5]	44.44	0.89	<u>18.93</u>	0.83	16.27
TO Dropout	62.22	17.56	16.67	25.00	30.36
TO	54.44	12.11	12.80	29.33	27.17
Ours Dropout	**70.89**	**60.22**	**38.93**	**46.17**	**54.05**
Ours	<u>70.00</u>	<u>20.00</u>	11.20	<u>32.17</u>	<u>33.34</u>

TABLE I: **Simulation success rates (%).** "Dropout" models are models that learn to generate visual reasoning traces in training but skip the reasoning steps for the target trajectory in inference. Results are reported on 3 unseen tasks of LIBERO-Object and 15 unseen tasks of LIBERO-90 (6 of kitchen scenes, 5 of living room scenes, and 4 of study scenes). In each column, the highest score is **bolded**, and the second-best performance is <u>underlined</u>.

strategies in [39], [23], where the model omits reasoning trace generation during inference despite being trained to generate them. We adopt a teacher-forcing training scheme, where ground-truth reasoning and action tokens are used instead of the model's own predictions when conditioning the next-token prediction.

Inference. At test time, the human demonstrator provides one or more teleoperated demos consisting of state–action pairs. Molmo2 is used for detecting the robot gripper's positions in the third-view camera images concurrently with the teleoperation session. After an episode is completely recorded, the visual reasoning traces for that episode are generated following the visual trace generation process described in Section IV-B. Conditioned on the augmented prompt demonstrations and the current state, the model first predicts the next visual reasoning trace and then the corresponding action chunk, of which the executed immediate action is computed via temporal ensembling [38]. After executing each action, the policy receives the updated state, enabling it to iteratively generate the next reasoning trace before predicting and executing subsequent actions. We apply key-value caching [36] to accelerate our transformer model's inference, where previously computed transformer key and value states are reused, enabling incremental decoding without recomputing the full sequence. Practically, it takes our model roughly 0.0278 ms to predict a visual trace or an action on an NVIDIA GeForce RTX 5090 GPU. For the reasoning dropout variant of our method, the visual trace generation step is skipped, and a zero vector is used in place of the reasoning trace to condition action prediction.

V. EXPERIMENTS

Here, we detail our experimental setups and results in both simulation and real-world settings to evaluate our ICLR.

A. Models

As mentioned in Section IV-C, our training mechanism allows for two model variants at test time, one complete model and one that does not predict visual traces for the target trajectory (while still conditioning on prompt demos augmented with visual traces). We include both of them in our experiments and denote them as **Ours** and **Ours Dropout**. Additionally, we employ ICRT [5], a state-of-the-art in-context imitation learning framework, as a baseline to compare with our models. As described earlier, our approach builds directly upon the ICRT architecture. Comparing against ICRT allows us to isolate and evaluate the effect of integrating visual reasoning into the in-context imitation learning framework. Although we do not compare our method with certain prior approaches due to differences in experimental settings, for example, Instant Policy [10] relies on two external depth cameras, while KAT [9] requires a calibrated wrist-mounted camera, we believe that the principle of visual reasoning proposed is broadly applicable and can be adapted to other robotic setups. To further evaluate the importance of including visual reasoning traces in prompt demonstrations, we implement another target-only (TO) reasoning approach in which visual reasoning traces are omitted from the prompt demonstrations, while the model is still trained to generate reasoning traces before predicting actions for the target trajectories. We apply the same reasoning dropout training strategy used in our method to this method. As a result, we obtain two versions of this approach for inference, denoted as **TO** and **TO Dropout** in our experimental results. To ensure fair comparisons, any components shared between models are kept identical. We train all models on 2 NVIDIA A6000 Ada GPUs for both simulation and real-world experiments.

B. Simulation Experiments

Setup. We first perform our experiments in LIBERO [40], a widely-used simulation benchmark for robot learning, with a Franka Panda robot arm. In particular, we use the two LIBERO-Object and LIBERO-90 task suites. There is no standard setting for in-context imitation learning on LIBERO, so we repurpose it for our experiments. More specifically, for LIBERO-Object, which has 10 tasks, we randomly select 7 tasks for training, and reserve the remaining 3 as unseen tasks for testing. In LIBERO-90, there are 90 tasks that span 3 different environments (i.e., kitchen scenes, living room scenes, and study scenes). We apply a train/test ratio of 75/15, ensuring that both the training and testing sets contain tasks from all three environments. In particular, we randomly select 6 testing tasks for kitchen scenes, 5 for living room scenes, and 4 for study scenes. Each task in LIBERO comes with 50 expert demos, all of which are used for training. Following [41], we apply a preprocessing step to filter out no-op actions and unsuccessful demonstrations. We inherit training hyperparameters from [5] for all models. During testing, for each unseen task, we use three different prompt episodes (expert demonstrations of that task) and evaluate the model under 50 distinct task initializations with varying object configurations. This results in 150 rollouts per task per model. We report the overall average success rate across all tasks for LIBERO-Object and the average success rate for each environment for LIBERO-90. We also report the overall success rate aggregated across all four settings of LIBERO.

Results. The results are shown in Table I, indicating that our models (both the complete and dropout variants) significantly outperform other baselines. In particular, our dropout model consistently obtains the highest success rates

Method	Poking							Pick-and-Place						
	Hippo	Dumpling	Jaguar	Monkey	Lion	Potato	Avg.	Dumpling to Red box	Zebra to Blue bowl	Tomato to Grey bowl	Monkey to Red box	Lion to Blue bowl	Potato to Grey bowl	Avg.
ICRT [5]	20	40	80	60	40	50	48.33	15	35	30	15	25	5	22.50
TO Dropout	0	60	40	40	60	20	40.00	55	50	30	5	40	10	31.67
TO	30	60	80	30	40	70	51.67	65	45	45	55	20	15	40.83
Ours Dropout	70	60	60	60	70	70	65.00	55	55	50	45	30	45	46.67
Ours	60	80	80	70	70	70	**71.67**	65	70	65	50	45	65	**60.00**

TABLE II: **Real-world task success rates (%).** Every model has 10 rollouts for each task. See Section V-C for our detailed evaluation methodology.

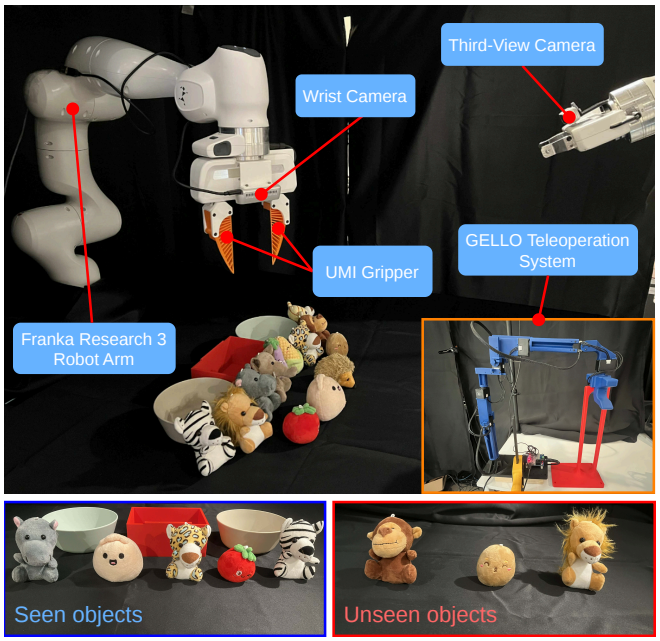

Fig. 3: **Real robot setting.** We use a Franka Research 3 robot arm equipped with a UMI gripper. Visual observations are captured by two RealSense cameras. Teleoperation for data collection and test-time prompt demonstration recording is performed using a GELLO system. Testing objects appearing in training episodes are shown in the bottom-left box, while completely unseen testing objects are shown in the bottom-right box.

across all settings, while the complete ICLR model achieves the second-best scores on three out of four settings and on the overall average success rate. The target-only models rank third and fourth, while ICRT performs the worst overall. These results demonstrate the effectiveness of incorporating explicit reasoning into in-context imitation learning and underscore the importance of including visual reasoning traces in prompt demos, which enable the model to learn to reason and, in turn, facilitate better action prediction in unseen tasks.

C. Real Robot Experiments

Setup. In the real world, we consider a tabletop manipulation setup using a Franka Research 3 robot arm with a UMI gripper [42]. We use the UMI gripper because of its compliance and its bright color, which facilitates more reliable detection by Molmo2. We set up two cameras, one RealSense D435if wrist-mounted camera and another RealSense D415 camera as the fixed third-view camera. To collect training data and record prompt trajectories at test time, we employ a GELLO teleoperation system [43]. Our real-world setting is depicted in Figure 3. For training, we collect 825 demonstrations for 5 poking and 10 pick-and-place tasks across many objects. In each training demo, we ensure that multiple tasks are feasible from the same initial configuration. This prevents shortcut learning and instead encourages the models to perform in-context learning by inferring task intent from the prompt demonstrations. We deploy all models at 30Hz using a computer with a 32GB NVIDIA GeForce RTX 5090 GPU connected to the robot.

Evaluation Protocol. The models are evaluated on 12 unseen tasks that are not included in the training data. Specifically, there are 6 unseen poking tasks, corresponding to the objects *hippo*, *dumpling*, *jaguar*, *monkey*, *lion*, and *potato*. Additionally, there are 6 unseen pick-and-place tasks, which are *dumpling to red box*, *zebra to blue bowl*, *tomato to grey bowl*, *monkey to red box*, *lion to blue bowl*, and *potato to grey bowl*. Note that *hippo*, *dumpling*, *jaguar*, *zebra*, *tomato*, *red box*, *blue bowl*, and *grey bowl* are objects appearing in training (although the associated tasks are unseen), while *monkey*, *lion*, and *potato* are completely unseen objects. Following the evaluation setting of [5], each task has five levels of difficulty, with the number of distractor objects increasing. For each model, we perform 10 rollouts per task, with two rollouts (different configurations) for each difficulty level. For each task, we record 3 prompt demonstrations, i.e., demos with zero, one distractor object, and a distractor receptacle for pick-and-place, or two distractor objects for poking tasks. For every rollout, a one-in-three random prompt demo is used. In a poking task rollout, a model receives 1 point if the object is poked by the robot gripper. In a pick-and-place rollout, the model is scored with 0.5 for a successful pick and 1 if the object is also placed in the correct receptacle. For each rollout, the model has 300 steps for retries. Note that each training episode has from 80 to 200 steps. We report the success rate for each task, as well as the average success rate across the two task types for all models.

Results. The results are shown in Table II, where similar to simulation results, our models largely outperform other baselines on both poking and pick-and-place tasks, reaffirming the effectiveness of our proposed method. Interestingly, while the dropout variants (TO Dropout and Ours Dropout) achieve better performance than the full models in simulation, the complete models obtain higher success rates in real-world settings. We hypothesize that this discrepancy arises from differences in scene configurations between training and testing in simulation and real-world settings. In the LIBERO

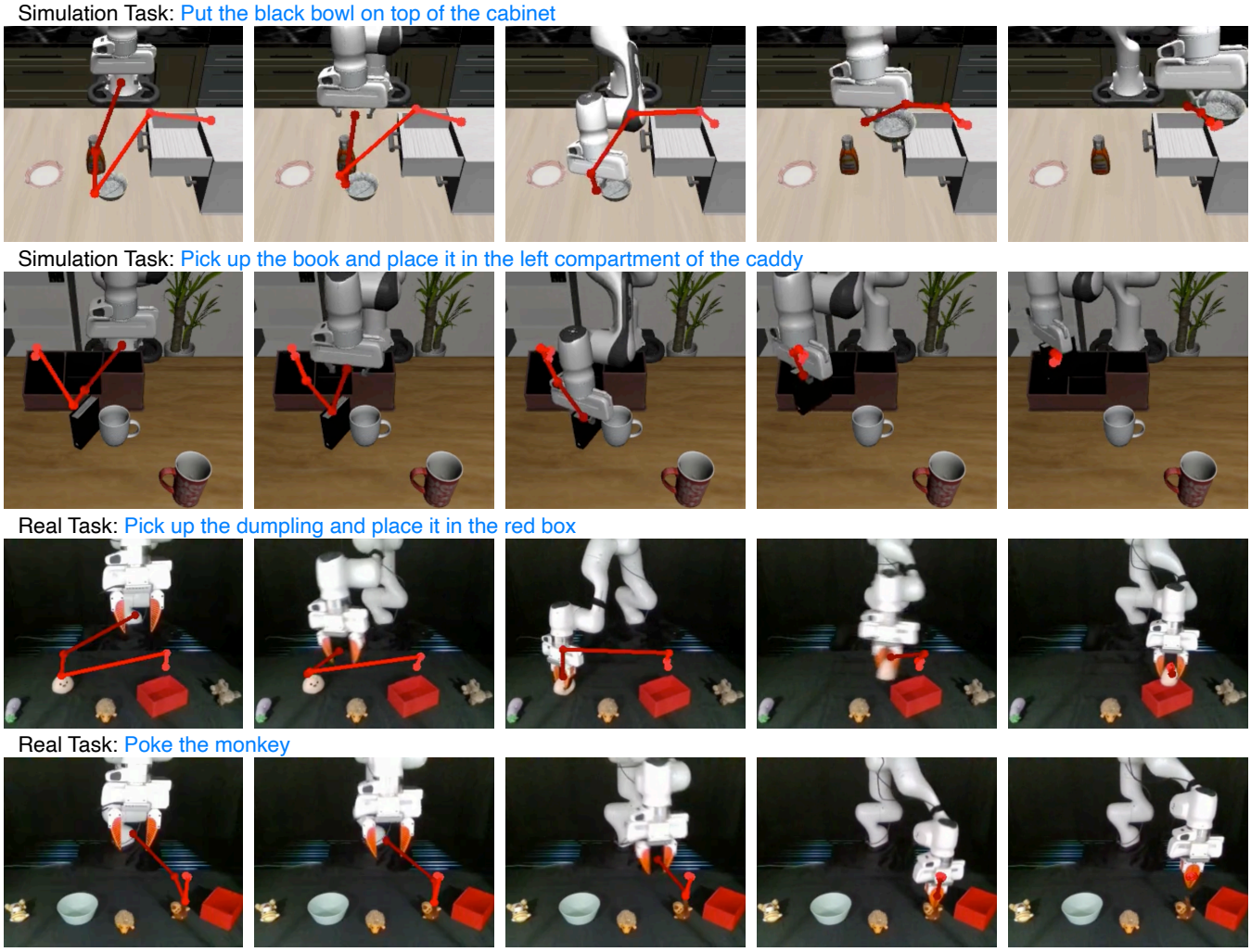

Fig. 4: **Qualitative results.** Rollout examples of our complete ICLR model in simulation (first two rows) and real-world settings (two bottom rows). All presented visual traces are predicted by our model.

simulation, the differences between training and testing scene configurations, particularly object positions, are relatively small. This reduces the need for explicitly generating visual traces and allows the dropout models to "internalize" the reasoning process. We also observe that, in the LIBERO experiments, the actions generated by the dropout models are more stable than those produced by the full models. This may be due to the limited diversity of visual traces in the training data, which restricts the model's ability to reliably learn trace generation. As a result, errors in predicted visual traces can propagate to action prediction, reducing the overall stability of the complete models' generated actions. Together, these factors contribute to the better performance of the dropout models in the LIBERO experiments. In contrast, real-world training data is substantially more diverse and the differences between training and testing configurations are considerably bigger than in LIERBO. Consequently, the models can learn to generate visual traces more effectively and explicit reasoning becomes much more important for guiding action prediction, leading to the superior performance of the full reasoning-enabled models. Refer to Figure 4 for rollout examples of our complete ICLR model. In fact, we find that the reasonably strong performance of our ICLR dropout variant, along with the discrepancy between low- and high-diversity settings, aligns with empirical observations from previous works on embodied reasoning [23], [39].

D. Ablation Studies

In this section, we conduct additional experiments in the real-world setting to further evaluate our method.

Prompt Demonstrations. We investigate the effect of different types of prompt demos on the models' performances. We consider the task of putting the tomato in the grey bowl. Similar to the evaluation protocol described in Section V-C, we collect three prompt demonstrations: one with no distractor, one with one distractor, and another with one distractor receptacle (see Figure 5). We evaluate the models using five prompt configurations: each of the three demos individually, a two-demo configuration where two demos are randomly sampled for every rollout, and a three-demo configuration containing all demonstrations. We conduct 10 rollouts per model for each prompt configuration and report the success rates in Table III. The results indicate that our

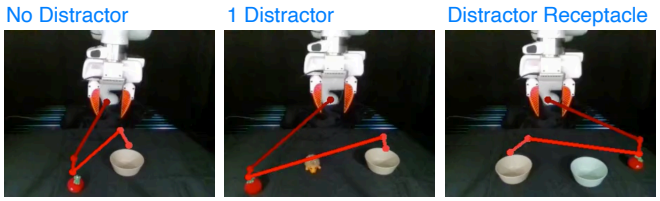

Fig. 5: **Three types of prompt demonstrations.** The task of picking up the tomato and putting it in the grey bowl is selected.

Method	1 Demo			2 Demos	3 Demos
	0 Distr.	1 Distr.	Distr. Receptacle		
ICRT [5]	35	30	40	35	45
TO Dropout	45	25	20	25	45
TO	40	40	50	50	30
Ours Dropout	35	35	40	55	40
Ours	**55**	**65**	**55**	**70**	**65**

TABLE III: **Results of different prompt types (%).** "Distr." stands for distractor. Success rates are calculated over 10 trials for each experiment on the task of picking up the tomato and putting it in the grey bowl.

complete ICLR model consistently achieves the highest success rates, demonstrating its stability under different prompt types. We hypothesize that this is attributable to the diverse range of prompt demonstrations the model encounters during training. We also observe that, across all models, increasing the number of prompt demos does not necessarily lead to a clear improvement in performance, aligning with similar findings in other in-context robot learning works [5], [9], [44]. This contrasts with the typical behavior observed in LLMs and VLMs, where performance generally improves as the number of in-context examples increases [45]. We leave a deeper investigation of this observation to future work.

Failure Analysis. The incorporation of reasoning improves our models' transparency, as the generated visual traces provide interpretable intermediate representations that help us better understand the robot's behavior. Leveraging this interpretability, we systematically analyze the relative proportions of different failure causes in inference. We consider the pick-and-place task of putting the tomato in the grey bowl and the task of poking the hippo. For the pick-and-place task, we categorize failures into three types. The first is visual trace errors, where the generated traces target the wrong object to grasp or terminate at an incorrect receptacle. If the visual traces are correct, failures are further classified as either grasp failures (i.e., failing to grasp the tomato) or placement failures (i.e., failing to place the grasped tomato into the grey bowl). For poking, failures are divided into visual trace errors (traces point to the wrong object) and poking failures (failure to poke the hippo given correct reasoning traces). For each task, we report the percentages over 20 failed rollouts, as shown in Figure 6. In the pick-and-place task, grasp failure is the most significant error type, while in the poking task, mis-poking accounts for the largest proportion. Although visual trace errors contribute 40% and 45% of failures in pick-and-place and poking, respectively, they are not the primary failure mode in either task. This suggests that the proposed integration of visual reasoning is generally effective at capturing task intent, and that overall

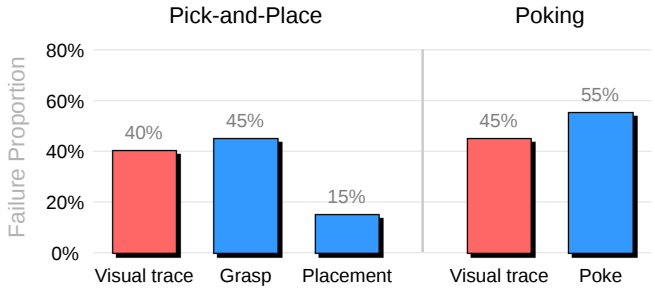

Fig. 6: **Failure analysis.** We report the proportion of visual trace errors (red) versus other failure types (blue), computed over 20 failed rollouts for each of the pick-and-place (tomato to grey bowl) and poking (hippo) tasks.

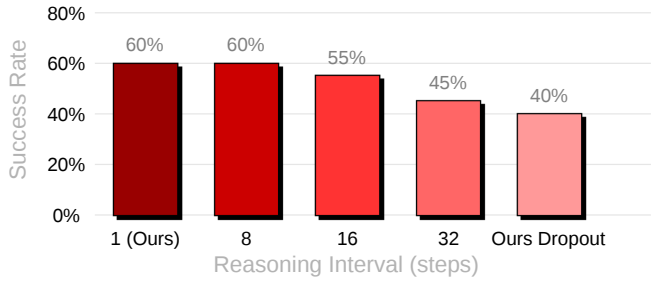

Fig. 7: **Results of different reasoning intervals.** The experiment is conducted on the task of putting the hedgehog into the red box, with 10 rollouts for each model variant.

performance is more often limited by downstream execution challenges rather than incorrect reasoning. Moreover, visual trace errors could potentially be mitigated by improved gripper localization models (e.g., future Molmo versions), which would likely translate into additional performance gains. Overall, the results indicate that improving low-level control robustness may further enhance performance, building upon the strong reasoning capability of our method.

Efficient Reasoning. As the proposed reasoning-dropout training approach allows our model to either predict the reasoning trace or omit it at any time step during inference, we further conduct an experiment to benchmark the performance of different model variants with varying reasoning intervals. In particular, in addition to our complete ICLR model and the dropout model (which completely ignores reasoning during testing), we evaluate three additional reasoning-efficient variants that perform visual reasoning every 8, 16, and 32 time steps. For each model, we perform 10 rollouts on the unseen task of picking up the hedgehog and placing it in the red box. The corresponding success rate results are shown in Figure 7. In general, we observe a decreasing trend in performance as the interval between consecutive reasoning steps increases, with the complete model achieving the highest performance and the dropout model performing the worst. However, the 8-step and 16-step variants achieve results comparable to the full ICLR model, with the 8-step variant performing on par with the complete model while being roughly 8$\times$ faster. The results suggest that while explicit test-time reasoning is important for our method, we can reduce, to some extent, the frequency with which the model performs reasoning while still achieving commendable performance.

VI. DISCUSSION

Despite promising results, our work still has considerable room for improvement. Current experiments involve two types of tasks. Experimenting on a wider range of tasks would further validate the generalizability of our ICLR. While the current visual reasoning representation has demonstrated its usefulness, incorporating other forms of reasoning (e.g., bounding boxes, affordances, or depth information) is promising to benefit in-context imitation learning. Additionally, although in-context imitation learning aims to enable data-efficient robot policies, collecting training data remains time-consuming. A worthwhile yet under-investigated research direction is to develop in-context robot learning methods that can effectively condition on human video demos or demos collected on different robot embodiments. Looking further ahead, we believe that the paradigm of robotic in-context imitation learning remains underexplored. Designing in-context learning methods that can scale to bimanual, dexterous, and long-horizon manipulation is an important open research question that requires advances in both reasoning representations and hierarchical policy learning. We leave these promising research directions for future work.

VII. CONCLUSIONS

We present ICLR, a novel method that incorporates visual reasoning into robotic in-context imitation learning. We conduct a wide range of simulation and real-world robotic experiments, where our proposed method consistently outperforms other methods by a large margin, demonstrating its stronger generalization to unseen tasks and novel object configurations. These results suggest that embodied visual reasoning is a promising direction for improving the robustness and adaptability of robotic in-context learning systems.

VIII. ACKNOWLEDGMENTS

We sincerely thank our friends and colleagues for their support throughout this project. In particular, we are grateful to Kyle Hatch, Nhat Chung, and Sicheng He for their insightful discussions and valuable suggestions. We also thank Sicheng He, Bo-Ruei Huang, and Jason Chen for their assistance in repairing the robot. The USC Physical Superintelligence Lab acknowledges generous supports from Toyota Research Institute, Dolby, Google DeepMind, Capital One, Nvidia, Bosch, NSF, and Qualcomm. Yue Wang is also supported by a Powell Research Award.

REFERENCES

[1] K. Goldberg, "Good old-fashioned engineering can close the 100,000-year "data gap" in robotics," 2025.

[2] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," in *RSS*, 2024.

[3] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *CoRL*, 2022.

[4] L. Y. Chen, C. Xu, K. Dharmarajan, R. Cheng, K. Keutzer, M. Tomizuka, Q. Vuong, and K. Goldberg, "Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning," in *CoRL*, 2024.

[5] M. Fu, H. Huang, G. Datta, L. Y. Chen, W. Panitch, F. Liu, H. Li, and K. Goldberg, "Icrt: In-context imitation learning via next-token prediction," in *ICRA*, 2025.

[6] S. Yang, W. Yu, J. Zeng, J. Lv, K. Ren, C. Lu, D. Lin, and J. Pang, "Novel demonstration generation with gaussian splatting enables robust one-shot manipulation," in *RSS*, 2025.

[7] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang *et al.*, "World action models are zero-shot policies," *arXiv preprint arXiv:2602.15922*, 2026.

[8] F. Lin, K. Arora, J. Mercat, H. Nishimura, P. Shah, C. Xu, M. Zhang, M. Zolotas, M. Angeles, O. Pfannenstiehl *et al.*, "A systematic study of data modalities and strategies for co-training large behavior models for robot manipulation," *arXiv preprint arXiv:2602.01067*, 2026.

[9] N. Di Palo and E. Johns, "Keypoint action tokens enable in-context imitation learning in robotics," in *RSS*, 2024.

[10] V. Vosylius and E. Johns, "Instant policy: In-context imitation learning via graph diffusion," in *ICLR*, 2025.

[11] R. Shah, S. Liu, Q. Wang, Z. Jiang, S. Kumar, M. Seo, R. Martín-Martín, and Y. Zhu, "Mimicdroid: In-context learning for humanoid manipulation from human play videos," in *ICRA*, 2026.

[12] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *NeurIPS*, 2022.

[13] G. Zheng, B. Yang, J. Tang, H.-Y. Zhou, and S. Yang, "Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models," *NeurIPS*, 2023.

[14] Z. Zhang, A. Zhang, M. Li, H. Zhao, G. Karypis, and A. Smola, "Multimodal chain-of-thought reasoning in language models," *TMLR*, 2024.

[15] V. Vosylius and E. Johns, "Few-shot in-context imitation learning via implicit graph alignment," in *CoRL*, 2023.

[16] V. Jain, M. Attarian, N. J. Joshi, A. Wahid, D. Driess, Q. Vuong, P. R. Sanketi, P. Sermanet, S. Welker, C. Chan *et al.*, "Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers," in *RSS*, 2024.

[17] K. Sridhar, S. Dutta, D. Jayaraman, and I. Lee, "Ricl: Adding in-context adaptability to pre-trained vision-language-action models," in *CoRL*, 2025.

[18] C.-P. Huang, Y.-H. Wu, M.-H. Chen, Y.-C. F. Wang, and F.-E. Yang, "Thinkact: Vision-language-action reasoning via reinforced visual latent planning," in *NeurIPS*, 2025.

[19] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine, "Robotic control via embodied chain-of-thought reasoning," in *CoRL*, 2024.

[20] Q. Zhao, Y. Lu, M. J. Kim, Z. Fu, Z. Zhang, Y. Wu, Z. Li, Q. Ma, S. Han, C. Finn *et al.*, "Cot-vla: Visual chain-of-thought reasoning for vision-language-action models," in *CVPR*, 2025.

[21] J. Lee, J. Duan, H. Fang, Y. Deng, S. Liu, B. Li, B. Fang, J. Zhang, Y. R. Wang, S. Lee *et al.*, "Molmoact: Action reasoning models that can reason in space," in *ICRA*, 2026.

[22] Gemini Robotics Team, "Gemini robotics: Bringing ai into the physical world," *arXiv preprint arXiv:2503.20020*, 2025.

[23] Physical Intelligence, "pi0.5: a vision-language-action model with open-world generalization," in *CoRL*, 2025.

[24] Y. Li, Y. Deng, J. Zhang, J. Jang, M. Memmel, C. R. Garrett, F. Ramos, D. Fox, A. Li, A. Gupta *et al.*, "Hamster: Hierarchical action models for open-world robot manipulation," in *ICLR*, 2025.

[25] J. Li, Y. Zhu, Z. Tang, J. Wen, M. Zhu, X. Liu, C. Li, R. Cheng, Y. Peng, Y. Peng *et al.*, "Coa-vla: Improving vision-language-action models via visual-text chain-of-affordance," in *ICCV*, 2025.

[26] J. Zhang, M. Memmel, K. Kim, D. Fox, J. Thomason, F. Ramos, E. Bıyık, A. Gupta, and A. Li, "Peek: Guiding and minimal image representations for zero-shot generalization of robot manipulation policies," in *ICRA*, 2026.

[27] C. Clark, J. Zhang, Z. Ma, J. S. Park, M. Salehi, R. Tripathi, S. Lee, Z. Ren, C. D. Kim, Y. Yang *et al.*, "Molmo2: Open weights and data for vision-language models with video understanding and grounding," *arXiv preprint arXiv:2601.10611*, 2026.

[28] Gemini Robotics Team, "Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer," *arXiv preprint arXiv:2510.03342*, 2025.

[29] Qwen Team, "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025.

[30] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "Sam 2: Segment anything in images and videos," in *ICLR*, 2025.

[31] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa *et al.*, "Dinov3," *arXiv preprint arXiv:2508.10104*, 2025.
[32] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.
[33] M. Minderer, A. Gritsenko, and N. Houlsby, "Scaling open-vocabulary object detection," *NeurIPS*, 2023.
[34] M. Hong, A. Liang, K. Kim, H. Rajaprakash, J. Thomason, E. Bıyık, and J. Zhang, "Hand me the data: Fast robot adaptation via hand path retrieval," in *ICRA*, 2026.
[35] W. Chen, J. Bhatia, C. Glossop, N. Mathihalli, R. Doshi, A. Tang, D. Driess, K. Pertsch, and S. Levine, "Steerable vision-language-action policies for embodied reasoning and hierarchical control," *arXiv preprint arXiv:2602.13193*, 2026.
[36] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.
[37] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *ICML*, 2019.
[38] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *RSS*, 2023.
[39] W. Chen, S. Belkhale, S. Mirchandani, O. Mees, D. Driess, K. Pertsch, and S. Levine, "Training strategies for efficient embodied reasoning," in *CoRL*, 2025.
[40] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *NeurIPS*, 2023.
[41] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong *et al.*, "Openvla: An open-source vision-language-action model," in *CoRL*, 2024.
[42] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," in *RSS*, 2024.
[43] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, "Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators," in *IROS*, 2024.
[44] A. Ruoss, F. Pardo, H. Chan, B. Li, V. Mnih, and T. Genewein, "Lmact: A benchmark for in-context imitation learning with long multimodal demonstrations," in *ICML*, 2025.
[45] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *NeurIPS*, 2020.