In-Context Imitation Learning via Next-Token Prediction

Max (Letian) Fu[1]* **Huang Huang**[1]* **Gaurav Datta**[1]* **Lawrence Yunliang Chen**[1]

Will Panitch[1] **Fangchen Liu**[1] **Hui Li**[2] **Ken Goldberg**[1]

Abstract

We explore how to enable in-context learning capabilities of next-token prediction models for robotics, allowing the model to perform novel tasks by prompting it with human teleop demonstration examples without fine-tuning. We propose In-Context Robot Transformer (ICRT), a causal transformer that performs autoregressive prediction on sensorimotor trajectories, which include images, proprioceptive states, and actions. This approach allows flexible and training-free execution of new tasks at test time, achieved by prompting the model with demonstration trajectories of the new task. Experiments with a Franka Emika robot demonstrate that the ICRT can adapt to new tasks specified by prompts, even in environment configurations that differ from both the prompts and the training data. In a multi-task environment setup, ICRT significantly outperforms current state-of-the-art robot foundation models on generalization to unseen tasks.

1 Introduction

Learning-based single and multi-task robot policies have become increasingly capable [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. This improvement in robot capabilities can largely be attributed to progress in related fields, particularly in vision and language modeling. Inspired by the recent development of large language models (LLMs) and large vision models (LVMs) [11, 12, 13], which formulate natural language processing and vision problems all as next-token-prediction, recent works also have formulated robot learning as next-token-prediction problems and achieved state-of-the-art performance [7, 8, 14, 15]. Concurrently, there has been a surge in collecting large-scale robot datasets [16, 17, 18, 19, 20, 21, 22, 23] and pre-training models on these datasets [24, 25, 26, 27, 15].

Despite being pre-trained on large datasets and showing some generalization ability, it is still challenging to teach these models to perform unseen tasks in different environments without additional training. New human demonstrations via teleoperation or new data collected from hand-crafted motion primitives, as well as another round of model-finetuning, are often needed to complete the new tasks. This process adds complexity to the workflow, making it challenging to apply these methods in real-world environments. Ideally, given one or a few demonstrations, the robot should be able to perform the task *immediately*. In their respective domains, LLMs and LVMs [11, 12, 13] have exhibited a similar ability, named *in-context learning*: a capability allowing the model to rapidly adapt to and recognize the task corresponding to the prompt provided at inference time without additional training.

Is the in-context learning capability of next-token prediction models limited to vision and language domains? In this paper, we introduce In-Context Robot Transformer (ICRT), where we explore how next-token prediction models can be extended to perform real-robot in-context learning. For ICRT, the context is provided as a series of robot trajectories corresponding to a new task. The model learns from this context to perform the task in a different environment configuration without requiring additional training. A robot trajectory is a sequence of image observations, robot proprioceptive states, and actions. This trajectory implicitly encodes task primitives and the objects the robot needs

Workshop on Open-World Agents (OWA-2024)

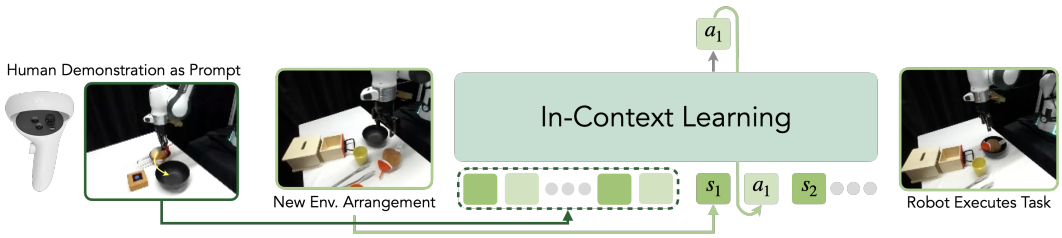

Figure 1: In-Context Robot Transformer (ICRT): A robot foundation model with in-context imitation learning capabilities. ICRT performs next-token prediction on large-scale sensorimotor trajectories. At inference time, it takes raw sensorimotor trajectories of human teleoperation demonstrations as prompts, enabling the model to execute new tasks with real-time continuous control, without requiring fine-tuning.

to interact with. The model extracts this information from the prompt and then executes actions following a similar pattern in its current environment.

Compared to existing few-shot imitation learning approaches, ICRT offers a simple framework that avoids complicated loss functions, prior knowledge, and the need to identify key points or key frames, and operates directly on raw robot trajectories for continuous control. Additionally, unlike existing next-token prediction models for robot learning, ICRT features a long context window, allowing it to train on multiple sensorimotor trajectories from the same task and use one or more sensorimotor trajectories as prompts during inference.

Importantly, we observe that certain properties of the dataset are crucial for enabling in-context learning on real robots. Specifically, datasets that allow multiple tasks to be performed from the same initial observation are particularly beneficial. Unlike existing single-task datasets or many multi-task datasets where each environment has a unique object for robot interaction, these scenarios require the model to rely on the prompt to correctly identify the task and determine the appropriate object for interaction.

We make the following contributions:

1. We introduce ICRT, a robot foundation model capable of in-context learning on real robots. It uses context trajectories and performs unseen tasks without additional training.
2. We provide a new multi-task robot dataset and a training paradigm for fostering multi-task and in-context capability at inference time.
3. Physical experiments on a Franka Emika robot demonstrate that ICRT can learn from the provided context and perform the unseen tasks specified by the prompt at various generalization levels.

2 Related Works

2.1 Multi-Task Imitation Learning for Robotics

Imitation learning is an effective paradigm for equipping robots with various skills. The simplest algorithm in this domain, behavior cloning, has been successful across a wide range of tasks [28, 29, 30]. In recent years, alternative architectures such as energy-based models [31] and diffusion models [1] have also been proposed. Typically, these approaches require training a *separate* model for each task, although multi-task policies can be distilled from task-specific models after training [32].

Recent advancements have shown that using transformers for next-token prediction in sequence modeling has been particularly effective in both language and vision domains, especially for *multi-task learning* [33, 12, 34]. In pursuit of developing generalist agents and multi-task robot policies, robot action planning is framed as a next-token prediction task using transformer-based architectures trained on large multi-task robot datasets [35, 36, 7, 8, 14, 15, 37, 38, 39, 40, 5]. Octo [15] and OpenVLA [14] represent the state-of-the-art among multi-task robotic policies. Octo [15] conditions on both goal images and language instructions, utilizing a transformer architecture with a diffusion head that fuses these inputs with current image observations to predict robot actions. OpenVLA[14] conditions solely on language instructions, fine-tuning a pre-trained vision-language model to predict robot actions based on visual observations and language inputs.

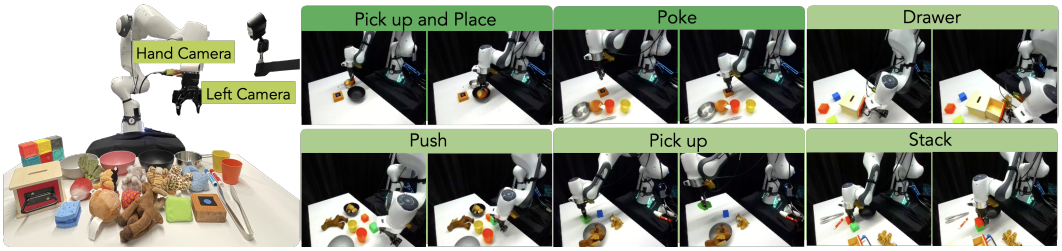

Figure 2: Our physical setup with the Franka Emika robot, the wrist and side camera and the objects used in training and evaluation. We consider 6 primitives for training and choose "pick up and place" and "poke" as the primitives for evaluation (dark green).

2.2 In-Context Learning

Despite training on large datasets, multi-task policies often struggle when faced with new objects, tasks, or environments, frequently requiring fine-tuning. Meta-learning has been shown to increase fine-tuning efficiency for generalization to new tasks [41, 42, 43], which has led to progress in few-shot imitation learning. To simplify the application of learned policies in the real world, recent approaches focus on methods that avoid fine-tuning model parameters for task generalization. Instead, these methods teach the model by providing demonstrations of tasks [44, 45]. Brown et al. [33] refers to this as "in-context learning", distinguishing it from approaches that rely on parameter fine-tuning.

Many in-context learning methods often employ contrastive learning to train context encoders, which identify the most similar training tasks to the test task in the latent space [37, 46]. However, how to effectively integrate these methods within the next-token-prediction framework remains unclear. Valassakis et al. [47] achieved one-shot in-context learning by training a visual servoing network to align the robot's end-effector with the object's relative pose during the demonstration, but this approach requires an additional object segmentation model. Di Palo et al. [44] introduced Keypoint Action Tokens, demonstrating in-context imitation learning using a large language model by representing demonstration trajectories as 3D coordinates with few-shot prompting. Unlike these approaches, ICRT operates without additional perception modules, processing raw image observations directly. Additionally, Vid2Robot [45] developed an encoder-decoder transformer that uses a demonstration video of a human and the current robot state as the prompt to generate robot actions. However, this method requires many auxiliary losses while ICRT uses a simple next-token prediction loss.

In this paper, we focus on enhancing next-token-prediction models to perform real-world in-context imitation learning with robots. ICRT bypasses the need for additional context encoders by directly using robot sensorimotor trajectories from new tasks as prompts for the transformer-based model. ICRT is closely related to the seminal work, One-Shot Imitation Learning [48] and Prompting Decision Transformer [49]. [48] predicts the next action by applying cross-attention between a demonstration sequence on a new task and the current environment's state, while [49] employs a short trajectory prompt to encode task-specific information for guiding policy generation in offline reinforcement learning, using full state information and known reward functions. While both of these methods show their effectiveness in simulation, it is hard to have full-state information and known reward functions for all real-world robot manipulation tasks. To address these challenges, ICRT does not model rewards, utilizes a significantly longer context window, and demonstrates in-context learning capabilities in physical experiments using image observations.

3 Problem Statement

We consider in-context imitation learning under a real-robot manipulation setting. The goal is to train a model with in-context learning capabilities using a multi-task robotic dataset. At test time, the model can handle unseen tasks in novel environment configurations by using a few new human-teleoperated robot demonstrations as prompts. Here, environment configuration refers to the objects present in the scene and their spatial arrangement. Notably, this process is achieved *without any additional training* on the new demonstrations.

We define *motion primitives* as distinct robot actions utilized to accomplish various *tasks*. Each task is characterized by 1) a specific motion primitive and 2) the set of objects the robot interacts with using that primitive. By altering the environment configuration at test time compared to the one

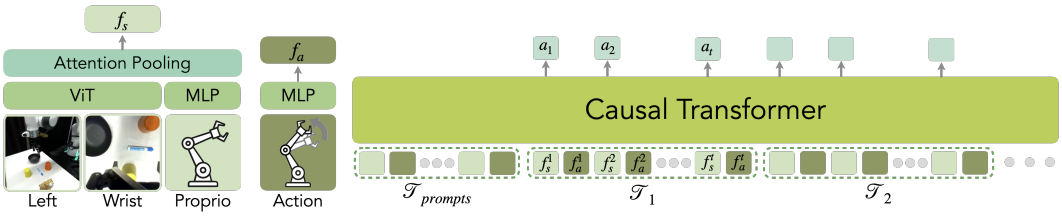

Figure 3: Method Overview: (Left) We encode camera observations with a pre-trained vision transformer. Additionally, we encode proprioception with an MLP. We concatenate the visual latent and the proprioception's latent and use attention pooling to extract a feature f_s as the current state representation. We encode the current action with an MLP to get f_a. (Right) We concatenate multiple trajectories of the same task and randomly sample the first k trajectories as the prompt. A causal transformer autoregressively predicts the next token. We decode the tokens that are at the position of the state features to generate the next $h = 16$ action via an MLP.

in the prompt, we assess the model's ability to select the appropriate motion primitive and identify the correct object for interaction. In this work, we consider new tasks to be tasks involving unseen objects but using motion primitives from the training data (for example, training on picking up a tiger toy and testing on picking up a cube).

We make the following assumptions for ICRT experiments:

1. The model is trained on a diverse multi-task dataset. Each trajectory contains RGB observations from a fixed camera and a wrist-mounted camera, proprioception, and action.
2. The task tested on the robot is within the reachable workspace of the robot.

4 Approach

In this section, we first introduce the data composition to facilitate in-context imitation learning. We then introduce the architecture and training objective for the transformer-based policy to effectively leverage the data.

4.1 Data Formulation

For model training, we consider a dataset $\mathcal{D}$ of visuomotor trajectories $\mathcal{T}$. Each trajectory of length t is a sequence of camera images i_t, proprioceptive robot states s_t, and actions a_t: $\mathcal{T} = (i_1, s_1, a_1, ..., i_t, s_t, a_t)$. We use the absolute end-effector pose as the robot's proprioceptive state and the delta robot end-effector pose between time steps as the action, which consists of delta translation, delta rotation and the continuous gripper action. We assume a known grouping of the trajectories so that the dataset can be partitioned into disjoint sets of tasks $\mathcal{D} = \bigcup_{k=1}^{K} S_k$, with $S_k \cap S_\ell = \emptyset,\ k \neq \ell$, where $\mathcal{S}_k = \{\mathcal{T}_{k_1}, ..., \mathcal{T}_{k_n}\}$. In practice, this grouping can be retrieved from the semantic labels of the dataset. In this work, we utilize the existing large robotic dataset DROID [50] and a multi-task dataset manually collected in our robot setup, which we name ICRT-Multi-Task (ICRT-MT).

DROID [50] is a multi-task dataset that contains 76k real-world demonstrations. We randomly sample 10k demonstrations from DROID after filtering out demonstrations shorter than 30 steps and longer than 450 steps. DROID dataset labels the task through human-specified language instructions, which may be different for the same task. We organized the DROID data by grouping demonstrations based on their language instructions CLIP text embedding cosine similarity. Specifically, we use a threshold of 0.9 for grouping demonstrations. To further facilitate in-context learning, we make sure that each task group contains at least 4 trajectories so that there are sufficient trajectories to serve as prompts for each other. This results in roughly 2k trajectories that we use for pre-training ICRT.

Many trajectories in the DROID dataset are collected in a single-task setup, where only one task is available per scene (e.g., only one object is presented). In this case, the model can learn a shortcut by focusing only on the current observation, ignoring the prompt. Therefore, multi-task data is essential for the model to effectively learn from the prompt. We manually collected a multi-task dataset, ICRT-Multi-Task (ICRT-MT), using the DROID setup (Figure 2), which includes 1,098 trajectories across 29 tasks and 6 primitives: picking, pick-and-place, stacking, pushing, poking, and opening/closing drawers. Objects and primitive examples are shown in Figure 2. In ICRT-MT, each

environment contains at least two possible tasks, requiring the model to distinguish and learn the motion from the prompt.

During the training, for each trajectory, we independently apply vision augmentation on the image observations by augmenting the brightness and contrast. We additionally apply random crops and scaling to the side camera observation. We also apply proprioception noise sampled from a normal Gaussian distribution $\mathcal{N}(0, 0.005)$. For each epoch, we randomly shuffle the order of trajectories from each task and concatenate them to form the training sequence. For each batch, we sample a subsequence of length $L = 512$ as the input to the model, where L is the sequence length defined as the number of observation, state, and action tuples. In practice, 512 steps usually contain up to 5 trajectories from the same task. We randomly select the first k trajectories and label them as the prompt within the sequence. At least one complete trajectory is included in the prompt. This data grouping aims to capture inter-trajectory patterns, encouraging the model to generate action conditioned on the prompt trajectories. This approach differs from traditional behavior cloning methods, which typically use short input sequences that focus on modeling intra-trajectory behaviors.

4.2 Model Architecture

We construct the ICRT model with three parts: a pre-trained vision encoder, a series of projectors for each input modality, and a causal transformer backbone (Figure 3).

Vision Encoder The model processes multi-view image observations through a pre-trained vision transformer. However, most visual pre-trained networks are trained on ImageNet or human videos [27, 51, 52, 24], which exhibit a significant domain gap when compared to typical images from robot datasets, where the images frequently include robots or grippers. To minimize the domain gap, we pre-train a vision transformer [53] (ViT-Base) on an equal mix of ImageNet [54] and Open X-Embodiment [40] data, using CrossMAE as the pre-training method [55]. During the training of the ICRT model, we freeze the vision encoder for efficiency. The vision encoder outputs the entire feature map for each of the cameras and is then fed into the proprioception projector (Figure 3 left).

Modality-Specific Projectors To project image observations, the robot's proprioceptive state, and actions into a shared latent space for sequence modeling, we design modality-specific projectors. At each timestep, the model takes as input a token representing either an observation or an action. To produce a single state token that captures fine-grained visual information and the proprioceptive state of the robot, we use attention pooling [56] between all visual tokens from a single camera's observation and a proprioception embedding produced by a multi-layer perceptron (MLP). The resulting embeddings for each camera are concatenated to produce a single state token f_s^t of dimension equal to the transformer latent dimension. Similar to proprioception, the action is embedded with an MLP into an action token f_a^t. This process produces a sequence of state and action tokens that are passed into the transformer.

Transformer Model The encoded sequence of state and actions is passed into a Transformer model [57], following the design of Llama2 [12]. The transformer takes as input the sequence of state and action features $(f_s^1, f_a^1, \cdots, f_s^t, f_a^t)$ that are produced by the modality-specific projectors. We add MLP decoders to produce state and action outputs from the last layer of the transformer at the appropriate positions. We denote the transformer with the decoder heads as g_θ. Therefore, the desired outputs are the shifted sequence of proprioceptive states and actions $(a^1, s^2, a^2, \cdots, a^t, s^{t+1})$. This naturally forms a next token prediction problem, as $g_\theta(f_s^1)$ predicts a^1 and $g_\theta(f_s^1, f_a^1, \cdots, f_s^n)$ predicts a^{n+1}. In practice, we find it beneficial to predict the next h actions at each time step, and use temporal ensembling [2] to execute the final action.

Inspired by Octo [15] and vision transformers [53], we consider a randomly initialized Llama2 model of 12 layers with a latent dimension of 768, which we name *Llama2-Base*. In addition, multiple works have shown that multimodal inputs can be aligned to large-language models [34, 58, 59, 8, 60]. Multi-modal language model, Palm-E [10] has shown success in enhancing generalization when being directly incorporated into robotic control [8]. Therefore, we also investigate the effectiveness of using a large-language model for in-context robot learning by initializing the transformer with a pre-trained Llama2-7B. Due to the large domain gap between natural language and robot trajectories, a frozen language model may not be sufficient. Therefore, similar to prior work in multimodal alignment, we fine-tune the language model with LoRA [61], with a rank of 32. Due to compute resource limitations, we are unable to fully fine-tune the model.

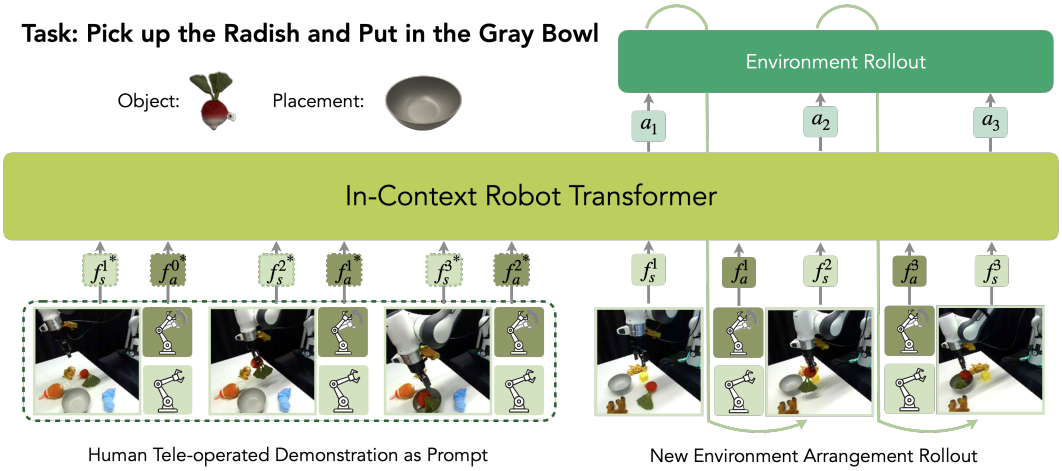

Figure 4: An example inference pipeline of ICRT involves picking up a radish and placing it in a gray bowl. A human teleoperated demonstration, including image observations, proprioception, and actions, is provided as the prompt. ICRT then uses the prompt and the current observation in a new environment to complete the task.

Loss Function To provide more supervision signals so that the model can better respond to the trajectory "prompt" we provide at test time, we reference works in training multi-turn conversation chatbots [62, 34], where they only compute loss on the response generated by the chatbot, instead of the prompt. Recall that in Section 4.1, we randomly sample the subsequence of the concatenated trajectories as the prompt trajectory. Analogously, we only compute action prediction with L1-loss for the actions after the prompt trajectories.

Inference The simplicity of the next-token prediction objective makes inferencing with ICRT straightforward at test time. As shown in Figure. 4, we provide one or more human-teleoperated demonstrations in the form of robot sensorimotor trajectories (formatted identically to the training data), along with the current image observations and the robot's proprioceptive state as inputs. The model then predicts the next action, which is executed by the robot. After each action, the policy receives updated image observations and proprioceptive state, allowing it to iteratively predict and execute subsequent actions.

A key advantage of this framework is its use of the transformer's sequential processing capability. Instead of reprocessing the entire sequence history for each model evaluation, as seen in previous works [15, 14, 7, 8], the model employs a key-value (KV) caching mechanism, as discussed in [12]. This mechanism stores previous outputs, allowing the model to compute only the outputs for the new token. This approach significantly reduces computational overhead, lowering the complexity from quadratic to linear relative to the sequence length. Empirically, ICRT can inference at 39.6 Hz, allowing it to perform real-time close-loop control.

5 Experiments

In this section, we design an experimental setup to evaluate the in-context learning capabilities of the proposed models and compare them against several baselines. Instead of focusing on the difficulty of learning a specific task primitive, we design the experiments to assess the policy's ability to accomplish novel tasks based on the provided prompt trajectories.

Experiment Design We consider two action primitives: a *pick-and-place* primitive and a *poking* primitive. For each action primitive, we design *six unseen tasks* (as defined in Section 3), with three tasks evaluating *in-domain* object generalization (selected from *yellow cube, red cube, black cube, pink bowl, and blue bear* and three objects *unseen* during training (selected from *radish, blue sponge, grey dog, and black dog*).

Each task has five difficulty tiers. In the pick-and-place task, the model must identify the correct object to grasp and where to place it in a multi-object or multi-placement scenario. The tiers include: 1) no distractors, 2) one distractor object, 3) two distractors, 4) three distractors, and 5) one distractor placement position. In the poking task, the robot closes the gripper, pokes the object, lifts the end-effector, and opens the gripper, with tiers involving 0-4 distractor objects. The pick-and-place

	Pick and Place	Poke	Average
Goal Condition	33.3 (±6.5)	6.7 (±4.6)	20.0 (±4.3)
Octo [15]	5.0 (±2.7)	13.3 (±6.2)	9.2 (±3.5)
OpenVLA [14]	11.7 (±4.6)	3.3 (±3.3)	7.5 (±2.9)
ICRT-Llama2	43.3 (±7.9)	73.3 (±8.2)	58.3 (±6.0)
ICRT (DROID)	0.0 (±0.0)	0.0 (±0.0)	0.0 (±0.0)
ICRT (MT)	**76.7 (±7.1)**	70.0 (±8.5)	73.3 (±5.5)
ICRT +Prompt Loss	21.7 (±6.2)	23.3 (±7.9)	22.5 (±5.0)
ICRT	65.0 (±7.3)	**93.3 (±4.6)**	**79.2 (±4.6)**

Table 1: Results. We evaluated each task primitive using *six tasks* not seen during training, conducting five trials per task for 30 trials per primitive and 60 trials overall to calculate average performance. ICRT outperforms two state-of-the-art goal or language-conditioned robot foundation models in both pick-and-place and poking tasks. We additionally ablate three key design choices: fine-tuning ICRT using a language model, training solely on the relevant dataset or the fine-tuning dataset, and the impact of including prompt loss in the training process. For each model, we report the mean success rate for each task, the overall success rate, and the corresponding standard error in parentheses.

task is scored with 0.5 for a correct pick and 1 for a successful placement. In the poking task, failure is marked if the wrong object is poked. The model has 25 seconds (375 steps) for retries. Each difficulty level is attempted once, and we report the average success rate per task, along with the average success rate and standard deviation across the six tasks for each action primitive.

Models The default **ICRT** is a randomly initialized Llama2-Base model pretrained on DROID and fully fine-tuned on ICRT-MT. We evaluate the impact of model initialization and training datasets by introducing the following three variants: 1) **ICRT-Llama2**, a pre-trained Llama2-7B language model fine-tuned on ICRT-MT with LoRA; 2) **ICRT (DROID)**, a randomly initialized Llama2-Base model trained only on the DROID dataset; and 3) **ICRT (MT)**, a randomly initialized Llama2-Base model trained only on the ICRT-MT dataset.

We consider 3 baseline models. We train a goal-conditioned policy, where the goal observations are always prepended to the sequence, and each sequence is from one trajectory. This resembles the normal goal-conditioned imitation learning setup. Additionally, we finetune Octo [15], the state-of-the-art goal-image and language-conditioned policy, and OpenVLA [14], the state-of-the-art language conditioned multi-task imitation learning policy. Octo is fine-tuned using their official fine-tuning recipe. We incorporate action chunking into OpenVLA by asking it to predict the next 16 actions, which performs better than vanilla OpenVLA which predicts only the next step. Both of these methods are representative of robot policies that use next-token prediction objectives.

Prompt Generation For each task, we collect 3 demonstrations (with zero, one distractor object, a distractor placement for pick-and-place, or two distractor objects for poking) as the prompt in total before running the experiment. Please refer to the appendix for a visual example. During testing, a random demonstration is drawn as a prompt to assess the model's ability to generalize to different prompts. It's important to note that the environment setup during policy rollout *differs* from the prompts' setup, ensuring that the evaluation measures the model's understanding of task-relevant information from the prompt, rather than simply copying actions from it.

Results We present the results in Table 1. For the pick-and-place primitive, we observe that the goal-conditioned policy generally succeeds in identifying the correct object to grasp when no distractor objects are present. However, its performance degrades significantly as the number of distractors increases. When the goal image only specifies the task but not the specific way to achieve it in the current environment, goal-conditioned policies often fail to execute the task effectively.

Octo struggles to determine which object to interact with and where to place it, underscoring the difficulty of our multi-task setup. While OpenVLA often approaches the correct object, it frequently fails to grasp it or performs the wrong task (e.g., grasping instead of poking). This suggests OpenVLA may need more than 50 demonstrations per task for improved performance, and that language conditioning alone may not be enough for generalization to new tasks.

The results indicate that ICRT outperforms the goal-conditioned policy in selecting the correct object and placement location. In the poking task, where goal-conditioned policies struggle due to the goal's similarity to the start position, ICRT, after prompt conditioning, correctly identifies the task and consistently targets the correct object while ignoring distractors. However, ICRT occasionally fails,

either by missing the grasp, picking the wrong object, or misplacing items. The model particularly struggles when a distractor shares the same color but differs in shape, suggesting the need for further vision encoder fine-tuning, as also noted by OpenVLA [14].

6 Ablations

In this section, we provide additional experiments presented Table 1 that ablate on a few core design choices. We provide additional ablation studies in the appendix.

Model Initialization We conducted ablation studies to examine the impact of using a pretrained Llama2 on language data and fine-tune it for robot sensorimotor sequence modeling. The results, presented in Table 1, show that although ICRT-Llama2-7B achieves a lower training loss, its performance is worse compared to its smaller counterparts. This discrepancy may be attributed to a lower inference frequency of ICRT-Llama2 (10.7 Hz vs 39.6 Hz). Future work can focus on optimizing the inference speed of ICRT-Llama2 to improve performance.

Training Dataset We find that training on the DROID subset (see Section 4.1) is insufficient for completing any of the test tasks; the policy (ICRT (DROID)) shows no progress across all tasks. This suggests that although the DROID subset may offer greater visual diversity, the unique structure of ICRT-MT—where multiple tasks are performed from the same initial observation—is particularly beneficial in developing the in-context learning capabilities of a next-token prediction robot model.

ICRT (MT) shows similar performance to ICRT that is pre-trained on DROID, especially for the pick-up and place primitive, even surpassing ICRT on the *put radish in grey bowl* task. However, ICRT (MT) does not perform as well on the poking primitive. The results suggest that it may be beneficial to pre-train the autoregressive model on a large dataset, as a diverse dataset may help the transformer to perform better alignment between visual features and control.

No Prompt Loss Following the design of many multi-turn conversation large language models or vision language model fine-tuning works [34, 62, 63, 64], we do not calculate the loss for the predicted action in the prompt trajectories but only do so on the predictions after the prompt trajectories. We mark the model that calculates loss on the prompt as **ICRT +Prompt Loss** and the default model as **ICRT**. The results are shown in Table 1. We find that only predicting the trajectories after the designated prompt trajectories can significantly improve the model's performance. We hypothesize that when there is a loss on the prompt trajectories, the model is forced to do unconditional generation based on current observations for those prompts. This may cause the model to stop paying attention to the prompt, especially when there are multiple possible tasks available.

7 Limitations and Conclusion

While results show that ICRT learns from prompt trajectories and generalizes to unseen objects, tasks, and similar action primitives, it remains unclear how to generalize to entirely new primitives. Future work should explore how scaling model capacity and datasets can aid in primitive-level generalization. Additionally, ICRT assumes a fixed robot morphology with a set impedance controller; further research could focus on enabling transfer across different morphologies by learning a unified policy. ICRT-Llama2's low inference frequency likely affects its performance, and speeding it up at inference time is a goal for future improvements.

In summary, we present ICRT, where we study in-context, multi-task imitation learning on a real robot. We achieve this by training a causal transformer model on sequences of robot trajectories, where trajectories from the same task are combined to provide context for task execution. Additionally, we introduce a multi-task dataset to facilitate this in-context learning approach. Our experiments show that by using robot sensorimotor trajectories as context, the model can generalize learned motion primitives to unseen objects and novel environment configurations, particularly in scenarios where multiple tasks are present.

References

[1] Cheng Chi et al. "Diffusion Policy: Visuomotor Policy Learning via Action Diffusion". In: *arXiv preprint arXiv:2303.04137* (2023).

[2] Tony Z Zhao et al. "Learning fine-grained bimanual manipulation with low-cost hardware". In: *arXiv preprint arXiv:2304.13705* (2023).

[3] Corey Lynch et al. "Interactive language: Talking to robots in real time". In: *IEEE Robotics and Automation Letters* (2023).

[4] Scott Reed et al. "A generalist agent". In: *arXiv:2205.06175* (2022).

[5] Dhruv Shah et al. "ViNT: A Foundation Model for Visual Navigation". In: *7th Annual Conference on Robot Learning (CoRL)*. 2023.

[6] Homanga Bharadhwaj et al. "RoboAgent: Towards Sample Efficient Robot Manipulation with Semantic Augmentations and Action Chunking". In: *arxiv* (2023).

[7] Anthony Brohan et al. "Rt-1: Robotics transformer for real-world control at scale". In: *arXiv:2212.06817* (2022).

[8] Anthony Brohan et al. "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control". In: *arXiv preprint arXiv:2307.15818* (2023).

[9] Xi Chen et al. *PaLI-X: On Scaling up a Multilingual Vision and Language Model*. 2023. arXiv: 2305.18565 [cs.CV].

[10] Danny Driess et al. "Palm-e: An embodied multimodal language model". In: *arXiv:2303.03378* (2023).

[11] Josh Achiam et al. "Gpt-4 technical report". In: *arXiv preprint arXiv:2303.08774* (2023).

[12] Hugo Touvron et al. "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288* (2023).

[13] Yutong Bai et al. "Sequential modeling enables scalable learning for large vision models". In: *arXiv preprint arXiv:2312.00785* (2023).

[14] Moo Jin Kim et al. *OpenVLA: An Open-Source Vision-Language-Action Model*. 2024. arXiv: 2406.09246 [cs.RO].

[15] Octo Model Team et al. "Octo: An Open-Source Generalist Robot Policy". In: *Proceedings of Robotics: Science and Systems*. Delft, Netherlands, 2024.

[16] Amaury Depierre, Emmanuel Dellandréa, and Liming Chen. "Jacquard: A large scale dataset for robotic grasp detection". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 3511–3516.

[17] Dmitry Kalashnikov et al. "Scalable deep reinforcement learning for vision-based robotic manipulation". In: *CoRL*. 2018.

[18] Sergey Levine et al. "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection". In: *IJRR* (2018).

[19] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. "ACRONYM: A Large-Scale Grasp Dataset Based on Simulation". In: *2021 IEEE Int. Conf. on Robotics and Automation, ICRA*. 2020.

[20] Nur Muhammad Mahi Shafiullah et al. *On Bringing Robots Home*. 2023. arXiv: 2311.16098 [cs.RO].

[21] Hao-Shu Fang et al. "RH20T: A Robotic Dataset for Learning Diverse Skills in One-Shot". In: *RSS 2023 Workshop on Learning for Task and Motion Planning*. 2023.

[22] Frederik Ebert et al. "Bridge data: Boosting generalization of robotic skills with cross-domain datasets". In: *arXiv:2109.13396* (2021).

[23] Homer Walke et al. *BridgeData V2: A Dataset for Robot Learning at Scale*. 2023. arXiv: 2308.12952 [cs.RO].

[24] Suraj Nair et al. "R3m: A universal visual representation for robot manipulation". In: *arXiv:2203.12601* (2022).

[25] Tete Xiao et al. "Masked visual pre-training for motor control". In: *arXiv:2203.06173* (2022).

[26] Yecheng Jason Ma et al. "Vip: Towards universal visual reward and representation via value-implicit pre-training". In: *arXiv preprint arXiv:2210.00030* (2022).

[27] Ilija Radosavovic et al. "Real-world robot learning with masked visual pre-training". In: *arXiv:2210.03109* (2022).

[28] Dean A. Pomerleau. "ALVINN: An Autonomous Land Vehicle in a Neural Network". In: *NeurIPS*. Ed. by D. Touretzky. Vol. 1. Morgan-Kaufmann, 1988.

[29] Brenna D Argall et al. "A survey of robot learning from demonstration". In: *Robotics and autonomous systems* 57.5 (2009), pp. 469–483.

[30] Sergey Levine et al. "End-to-end training of deep visuomotor policies". In: *JMLR* (2016).

[31] Peter R. Florence et al. "Implicit Behavioral Cloning". In: *CoRL*. 2021.

[32] Huy Ha, Pete Florence, and Shuran Song. "Scaling up and distilling down: Language-guided robot skill acquisition". In: *Conference on Robot Learning*. PMLR. 2023, pp. 3766–3777.

[33] Tom B Brown et al. "Language models are few-shot learners". In: *NeurIPS* (2020).

[34] Haotian Liu et al. "Visual Instruction Tuning". In: *NeurIPS*. 2023.

[35] Ilija Radosavovic et al. "Robot Learning with Sensorimotor Pre-training". In: *arXiv:2306.10007* (2023).

[36] Ilija Radosavovic et al. *Humanoid Locomotion as Next Token Prediction*. 2024. arXiv: 2402.19469 [cs.RO].

[37] Eric Jang et al. "Bc-z: Zero-shot task generalization with robotic imitation learning". In: *Conference on Robot Learning*. 2022.

[38] Yunfan Jiang et al. "VIMA: General robot manipulation with multimodal prompts". In: *International Conference on Machine Learning (ICML)* (2023).

[39] Scott Reed et al. "A generalist agent". In: *arXiv preprint arXiv:2205.06175* (2022).

[40] Embodiment Collaboration et al. *Open X-Embodiment: Robotic Learning Datasets and RT-X Models*. 2024. arXiv: 2310.08864 [cs.RO].

[41] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International conference on machine learning*. PMLR. 2017, pp. 1126–1135.

[42] Chelsea Finn et al. "One-shot visual imitation learning via meta-learning". In: *Conference on robot learning*. PMLR. 2017, pp. 357–368.

[43] Mengdi Xu et al. "Hyper-decision transformer for efficient online policy adaptation". In: *arXiv preprint arXiv:2304.08487* (2023).

[44] Norman Di Palo and Edward Johns. "Keypoint Action Tokens Enable In-Context Imitation Learning in Robotics". In: *arXiv preprint arXiv:2403.19578* (2024).

[45] Vidhi Jain et al. "Vid2Robot: End-to-end Video-conditioned Policy Learning with Cross-Attention Transformers". In: *arXiv preprint arXiv:2403.12943* (2024).

[46] Zhao Mandi et al. *Towards More Generalizable One-shot Visual Imitation Learning*. 2022. arXiv: 2110.13423 [cs.RO].

[47] Eugene Valassakis et al. "Demonstrate once, imitate immediately (dome): Learning visual servoing for one-shot imitation learning". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 8614–8621.

[48] Yan Duan et al. "One-shot imitation learning". In: *Advances in neural information processing systems* 30 (2017).

[49] Mengdi Xu et al. "Prompting decision transformer for few-shot policy generalization". In: *international conference on machine learning*. PMLR. 2022, pp. 24631–24645.

[50] Alexander Khazatsky et al. *DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset*. 2024. arXiv: 2403.12945 [cs.RO].

[51] Arjun Majumdar et al. "Where are we in the search for an Artificial Visual Cortex for Embodied Intelligence?" In: *arXiv preprint arXiv:2303.18240* (2023). arXiv: 2303.18240 [cs.CV].

[52] Shizhe Chen et al. "SUGAR: Pre-training 3D Visual Representations for Robotics". In: *arXiv preprint arXiv:2404.01491* (2024).

[53] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *ICLR*. 2020.

[54] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *CVPR*. 2009.

[55] Letian Fu et al. "Rethinking Patch Dependence for Masked Autoencoders". In: *arXiv preprint arXiv:2401.14391* (2024).

[56] Juho Lee et al. "Set transformer: A framework for attention-based permutation-invariant neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 3744–3753.

[57] Ashish Vaswani et al. "Attention is all you need". In: *NeurIPS*. 2017.

[58] Jiaming Han et al. *ImageBind-LLM: Multi-modality Instruction Tuning*. 2023. arXiv: 2309.03905 [cs.MM].

[59] Letian Fu et al. "A Touch, Vision, and Language Dataset for Multimodal Alignment". In: *arXiv preprint arXiv:2402.13232* (2024).

[60] Suvir Mirchandani et al. *Large Language Models as General Pattern Machines*. 2023. arXiv: 2307.04721 [cs.AI].

[61] Edward J Hu et al. "Lora: Low-rank adaptation of large language models". In: *arXiv preprint arXiv:2106.09685* (2021).

[62] Wei-Lin Chiang et al. "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality". In: *See https://vicuna. lmsys. org (accessed 14 April 2023)* 2.3 (2023), p. 6.

[63] Haotian Liu et al. *Improved Baselines with Visual Instruction Tuning*. 2023.

[64] Wenliang Dai et al. "Instructblip: Towards general-purpose vision-language models with instruction tuning". In: *Advances in Neural Information Processing Systems* 36 (2024).

8 Supplementary Material

8.1 Scene Illustrations

We provide illustrations on the prompt trajectories and test scenes for the pick up the black dog and place in the pink bowl task in Figure 5. As mentioned in Section 5, we collected 3 types of prompt trajectories and test ICRT on 5 tiers of scenes that are different from the scenes in the prompt trajectories.

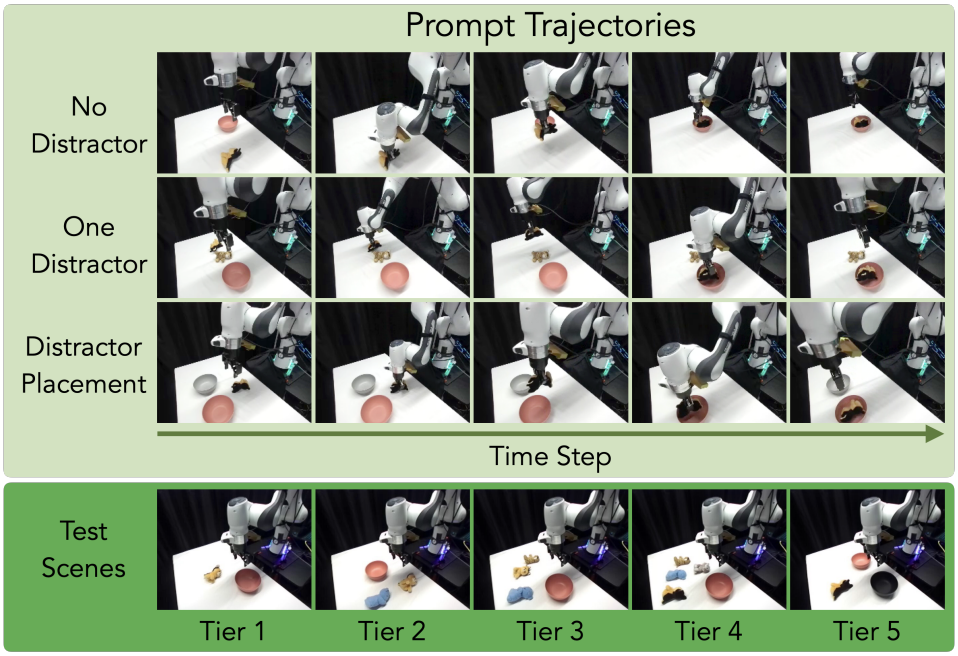

Figure 5: Illustrations of the prompt trajectories (top) and test scenes (bottom) for the pick up the black dog and place in the pink bowl task. Three prompt trajectories of different types are collected. The test scenes are different from all prompt trajectories and 5 tiers of scenes with different number of distractors are considered.

8.2 Ablation Studies

In this section, we provide additional ablation experiments on a few core design choices and different prompting strategies.

8.2.1 Repeatability Experiments

We conduct experiments to evaluate the repeatability of the performance of ICRT. We conduct a pick up the black dog and place in the pink bowl task and a poke blue sponge task for 5 rollouts, where each rollout contains 5 trials as in Section 5, resulting a total of 25 trials. We calculate the average and the standard deviation of the success rate. Results are shown in Table 2. The low std from Table 2 suggests that the ICRT can reliably achieve the task.

Task	Pick and Place Block Dog in Pink Bowl	Poke Blue Sponge
Success Rate Ave. $\pm$ Std.	60% $\pm$ 0.5%	88% $\pm$ 3.2%

Table 2: Repeatability experiments for a pick and place task and a poking task. Each task is conducted by 5 rollouts and each rollout contains 5 trials, resulting in a total of 25 trials.

8.2.2 Prompt Trajectories

We conduct experiments on different prompt types to evaluate the effect of different prompt trajectories on task performance. We consider the task of picking up a black dog and placing in a pink bowl. We

Prompt Type	No Distractor	One Distractor	Distractor Placement	Two Prompts	Three Prompts
Success Rate	60%	80%	70%	80%	80%

Table 3: Experiments on different prompt types on a pick up black dog and place in the pink bowl task. The first three columns are results for a single prompt trajectory of different types, while the last two columns are that for using two and three prompts. Success rates are calculated over 5 trials for each experiment.

have three prompt trajectories of different types: one with no distractors, one with one distractor and one with one distractor placement, as shown in Appendix Figure 5 top. All three prompts trajectories are collected by human teleoprating the robot. The object locations and the placement locations at test time are different from that in all three prompts. As in Section 5, for each prompt type, we conduct the task with 5 trials as shown in Appendix Figure 5 bottom. The average success rates are reported in Table 3. We conduct experiments with one prompt trajectory of different types (the first three columns in Table 3), two prompt trajectories and three prompt trajectories. All prompt types result in similar performance, indicating ICRT is not sensitive to the prompt trajectory types. We hypothesize this is because during the training, ICRT has seen different types and numbers of prompts.

8.2.3 Unseen Primitives

Task	Grasp and Drop the Toy Tiger	Grasp and Drop the Blue Sponge	Put Blue Sponge to Right of Toy Tiger
Success Rate	40%	80%	80%

Table 4: Experiments on three tasks using two unseen primitives. Success rates are calculated over 5 trials for each experiment.

We evaluate the generalization capability of ICRT on primitives that are unseen during the training but resemble the training primitives. We consider two such unseen primitives: grasp and drop an object and put object A to the right of object B. We consider three tasks: grasp and drop a toy tiger, grasp and drop a blue sponge (unseen objects during training) and put the blue sponge to the right of the toy tiger. As in Section 5, we conduct 5 tiers for each task. Experiment results are summarized in Table 4, where ICRT shows decent success rate on all three tasks, suggesting that ICRT can generalize to some unseen primitives that resemble the training primitives.

8.2.4 Co-training

For training ICRT, we opt to separate the training into two stages: a pre-training phase where the model is pre-trained on the DROID dataset [50], and a fine-tuning phase where the model is trained on the ICIL-MT dataset. In this ablation, we experiment with whether these two can be combined into a single stage, where the policy is end-to-end trained with DROID and ICIL-MT. To balance the two datasets, we first calculate the median number of trajectories per task across the two datasets, then for each epoch, sample each task with the median number of trajectories. This allows each task to be equally represented in each epoch. We train the model for the same number of epochs as for ICRT fine-tuning and report the results in Table 5. The results indicate that the model does not converge as quickly in the combined stage and fails to respond to prompts and complete tasks effectively. We hypothesize two reasons for this: firstly, the dataset is heavily biased towards DROID, which contains 200 tasks compared to only 29 tasks in ICIL-MT, making it difficult for the model to learn the tasks as effectively as in the separate stage training. Future works can analyze the data mixture and how to train with large-scale datasets more effectively.

	Pick and Place	Poke	Average
ICRT (Co-train)	13.3 ($\pm$5.8)	0.0 ($\pm$0.0)	6.7 ($\pm$3.0)
ICRT	**65.0 ($\pm$7.3)**	**93.3 ($\pm$4.6)**	**79.2 ($\pm$4.6)**

Table 5: Ablation on co-training with DROID [50]. Training both DROID and ICRT-MT datasets in a single stage leads to worse task performance in both pick-and-place and poking tasks. Same as Table 1, we evaluated each task primitive using six tasks not seen during training, conducting five trials per task for a total of 30 trials per primitive and 60 trials overall to calculate average performance. For each model, we report the mean success rate for each task, the overall success rate, and the corresponding standard error in parentheses.

Pick Object Place Location	Yellow Cube Black Bowl	Yellow Cube Grey Bowl	Blue Bear Pink Bowl	Radish Grey Bowl	Black Dog Pink Bowl	Blue Sponge Silver Pot	Average Success (± Std Err.)
Goal Conditioned	40%	30%	20%	40%	40%	30%	33.3% (±6.5%)
Octo	10%	0%	10%	10%	0%	0%	5.0% (±2.7%)
OpenVLA	0%	0%	0%	50%	20%	0%	11.7% (±4.6%)
ICRT-Llama2	40%	40%	40%	60%	40%	40%	43.3% (±7.9%)
ICRT (DROID)	0%	0%	0%	0%	0%	0%	0.0% (±0.0%)
ICRT (MT)	**90%**	**50%**	**80%**	**90%**	**60%**	**90%**	**76.7% (±7.1%)**
ICRT +Prompt Loss	20%	10%	20%	40%	30%	10%	21.7% (±6.2%)
ICRT (Co-train)	10%	0%	10%	0%	40%	20%	13.3% (±5.8%)
ICRT	60%	**50%**	**80%**	50%	**60%**	**90%**	65.0% (±7.3%)

Table 6: Experimental results for the *pick-and-place* primitive. Here we list the 6 tasks that were evaluated and their corresponding success rate. For each model, we also report the mean success rate and the standard error.

8.3 Detailed results

In this section, we present the per-task performance for the pick-and-place primitive (Table 6) and the poking primitive (Table 7). For each action primitive, we design *six unseen tasks* (as defined in Section 3), with three tasks evaluating *in-domain* object generalization (selected from *yellow cube, red cube, black cube, pink bowl, and blue bear* and three objects *unseen* during training (selected from *radish, blue sponge, grey dog, and black dog*).

Poke Object	Radish	Red Cube	Grey Dog	Black Cube	Pink Bowl	Blue Sponge	Average Success (± Std.)
Goal Conditioned	0%	0%	0%	0%	40%	0%	6.7% (±4.6%)
Octo	20%	0%	60%	0%	0%	0%	13.3% (±6.2%)
OpenVLA	20%	0%	0%	0%	0%	0%	3.3% (±3.3%)
ICRT-Llama2	60%	**100%**	80%	60%	60%	80%	73.3% (±8.2%)
ICRT (DROID)	0%	0%	0%	0%	0%	0%	0.0% (±0.0%)
ICRT (MT)	**100%**	**100%**	40%	60%	60%	60%	70.0% (±8.5%)
ICRT +Prompt Loss	0%	20%	20%	**80%**	0%	20%	23.3% (±7.9%)
ICRT (Co-train)	0%	0%	0%	0%	0%	0%	0.0% (±0.0%)
ICRT	**100%**	**100%**	**80%**	**80%**	**100%**	**100%**	**93.3% (±4.6%)**

Table 7: Experimental results for the *poking* primitive. Here we list the 6 tasks that were evaluated and their corresponding success rate. For each model, we also report the mean success rate and the standard error.

8.4 Hyperparameters

We provide the hyperparameters for both the pre-training and fine-tuning phase in Table 8 and Table 9.

Config	Value
optimizer	AdamW
base learning rate	1e-3
learning rate schedule	cosine decay
batch size	64
weight decay	0.05
optimizer momentum	β_1, β_2 = 0.9, 0.999
warm up epoch	0.5
total epochs	4
proprioception noise	0.005
action noise	0
sequence length	512
brightness augmentation	0.1
contrast augmentation	0.2
num action prediction	16

Table 8: Pre-training Hyperparameters

8.5 Parameterization

Proprioception The proprioception space is parameterized by the absolute end effector translation (x, y, z), a 6DoF rotation vector, and a continuous end-effector gripper state. This results in a 10-dimensional proprioception representation. The 6DoF rotation vector is flattened from the $SO(3)$ rotation's matrix's first two rows.

Action We use delta end effector pose as the action parameterization. At each prediction step,

Config	Value
optimizer	AdamW
base learning rate	5e-4
learning rate schedule	cosine decay
batch size	64
weight decay	0.01
optimizer momentum	β_1, β_2 = 0.9, 0.999
warm up epoch	1.25
total epochs	125
proprioception noise	0.005
action noise	0
sequence length	512
brightness augmentation	0.1
contrast augmentation	0.2
num action prediction	16

Table 9: Finetuning Hyperparameters

the model predicts t actions. Given *absolute* end effector action transforms in $T_1, T_2, \cdots, T_t$ in a trajectory and the current end-effector pose T_{ee}, we define the relative transforms that the model needs to predict as $T_{\text{ee}}^{-1}T_1, T_{\text{ee}}^{-1}T_2, \cdots T_{\text{ee}}^{-1}T_t$. We then append the continuous absolute gripper position to each delta action. Similar to proprioception, we present the delta action by the relative end effector translation and a 6DoF rotation. This results in a 10-dimensional action representation. When rolling out the predicted actions, in addition to temporal ensembling [2], we also use receding horizon control [1], and select an action horizon of 10 steps.

8.6 System Information

All models are trained on 4 NVIDIA A100 80GB GPUs. ICRT pre-training on DROID takes 56 minutes and fine-tuneing on ICRT-MT takes 18 hours. ICRT-Llama7B takes roughly 28 hours to finetune. We report the inference speed of ICRT and ICRT-Llama2 in Table 10 averaged over 100 steps. All tests are performed on a workstation with NVIDIA RTX 3090Ti and Intel i5-12400F with 64GB memory. We find that using the proposed formulation, which can leverage the KV cache, we can run ICRT-Llama2 at 10Hz naively.

	Inference Frequency
ICRT	39.6 Hz
ICRT-Llama2	10.7 Hz

Table 10: Inference frequency of ICRT, averaged over 100 steps.