In-Context Imitation Learning via Next-Token Prediction

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 enhance next-token prediction models to perform in-context imitation learning on a real robot, where the robot executes new tasks by interpreting contextual information provided during the input phase, without updating its underlying policy parameters. We propose In-Context Robot Transformer (ICRT), a causal transformer that performs autoregressive prediction on sensorimotor trajectories without relying on any linguistic data or reward function. This formulation enables flexible and training-free execution of new tasks at test time, achieved by prompting the model with sensorimotor trajectories of the new task composing of image observations, actions and states tuples, collected through human teleoperation. 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 prompt and the training data. In a multi-task environment setup, ICRT significantly outperforms current state-of-the-art next-token prediction models in robotics on generalizing to unseen tasks. Code, checkpoints and data are available on https://icrt.dev.

Keywords: Multi-Task Learning, Next-Token Prediction, In-Context Learning

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.

[*]Equal Contribution
[1]University of California, Berkeley
[2]Autodesk Research

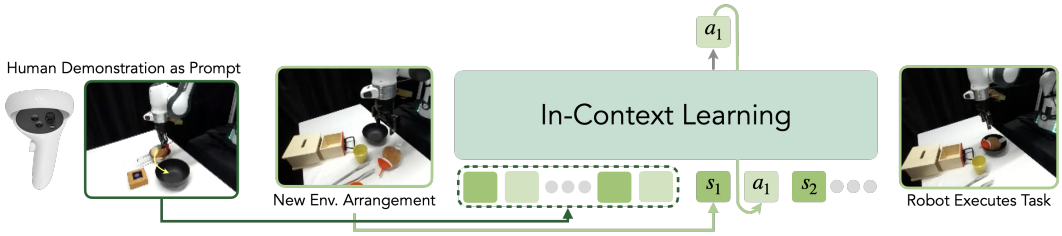

Figure 1: In-Context Robot Transformer. When we want a robot to learn a new task, in many cases, either we have to program new primitives to perform the task, or we have to provide many human demonstrations to train an imitation learning model. *Can a model learn the new task with few demonstrations without training?* We train a next-token prediction model to perform in-context imitation learning on a real robot. In particular, the model learns from robot trajectories to perform continuous action predictions. At inference time, we prompt the model with robot sensorimotor trajectories collected by human teleoperation and provide the model with the observation of the new environment, and roll out the policy on a physical robot.

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-world 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 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 one or few-shot imitation learning approaches, ICRT offers a simple framework that avoids complicated loss functions or key-point selection, and operates directly on 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.

Moreover, we observe that certain properties of the dataset are crucial for enabling effective in-context learning on real robots. Specifically, datasets that allow multiple tasks to be performed from the same initial observation are particularly beneficial. In such scenarios, unlike existing single-task datasets and many multi-task datasets where each environment has a unique object for the robot to interact with, the model must 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 next-token prediction model that performs in-context learning on a real robot. ICRT takes robot's sensorimotor trajectories on new tasks as context to perform specified tasks in unseen environment configurations.
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. We perform physical experiments on a Franka Emika robot at various levels of task difficulties to evaluate the in-context learning abilities of ICRT. Results suggest that ICRT can perform the unseen tasks specified by the prompt.

2 Related Works

2.1 Imitation Learning for Robotics

Imitation learning is a popular and 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 these 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]. This approach has also been extended to robotic learning, where robot action planning is framed as a next-token prediction task using transformer-based architectures [35, 36, 7, 8, 14, 15]. In these models, observations are used to predict the next robot actions. In addition, in pursuit of developing generalist agents and robust robot policies, recent research has demonstrated that training policies on large, diverse datasets encompassing multiple tasks can lead to more robust and generalizable models [37, 38, 39, 40, 15, 5, 7, 36]. Octo [15] and OpenVLA [14] are trained on large robotic datasets, and are the state-of-the-art policies conditioning on goal images and language instructions (Octo) or just language instructions (OpenVLA). Octo employs a transformer with a diffusion head, which processes the goal conditions—language instructions or goal images—and the current image observation to predict robot actions. OpenVLA fine-tunes a pre-trained vision-language model to predict robot actions given vision observations and language instructions.

2.2 In-Context Learning

Despite the effort of training on large datasets, these policies still struggle with novel tasks or environments and often require fine-tuning. Several works have explored ways to bypass the need for model fine-tuning or to increase sample efficiency when generalizing to new tasks, leading to advances in zero-shot and few-shot imitation learning. Some approaches in meta-learning [41, 42, 43] enable few-shot imitation learning after training on a wide range of tasks, but still require fine-tuning in each new domain. Other works don't require fine-tuning model parameters for generalizing to new tasks. Brown et al. [33] refers to this as "in-context learning" to differ from works that fine-tune the model parameters.

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, 44]. However, how to effectively integrate these methods within the next-token-prediction framework remains unclear. Valassakis et al. [45] 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 and Johns [46] 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 [47] 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 observation, while [49] employs a short trajectory prompt to encode task-specific information for guiding policy generation in offline reinforcement learning. However, neither of these approaches considers image observations as inputs, nor do they extend beyond tasks in simulation. In contrast, 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 investigate in-context imitation learning on a real-robot in a continuous control setting. The objective is to train a model with in-context learning capabilities using a multi-task dataset. At test time, the model can perform an unseen task in a new environment configuration by taking a few new human-teleoperated robot demonstrations as a prompt. We define environment configuration as the objects in the scene and their locations. Importantly, this is accomplished *without any additional training* on the new demonstrations.

We define motion primitives as distinct robot motions used to complete different tasks. Each task is characterized by 1) a motion primitive and 2) the set of objects the robot interacts with using that primitive. By varying the test-time environment configuration from the one in the prompt, we evaluate the model's ability to determine the appropriate motion primitive and identify the correct object to interact with. 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 dataset consisting of diverse demonstrations of a single robot. Each demonstration trajectory contains observations from an RGB camera at a fixed position and a wrist-mounted RGB camera, proprioception, action, and the associated task type.
2. The task tested on the robot can be completed by human teleoperating the robot and is thus 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 transformer-based policy and its training formulation to 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 (see Appendix Section 8.4 for more detail). 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 joint effort from different organizations and 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 can be completed in the given environment. In such setup, the model can learn the shortcut of performing the task just conditioned on the current state and observation and never looks at the prompt, even

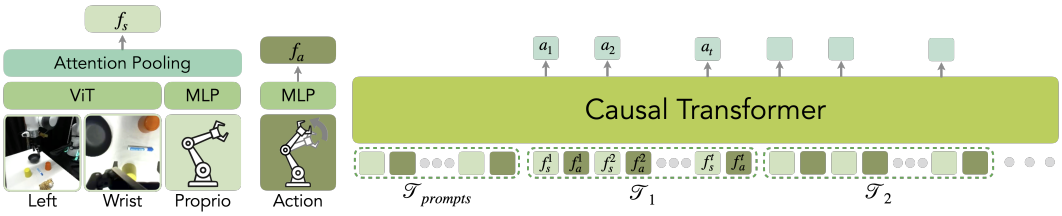

Figure 2: Method Overview: (Left) We encode the left and wrist camera observation with a pre-trained vision transformer. Additionally, we encode proprioception with a multilayer perceptron (MLP). We concatenate the visual latent and the proprioception's latent and use attention pooling to extract a feature f_s to represent the current state. We use another MLP to encode the action taken at the current step as the action feature f_a. (Right) We concatenate multiple trajectories of the same task and randomly sample the first k trajectories as the prompt. We encode the trajectories via a causal transformer, and the model decodes a series of tokens. We decode the tokens that are at the position of the state features to generate the next $h = 16$ action via a MLP.

though the prompt trajectories are similar to the current task to be performed. Therefore multi-task data is crucial for the model to learn from the prompt. We manually collected a multi-task dataset ICRT-Multi-Task (ICRT-MT) using the DROID setup (Figure. 4). This dataset has 1098 trajectories in total, and contains 29 tasks with 6 primitives: picking, pick-and-place, stacking, pushing, poking, opening and closing drawers. Objects used in the data collection and examples of the primitives are shown in Figure. 4. In ICRT-MT, each environment is set so that there exist more than 2 possible tasks for the current observation so that the model has 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 subsample for 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

To facilitate in-context learning in a robotics setting, the model should have a sufficiently long context window to support prompting by providing robot trajectories as demonstrations. 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 2).

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 an efficient 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 2 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 the robot's state or an action.

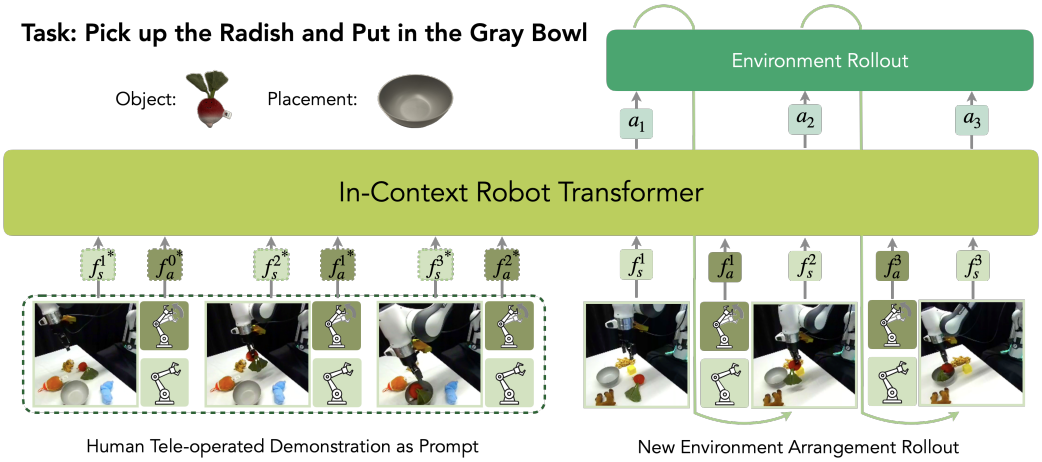

Figure 3: Example inference pipeline of ICRT on the task of picking up the radish and putting in the gray bowl. A human teleoperated demonstration trajectory consisting of image observations, proprioception and actions are provided as the prompt. ICRT takes the prompt and the current observation in a different environment to accomplish the task.

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.

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 sampled 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. 3, we provide one or more human-teleoperated demon-

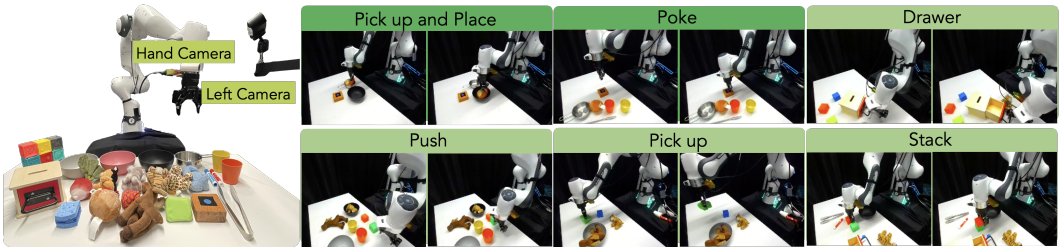

Figure 4: Physical experiments setup are shown on the left, showing the Franka Emika robot, the wrist and side camera and the objects used in training and evaluation. We consider 6 primitives and collect human demonstrations for training. We consider pick up and place and poke primitives for evaluation (dark green).

strations 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.

5 Experiments

In this section, we design an experimental setup to evaluate the in-context learning capabilities of the proposed models for continuous robot control 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 unseen tasks among all executable options from 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 and three evaluating on objects unseen during training (selected from *radish, blue sponge, grey dog, and black dog*, see Table 1 and Table 2).

Each task is designed with five tiers of difficulty. In the *pick-and-place* primitive, the model is tasked with identifying which object to grasp and where to place it in a multi-object or multi-placement setting. The tiers are: 1) pick and place the object without any distractors, 2) with one distractor object, 3) with two distractor objects, 4) with three distractor objects, and 5) with one distractor placement position. For the *poking* primitive, the robot must close the gripper, poke the object, lift the end-effector, and open the gripper. The five tiers of difficulty involve the target object presented with 0-4 distractors in the scene.

The pick-and-place primitive is evaluated by assigning a partial credit of 0.5 if the robot correctly picks up the object. A successful placement results in a total score of 1. The poking task is evaluated by whether the model pokes the correct object; if an incorrect object is poked, the trial is marked as a failure. The model is allowed retries within a time limit of 25 seconds (or 375 steps). Each tier of difficulty is performed once, and we report the average success rate per task, as well as the average success rate and standard deviation per action primitive across the six tasks.

Algorithms The default **ICRT** model 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

Pick and Place							
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.)
Goal Conditioned	40%	30%	20%	40%	40%	30%	33.3% (±7.5%)
Octo	10%	0%	10%	10%	0%	0%	5.0% (±5.0%)
OpenVLA	0%	0%	0%	50%	20%	0%	11.7% (±18.6%)
ICRT-Llama2	40%	40%	40%	60%	40%	40%	43.3% (±7.5%)
ICRT (DROID)	0%	0%	0%	0%	0%	0%	0.0% (±0.0%)
ICRT (MT)	**90%**	**50%**	**80%**	**90%**	**60%**	**90%**	**76.7% (±16.0%)**
ICRT	60%	**50%**	**80%**	50%	**60%**	**90%**	65.0% (±15.0%)

Table 1: Pick up and place primitive performed with goal conditioned or by using one sequence as the prompt.

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 algorithms. We train a goal-conditioned policy, where in each sample of the dataset, the goal observation and state pair are always prepended to the sequence, and in each sequence, there exists only 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 algorithm. 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 8.1 Figure 5 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.

	Poke						
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% (±14.9%)
Octo	20%	0%	60%	0%	0%	0%	13.3% (±22.1%)
OpenVLA	20%	0%	0%	0%	0%	0%	3.3% (±7.4%)
ICRT-Llama2	60%	**100%**	80%	60%	60%	80%	73.3% (±14.9%)
ICRT (DROID)	0%	0%	0%	0%	0%	0%	0.0% (±0.0%)
ICRT (MT)	**100%**	**100%**	40%	60%	60%	60%	70.0% (±22.4%)
ICRT	**100%**	**100%**	**80%**	**80%**	**100%**	**100%**	**93.3% (±9.4%)**

Table 2: Poking primitive performed with goal conditioned or by using one sequence as the prompt.

Results We present the results in Table 1 and Table 2. 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 with determining which object to interact with and where it should be placed, highlighting the challenges posed by our experimental setup for multi-task policies. OpenVLA, while often moving towards the correct object, frequently fails in grasping the object or mistakenly performs the wrong task (e.g., grasping instead of poking, and vice versa). This indicates that OpenVLA may require a greater number of demonstrations (more than 50) per task to achieve better performance, and that relying solely on language conditioning may not be sufficient for generalization to new tasks.

The results suggest that ICRT outperforms the goal-conditioned policy in identifying the correct object to pick up and the appropriate placement location. The poking task presents a significant challenge for the goal or language-conditioned policies, as the goal position often closely resembles the start configuration. However, after conditioning on the prompt trajectory, ICRT is able to correctly identify the task as poking, and the results indicate that it consistently reaches the correct target

object while ignoring distractors. Despite this, we do observe some failure modes with ICRT, such as missing the grasp of the target object, grasping the wrong object, or placing objects in incorrect locations. Specifically, when a distractor object shares the same color but has a different shape, the model struggles to accurately determine which object to grasp. This implies that additional fine-tuning of the vision encoder might be required to enhance model performance, a conclusion also reached by OpenVLA [14].

6 Ablations

In this section, we provide additional experiments presented Table 1 and Table 2 that ablate on a few core design choices. We provide more ablation studies in Section 8.2.

6.1 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 and Table 2, 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. We suggest that future work should focus on optimizing the inference speed of ICRT-Llama2.

6.2 Training Dataset

We find that training on the DROID subset (see Section 4.1) is insufficient for successfully 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 pretrained 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.

6.3 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 3 and Table 4. We find that by letting the model only predict the trajectories after the designated prompt trajectories, the model's performance improves significantly. We hypothesize that in the situation where there is a loss on the prompt trajectories, the model is forced to do un-conditional generation based on the current environment observation, especially when there are multiple possible tasks available. This may cause the model to stop paying attention to the prompt.

	Pick and Place					
Pick Object	Yellow Cube	Yellow Cube	Blue Bear	Radish	Black Dog	Blue Sponge
Place Location	Black Bowl	Grey Bowl	Pink Bowl	Grey Bowl	Pink Bowl	Silver Pot
ICRT +Prompt Loss	20%	10%	20%	40%	30%	10%
ICRT	**60%**	**50%**	**80%**	**50%**	**60%**	**90%**

Table 3: Ablation on not applying loss on the prompt trajectories for pick and place tasks.

Poke Object	Radish	Red Cube	Grey Dog	Poke Black Cube	Pink Bowl	Blue Sponge
ICRT +Prompt Loss	0%	20%	20%	**80%**	0%	20%
ICRT	**100%**	**100%**	**80%**	**80%**	**100%**	**100%**

Table 4: Ablation on not applying loss on the prompt trajectories for poking tasks.

7 Limitations and Conclusion

This method has a few limitations. While results suggest that ICRT can generalize the primitive to unseen objects and certain primitives that resemble the ones in training (see Section 8.2.3), it is still unclear how to generalize to unseen primitives. Future works should investigate how scaling model capacity and scaling dataset can help with primitive-level generalization. In addition, ICRT assumes a fixed robot morphology with a fixed impedance controller. Future works can also investigate how to facilitate transfer between different robot morphologies by learning a unified policy on different robots. ICRT-Llama2 has a low inference frequency which may contribute to its low performance. We hope to speed up ICRT-Llama2 at inference time in the future.

In summary, we present ICRT, where we study in-context imitation learning on a real robot. We do so by training a causal transformer model on sequences of robot trajectories, where trajectories of the same task are combined to serve as the context for performing the task. We also present a corresponding multi-task dataset to help facilitate this in-context learning. We find that by using robot sensorimotor trajectories as the context, the model can generalize the learned primitives to unseen objects and different environment configurations, especially in environments where more than one task is present.

Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, and the CITRIS "People and Robots" (CPAR) Initiative. In their academic roles at UC Berkeley, Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, and Ken Goldberg are supported in part by donations from Autodesk, Meta, Google, Siemens, Toyota Research Institute, Bosch, and by equipment grants from PhotoNeo, Nvidia, NSF AI4OPT Centre, and Intuitive Surgical. L.Y. Chen is also supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. 2146752. We thank Xinyang Geng, Dantong Niu, and Chung Min Kim for their helpful discussions and feedback.

References

[1] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

[2] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[3] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023.

[4] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, et al. A generalist agent. *arXiv:2205.06175*, 2022.

[5] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. ViNT: A Foundation Model for Visual Navigation. In *7th Annual Conference on Robot Learning (CoRL)*, 2023.

[6] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar. RoboAgent: Towards sample efficient robot manipulation with semantic augmentations and action chunking. *arxiv*, 2023.

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv:2212.06817*, 2022.

[8] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[9] X. Chen, J. Djolonga, P. Padlewski, B. Mustafa, S. Changpinyo, J. Wu, C. R. Ruiz, S. Goodman, X. Wang, Y. Tay, S. Shakeri, M. Dehghani, D. Salz, M. Lucic, M. Tschannen, A. Nagrani, H. Hu, M. Joshi, B. Pang, C. Montgomery, P. Pietrzyk, M. Ritter, A. Piergiovanni, M. Minderer, F. Pavetic, A. Waters, G. Li, I. Alabdulmohsin, L. Beyer, J. Amelot, K. Lee, A. P. Steiner, Y. Li, D. Keysers, A. Arnab, Y. Xu, K. Rong, A. Kolesnikov, M. Seyedhosseini, A. Angelova, X. Zhai, N. Houlsby, and R. Soricut. Pali-x: On scaling up a multilingual vision and language model, 2023.

[10] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, et al. Palm-e: An embodied multimodal language model. *arXiv:2303.03378*, 2023.

[11] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[12] 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.

[13] Y. Bai, X. Geng, K. Mangalam, A. Bar, A. Yuille, T. Darrell, J. Malik, and A. A. Efros. Sequential modeling enables scalable learning for large vision models. *arXiv preprint arXiv:2312.00785*, 2023.

[14] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

[15] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

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

[17] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, et al. Scalable deep reinforcement learning for vision-based robotic manipulation. In *CoRL*, 2018.

[18] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *IJRR*, 2018.

[19] C. Eppner, A. Mousavian, and D. Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *2021 IEEE Int. Conf. on Robotics and Automation, ICRA*, 2020.

[20] N. M. M. Shafiullah, A. Rai, H. Etukuru, Y. Liu, I. Misra, S. Chintala, and L. Pinto. On bringing robots home, 2023.

[21] H.-S. Fang, H. Fang, Z. Tang, J. Liu, J. Wang, H. Zhu, and C. Lu. RH20T: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023.

[22] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv:2109.13396*, 2021.

[23] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine. Bridgedata v2: A dataset for robot learning at scale, 2023.

[24] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv:2203.12601*, 2022.

[25] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik. Masked visual pre-training for motor control. *arXiv:2203.06173*, 2022.

[26] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[27] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. *arXiv:2210.03109*, 2022.

[28] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky, editor, *NeurIPS*, volume 1. Morgan-Kaufmann, 1988.

[29] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[30] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *JMLR*, 2016.

[31] P. R. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. S. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In *CoRL*, 2021.

[32] H. Ha, P. Florence, and S. Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023.

[33] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[34] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[35] I. Radosavovic, B. Shi, L. Fu, K. Goldberg, T. Darrell, and J. Malik. Robot learning with sensorimotor pre-training. *arXiv:2306.10007*, 2023.

[36] I. Radosavovic, B. Zhang, B. Shi, J. Rajasegaran, S. Kamat, T. Darrell, K. Sreenath, and J. Malik. Humanoid locomotion as next token prediction, 2024.

[37] 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 *Conference on Robot Learning*, 2022.

[38] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. VIMA: General robot manipulation with multimodal prompts. *International Conference on Machine Learning (ICML)*, 2023.

[39] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[40] E. Collaboration, A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Mart'in-Mart'in, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin, S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Cho,

Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open x-embodiment: Robotic learning datasets and rt-x models, 2024.

[41] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[42] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pages 357–368. PMLR, 2017.

[43] M. Xu, Y. Lu, Y. Shen, S. Zhang, D. Zhao, and C. Gan. Hyper-decision transformer for efficient online policy adaptation. *arXiv preprint arXiv:2304.08487*, 2023.

[44] Z. Mandi, F. Liu, K. Lee, and P. Abbeel. Towards more generalizable one-shot visual imitation learning, 2022. URL https://arxiv.org/abs/2110.13423.

[45] E. Valassakis, G. Papagiannis, N. Di Palo, and E. Johns. 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)*, pages 8614–8621. IEEE, 2022.

[46] N. Di Palo and E. Johns. Keypoint action tokens enable in-context imitation learning in robotics. *arXiv preprint arXiv:2403.19578*, 2024.

[47] 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. *arXiv preprint arXiv:2403.12943*, 2024.

[48] Y. Duan, M. Andrychowicz, B. Stadie, O. Jonathan Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba. One-shot imitation learning. *Advances in neural information processing systems*, 30, 2017.

[49] M. Xu, Y. Shen, S. Zhang, Y. Lu, D. Zhao, J. Tenenbaum, and C. Gan. Prompting decision transformer for few-shot policy generalization. In *international conference on machine learning*, pages 24631–24645. PMLR, 2022.

[50] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset, 2024.

[51] A. Majumdar, K. Yadav, S. Arnaud, Y. J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, P. Abbeel, J. Malik, D. Batra, Y. Lin, O. Maksymets, A. Rajeswaran, and F. Meier. Where are we in the search for an artificial visual cortex for embodied intelligence? *arXiv preprint arXiv:2303.18240*, 2023.

[52] S. Chen, R. Garcia, I. Laptev, and C. Schmid. Sugar: Pre-training 3d visual representations for robotics. *arXiv preprint arXiv:2404.01491*, 2024.

[53] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[55] L. Fu, L. Lian, R. Wang, B. Shi, X. Wang, A. Yala, T. Darrell, A. A. Efros, and K. Goldberg. Rethinking patch dependence for masked autoencoders. *arXiv preprint arXiv:2401.14391*, 2024.

[56] 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 *International conference on machine learning*, pages 3744–3753. PMLR, 2019.

[57] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[58] J. Han, R. Zhang, W. Shao, P. Gao, P. Xu, H. Xiao, K. Zhang, C. Liu, S. Wen, Z. Guo, X. Lu, S. Ren, Y. Wen, X. Chen, X. Yue, H. Li, and Y. Qiao. Imagebind-llm: Multi-modality instruction tuning, 2023.

[59] L. Fu, G. Datta, H. Huang, W. C.-H. Panitch, J. Drake, J. Ortiz, M. Mukadam, M. Lambeta, R. Calandra, and K. Goldberg. A touch, vision, and language dataset for multimodal alignment. *arXiv preprint arXiv:2402.13232*, 2024.

[60] S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. G. Arenas, K. Rao, D. Sadigh, and A. Zeng. Large language models as general pattern machines, 2023.

[61] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[62] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[63] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning, 2023.

[64] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, and S. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

8 Supplementary Material

8.1 Scene Illustrations

We provide an 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.

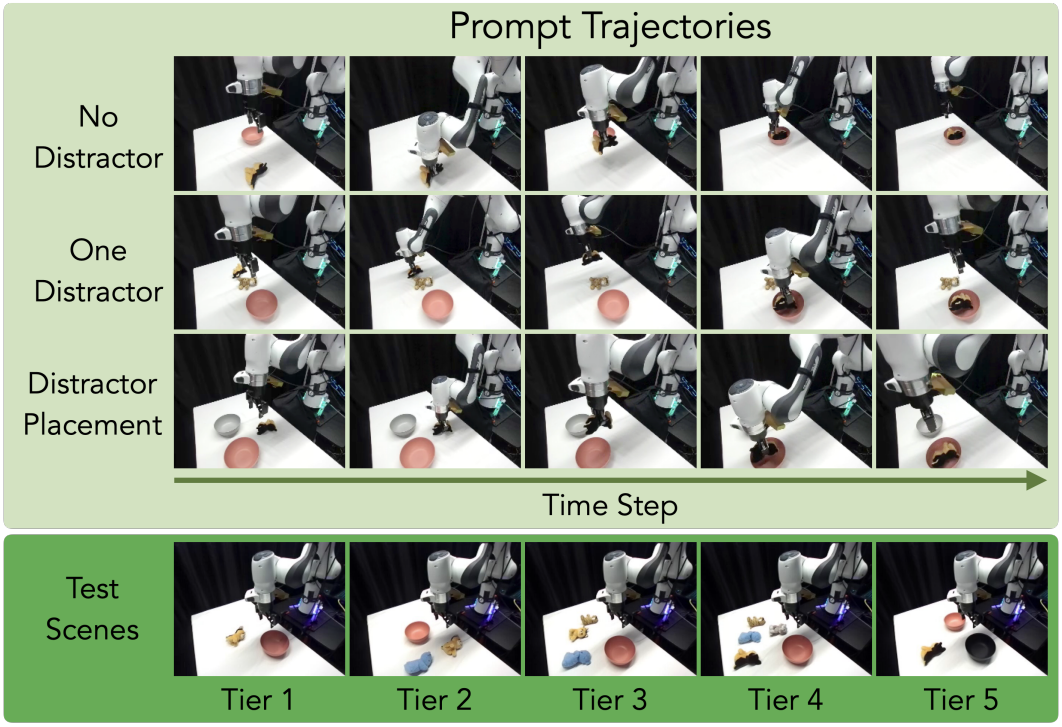

Figure 5: Illustrations of the prompt trajectories (top) and test scenes (bottom) for the pick up the black dog and place in the pink bowl task. Three prompt trajectories of different types are collected. The test scenes are different from all prompt trajectories and 5 tiers of scenes with different 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 5. The low std from Table 5 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 5: 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 a total of 25 trials.

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

Table 6: 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.

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 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 6. We conduct experiments with one prompt trajectory of different types (the first three columns in Table 6), 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 7: 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 7, 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 8 and Table 9. 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 26 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					
Pick Object	Yellow Cube	Yellow Cube	Blue Bear	Radish	Black Dog	Blue Sponge
Place Location	Black Bowl	Grey Bowl	Pink Bowl	Grey Bowl	Pink Bowl	Silver Pot
ICRT (Co-train)	10%	0%	10%	0%	40%	20%
ICRT	**60%**	**50%**	**80%**	**50%**	**60%**	**90%**

Table 8: Ablation on co-training with DROID [50] for pick up and place tasks.

Poke Object	Radish	Red Cube	Grey Dog	Poke Black Cube	Pink Bowl	Blue Sponge
ICRT (Co-train)	0%	0%	0%	0%	0%	0%
ICRT	**100%**	**100%**	**80%**	**80%**	**100%**	**100%**

Table 9: Ablation on co-training with DROID [50] for poking tasks.

8.3 Hyperparameters

We provide the hyperparameters for both the pre-training and fine-tuning phase in Table 10 and Table 11.

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 10: Pre-training Hyperparameters

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 11: Finetuning Hyperparameters

8.4 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, 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.

8.5 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 12 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 12: Inference frequency of ICRT, averaged over 100 steps.