Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

CLIP-RT: Learning Language-Conditioned Robotic Policies from Natural Language Supervision

Gi-Cheon Kang[1*] Junghyun Kim[1*] Kyuhwan Shim[1] Jun Ki Lee[1†] Byoung-Tak Zhang[1,2†]
[1]Seoul National University [2]Tommoro Robotics
https://clip-rt.github.io

Abstract—**Teaching robots desired skills in real-world environments remains challenging, especially for non-experts. A key bottleneck is that collecting robotic data often requires expertise or specialized hardware, limiting accessibility and scalability. We posit that natural language offers an intuitive and accessible interface for robot learning. To this end, we study two aspects: (1) enabling non-experts to collect robotic data through natural language supervision (*e.g.,* "move the arm to the right") and (2) training robot policies directly from this supervision. Specifically, we introduce a data collection framework that collects robot demonstrations based on natural language supervision and further augments these demonstrations. We then present CLIP-RT, a new vision-language-action (VLA) model that learns language-conditioned visuomotor policies from this supervision. CLIP-RT adapts the pretrained CLIP model and learns to predict language-based motion primitives via contrastive imitation learning. We train CLIP-RT on the Open X-Embodiment dataset and finetune it on in-domain data collected by our framework. In real-world evaluations, CLIP-RT demonstrates strong capabilities in learning novel manipulation skills, outperforming OpenVLA (7B parameters) by 24% in average success rates, while using 7x fewer parameters (1B). We further assess CLIP-RT's capabilities in few-shot generalization and collaborative scenarios involving large pretrained models or humans. In simulated environments, CLIP-RT also yields strong performance, achieving a 92.8% average success rate on the LIBERO benchmark with an inference throughput of 163 Hz.**

I. Introduction

Building robots that can understand natural language instructions and perform various real-world tasks is a long-standing goal of robotics and artificial intelligence. The research community has studied such robots in various domains, such as robotic manipulation [37, 6, 7], navigation [2, 11, 54, 32], and other instructions-following tasks [50, 42].

One key challenge for intelligent robots is grounding natural language to vision and action, bridging the abstraction gap between natural language instruction and visuomotor control in real-world tasks. Prior works on robotic manipulation have addressed this challenge by training language-conditioned policies, primarily through imitation learning [53, 35, 51, 26, 37, 6, 7]. This line of research has shown remarkable success as large amounts of robotic data become available [41]. However, even state-of-the-art models [7, 41, 3, 29] trained in large-scale robot data struggle to easily expand their set of manipulation skills for a wide range of real-world tasks. We argue that a major bottleneck lies in how robot demonstrations are typically collected. Specifically, obtaining real-world robot demonstration data often requires expertise in robot control or access to specialized hardware, such as teleoperation or virtual reality (VR) systems [58, 18]. This barrier severely limits accessibility, restricting the number of participants and environments from which data can be gathered. Consequently, this limited accessibility inherently hinders both the scalability (the volume of data) and the diversity (the range of scenarios and behaviors recorded) of the resulting datasets. We thus ask: *how can non-experts train robotic policies without relying on specialized expertise or devices for data collection?*

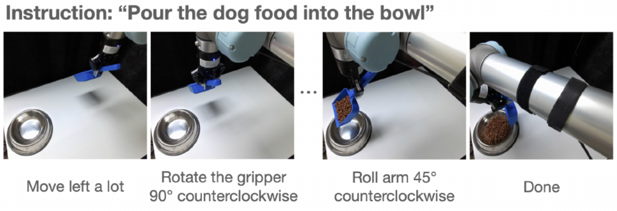

Instruction: "Pour the dog food into the bowl"
...
Move left a lot
Rotate the gripper 90° counterclockwise
Roll arm 45° counterclockwise
Done

Fig. 1: Overview of language-guided teleoperation.

We argue that natural language is an intuitive and accessible interface for robot learning. We thus explore a method for training robotic skills through natural language. To this end, we propose a data collection framework that enables non-experts to collect in-domain robot data through natural language. It consists of two steps: language-based teleoperation and stochastic trajectory augmentation (STA). Figure 1 illustrates language-based teleoperation in which a human collects data for a skill described in the instruction (*e.g.,* "pour the dog food into the bowl"). The human first provides natural language supervision (*e.g.,* "move left a lot") in each state. The large language model (LLM) [39] then translates this supervision into appropriate robotic behavior, which is ultimately executed by the robot. By repeating this process, we obtain a collection of robot demonstrations, where each state transition is associated with corresponding language supervision. After the language-based teleoperation, STA augments the demonstration into alternative trajectories. Specifically, it stochastically drives the robot into novel states that were not explicitly covered in the original demonstrations. STA then automatically labels the appropriate behavior at these novel states using a simple heuristic. In other words, STA generates new trajectory data, expanding the diversity of the training

* equal contribution; † equal advising.

dataset beyond the original demonstrations.

We introduce a vision-language-action (VLA) model that learns language-conditioned visuomotor policies from natural language supervision, which we call CLIP-RT (CLIP-based Robotics Transformer). A key idea is to leverage natural language as supervision to train visuomotor policies—inspired by CLIP [45], which uses language as a training signal for visual representation learning. CLIP-RT employs CLIP models trained in Internet-scale data [47, 17] and directly adapts them to predict language-based motion primitives (*e.g.,* "move the arm forward by 10cm") through contrastive imitation learning. Specifically, our model learns to measure the pairwise similarity between language supervision and contextual information (*i.e.,* current scene and language instruction) for language-conditioned policies. We train CLIP-RT through a two-step process: pretraining and in-domain fine-tuning. In the pretraining stage, we train our model on the large-scale robot learning dataset—Open X-Embodiment [41]—to improve generalization capabilities. The dataset does not contain language supervision, so we transform existing low-level robotic actions into templated natural language supervision to train CLIP-RT. During in-domain fine-tuning, CLIP-RT learns diverse robotic skills using our collected data.

Our contributions are fivefold. First, we propose CLIP-RT, a vision-language-action (VLA) model that learns language-conditioned policies from natural language supervision. Second, we propose a data collection framework that enables non-experts to collect robot data only through natural language and augment the human-collected demonstration data. Third, experiments demonstrate that CLIP-RT outperforms OpenVLA [29] by 24% in average success rates in 9 novel manipulation tasks. We further observe two important results: (1) language-based motion prediction and STA boost generalization capabilities of CLIP-RT and (2) CLIP-RT effectively learns shared structures across diverse robotic tasks, resulting in generalizable and transferable policies. Fourth, we demonstrate that CLIP-RT's language-based motion prediction capability enables collaboration with humans and large pretrained models [24], resulting in improved generalization. Fifth, to validate the generality of our method, we adapt CLIP-RT and evaluate it on the LIBERO simulation benchmark [33] that includes offline, human-teleoperated demonstrations. CLIP-RT achieves strong results, an average success rate of 92.8%, with an improved inference throughput of 163Hz.

II. Related Work

Vision-Language-Action (VLA) Models. Vision-language models (VLM) trained on Internet-scale data have been widely studied in robotics, including high-level planning [13, 22], success detection [14], and physical reasoning [19]. In particular, previous work [7, 41, 3, 29] directly fine-tunes VLMs to predict robotic actions. This category of models is called vision-language-action (VLA) models. CLIP-RT belongs to this category. Current VLA models discretize continuous action values (*e.g.,* end-effector actions) into discrete action tokens and learn to generate a sequence of these tokens. Unlike existing VLA models, CLIP-RT is a *discriminative* VLA model that predicts actions in a predefined list of actions, and these actions are represented in natural language (*e.g.,* "move the arm left") rather than low-level control commands.

Collecting Real-World Robot Demonstrations. Data collection has become an increasingly important challenge in robot learning. Previous works have collected real-world robot demonstrations through various interfaces, such as teleoperation devices [18, 1], virtual reality (VR) [61, 48], and kinesthetic teaching [4, 36, 16]. Some studies introduce natural language interfaces [34, 3] for data collection, but they are often used in limited scenarios. RT-H [3] and OLAF [34] first train visuomotor policies using data collected from other interfaces (*e.g.,* VR). During deployment, humans provide language feedback to correct robotic behaviors, and policies are updated based on this feedback. In other words, these methods focus on refining learned policies for *existing* skills. In contrast, our focus is to teach *any desired* skills by collecting complete demonstration trajectories through language-based teleoperation. To achieve this, our framework uses the in-context learning capabilities of large language models (LLMs) [20] to translate language supervision into action.

Language-Conditioned Policies. The research community has made extensive efforts to develop robotic systems that can follow language instructions [31, 8, 54, 50, 28, 27], often training language-conditioned policies [53, 35, 51, 26, 37, 6, 7, 29, 3]. We train language-conditioned visuomotor policies through imitation learning, similar to existing studies. Unlike existing studies, we train language-conditioned policies with contrastive imitation learning, which combines the ideas of contrastive learning [45] with imitation learning [43] for more discriminative representations of robotic behaviors.

III. Approach

A. Preliminaries

Language-Conditioned Imitation Learning. A robot dataset $\mathcal{D} = \{(\tau_n, \ell_n)\}_{n=1}^{N}$ consists of a demonstration trajectory τ paired with language instruction ℓ. Each trajectory contains a sequence of visual observations and expert actions $\tau_n = \{(v_1, a_1), \ldots, (v_{|\tau_n|}, a_{|\tau_n|})\}$. The goal of language-conditioned imitation learning is minimizing the negative log-likelihood of the expert action a_t given the observation history $v_{1:t} = (v_1, \ldots, v_t)$ and language instruction ℓ:

$$\mathcal{L}_{\text{IL}} = -\mathbb{E}_{(\tau,\ell)\sim\mathcal{D}}\left[\sum_{t=1}^{|\tau|} \log \pi_\theta(a_t | v_{1:t}, \ell)\right] \quad (1)$$

where π_θ denotes the policy model with model parameters θ. For vision-language action (VLA) models, θ is initialized from the parameters of vision-language models (VLMs). To maintain consistency with the pretraining setup of the VLMs, existing VLA models [7, 29, 3] typically use a single-image observation v_t rather than utilizing the full observations $v_{1:t}$. At test time, the policy model performs closed-loop robot control until it completes language instructions.

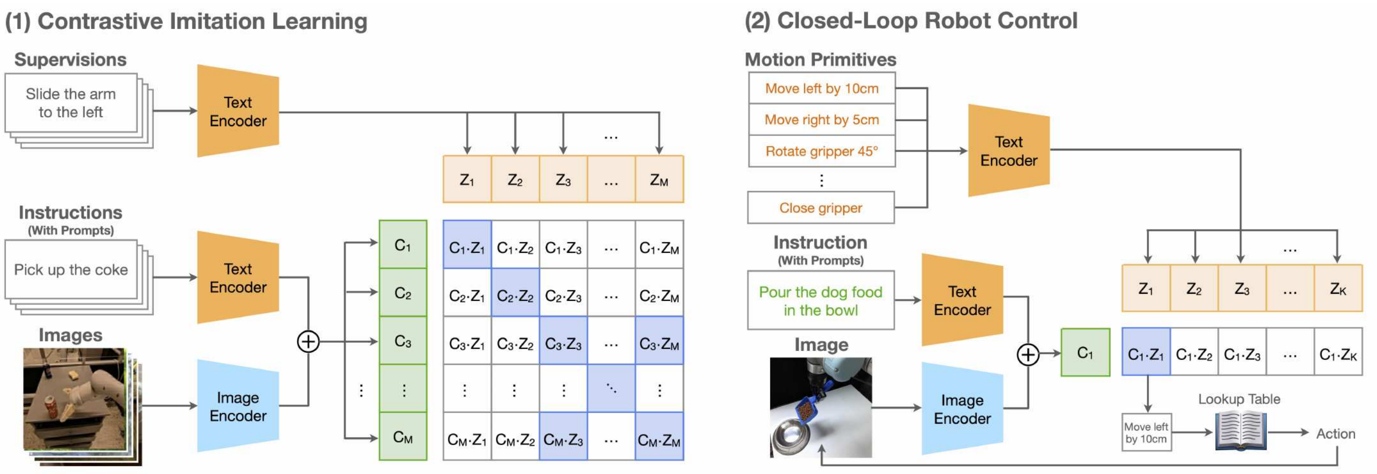


Fig. 2: **Overview of CLIP-RT.** CLIP-RT learns to optimize the pairwise similarity between the context and natural language supervision through contrastive imitation learning. At test time, CLIP-RT predicts the language-based motion primitive with the highest similarity from a list of language motions. We append a simple text prompt to instructions: *What motion should the robot arm perform to complete the instruction* $\{$*instruction*$\}$*?*

Contrastive Language-Image Pretraining (CLIP) [45] is a method to learn visual representations from natural language supervision at scale. Using the contrastive objective, CLIP trains an image encoder $f(\cdot)$ and a text encoder $g(\cdot)$ on 400M image-text pairs. Given a mini-batch of M image-text pairs $\{(I_i, T_i)\}_{i=1}^{M}$, the two encoders are jointly optimized to maximize the similarity between the correct pairs of image and text (I_i, T_i) while minimizing the similarity for incorrect pairs $(I_i, T_{j\neq i})$. As we describe later, we modify the contrastive loss to make CLIP-RT learn language-conditioned policies.

B. CLIP-Based Robotics Transformer (CLIP-RT)

Natural Language Supervision. Inspired by CLIP [45], which uses natural language as a training signal, we built a model to learn robotic policies from natural language. We define natural language supervision as language-based guidance that directs a robot's motion in specific states to complete given instructions. This typically involves shifting the robot's position, orientation, or gripper state (see Appendix A). As we discuss later, each supervision is associated with a specific low-level action. Learning from natural language supervision offers several advantages. It establishes a clear hierarchy between initial instruction and language supervision, enabling models to learn *shared structures* across diverse tasks [3]. Furthermore, language-based learning fosters collaboration with language-capable entities like humans or other AI systems.

Contrastive Imitation Learning (CIL). We describe contrastive imitation learning in Figure 2 (left). CLIP-RT takes a mini-batch of M triplets $\{(v_i, \ell_i, u_i)\}_{i=1}^{M}$, where v, ℓ, and u denote image observation, instruction, and language supervision. CIL aims to optimize the pairwise similarities in the set $\{((v_i, \ell_i), u_j) | i, j \in \{1, \ldots, M\}\}$. Specifically, CLIP-RT first extracts vector embeddings of v_i, ℓ_i and u_j using the CLIP model's image encoder $f(\cdot)$ and the text encoder $g(\cdot)$, and subsequently combines the image and instruction embeddings:

$$\mathbf{c}_i = f(v_i) + g(\ell_i), \quad \mathbf{z}_j = g(u_j) \tag{2}$$

where $\mathbf{c}_i$ represents the context that encapsulates the robot's current visual state and its explicit goal. $\mathbf{z}_j$ represents the immediate action that should be taken given the context. We design the loss function as:

$$\mathcal{L}_{\text{CIL}} = -\frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \Big[y_{ij} \log \sigma(\hat{\mathbf{c}}_i \cdot \hat{\mathbf{z}}_j) + (1 - y_{ij}) \log(1 - \sigma(\hat{\mathbf{c}}_i \cdot \hat{\mathbf{z}}_j)) \Big] \tag{3}$$

where $\hat{\mathbf{c}}_i = \frac{\mathbf{c}_i}{\|\mathbf{c}_i\|_2}$ and $\hat{\mathbf{z}}_j = \frac{\mathbf{z}_j}{\|\mathbf{z}_j\|_2}$ are normalized vector embeddings of $\mathbf{c}_i$ and $\mathbf{z}_j$. $\sigma(\cdot)$ is a sigmoid activation function and $y_{ij} \in \{0, 1\}$ denotes a label for pairwise similarity. The loss function maximizes the cosine similarity between context and language supervision for positive pairs, while minimizing it for negative pairs. The label y_{ij} is basically one if $i = j$; otherwise, it is zero. In other words, $((v_i, \ell_i), u_i)$ are positive pairs and $((v_i, \ell_i), u_{j\neq i})$ are negative pairs. However, the mini-batch often contains semantically interchangeable supervisions, such as "move upwards" and "raise the arm". Thus, CIL consults low-level actions a_i associated with language supervision u_i and treats the pair $((v_i, \ell_i), u_{j\neq i})$ as positive if two supervisions share the same low-level action. As a result, y_{ij} is one if $i = j$ or $a_i = a_j$ (see the blue boxes in Figure 2); otherwise, it is zero. Consequently, CLIP-RT learns to measure the likelihood of each motion described in language, given visual observation and language instruction.

Pretraining. We train CLIP-RT on the Open X-Embodiment (OXE) dataset [41], which contains 2.4M robotic trajectories from 70 individual datasets. We specifically use the OXE data curated by Kim et al. [29] to train CLIP-RT. However, the data do not contain natural language supervision, so we extract language supervision from low-level action similar to recent studies [3, 59]. Specifically, the low-level action is represented as a 7-dimensional vector consisting of the end-effector's delta positions, delta orientations, and the gripper open/close. We identify the entry with the dominant value and its corresponding axis for each action. Based on this information, we transform low-level actions into one of 899 templated natural language supervisions (see Appendix A). As a result, we train CLIP-RT on approximately 18.1M transition data through contrastive imitation learning. It requires four H100 GPUs for one day with a batch size of 128.

In-Domain Fine-Tuning. After pretraining, we fine-tune CLIP-RT on in-domain data via contrastive imitation learning. The in-domain dataset consists of 21K transitions in 18 robotic manipulation tasks, collected through our data collection framework. Details about the dataset and data collection are discussed in the following sections (III-C and IV-A).

Closed-Loop Robot Control. Figure 2 (right) shows an overview of closed-loop robot control. At each time step, CLIP-RT computes pairwise similarities between the context and a list of language-based motion primitives. Our model selects the motion with the highest probability. This selected motion is translated into a lower-level end-effector action based on a predefined lookup table (see Appendix B). Finally, the translated end-effector action is executed using inverse kinematics (IK). Unlike existing Transformer-based policy models [7, 6, 41, 3, 29] relying on autoregressive decoding, CLIP-RT predicts each action in a *single* forward pass since it is a discriminative model. CLIP-RT runs at 16Hz on one H100 GPU and 8Hz on one NVIDIA RTX 3090 GPU, both using float32 precision. These results are achieved without any speed-up tricks (*e.g.,* model quantization). Details regarding frequencies are discussed in Appendix F-E.

VLM Backbone & Codebase. CLIP-RT maintains the original CLIP model architecture without any new parameters. As our backbone model, we employ ViT-H-14-378-quickgelu [17, 25], an open-source CLIP model of 986M ($\approx$1B) parameters that achieves state-of-the-art performance in zero-shot image classification [46] at the time of writing. It consists of an image encoder [12] and a text encoder [44], both built on Transformer [57]. All model configurations can be found in the OpenCLIP codebase [25]. A key advantage of this codebase is that strong CLIP models are continuously updated to the dashboard, enabling users to easily use them through a plug-and-play approach.

C. In-Domain Data Collection

Language-Based Teleoperation. This step aims to collect a few robot demonstrations for each skill only through natural language. To this end, we employ a large language model (LLM) [39] and design a scenario where users collect in-domain data through interactions with the LLM. Specifically, users first provide an initial language instruction for each skill. Then, they provide natural language supervision in each state to complete the instruction. The LLM translates the language supervision into the low-level end-effector action based on a detailed text prompt (see Appendix C). Finally, the camera captures the current image observation and the robot executes the translated action. Consequently, we can obtain a sequence of tuples $\{(v_i, \ell_i, u_i, a_i)\}_{i=1}^{N}$ containing visual observation, instruction, natural language supervision, and low-level action. We collect 10 episodes for each skill through this process.

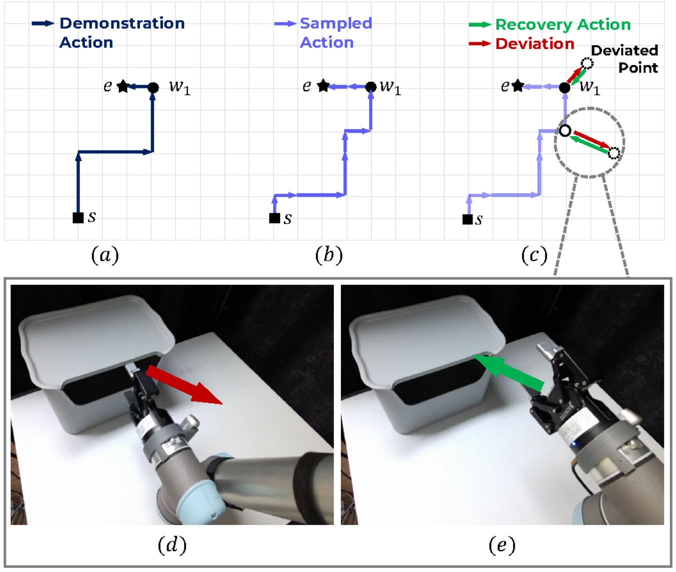


Fig. 3: **A simplified 2D example of stochastic trajectory augmentation (STA).** (a): a demonstration trajectory from the start s to the endpoint e, passing through a waypoint w_1. (b): a sampled trajectory generated by the diversification phase. (c)-(e): a visualization of the recovery phase.

Stochastic Trajectory Augmentation (STA) aims to augment the demonstration data collected from language-based teleoperation. Before delving into the details, we first define a *waypoint* as a key state in demonstrations that satisfies either of the following conditions: (1) the gripper state changes (*i.e.,* open $\rightarrow$ close or close $\rightarrow$ open) or (2) the cumulative progress of delta positions along any axis reverses. For example, w_1 in Figure 3-(a) is a waypoint since cumulative progress on a horizontal axis starts to reverse at w_1. STA consists of two phases: *diversification phase* and *recovery phase*. The diversification phase first builds alternative trajectories toward each waypoint (see Figure 3-(b)) by sampling a new action sequence. The robot then executes each action in the sequence, recording an image in every state it visits. In the recovery phase, STA drives the robot into novel states that deviate from the planned trajectory (see Figure 3-(d)) and then executes a recovery action, a simple reversal of the deviation to return to the trajectory (see Figure 3-(e)). Note that STA records only the recovery actions and images in the deviated states, not the deviation data. By alternating these two phases, STA automatically expands the diversity of the original demonstrations, potentially improving the robustness of policies under varied states. Further details of STA are discussed in Appendix E.

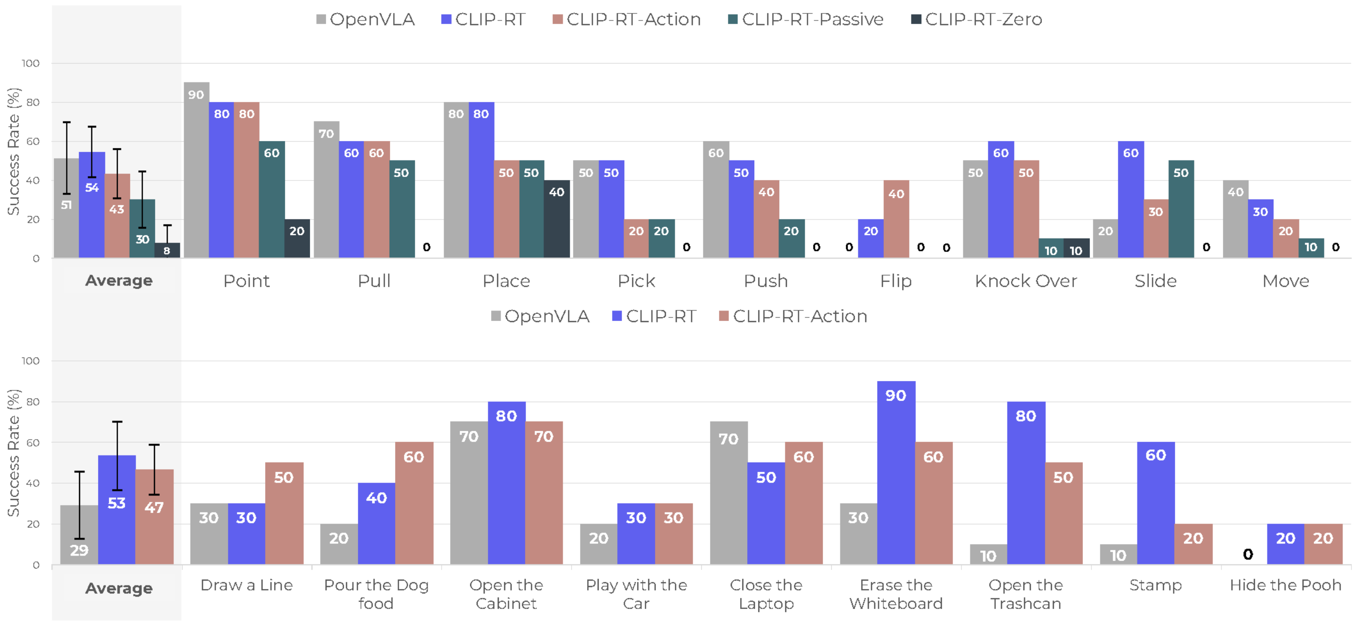


Fig. 4: **Success rates on 9 Common tasks (top) and 9 Novel tasks (bottom).** We conduct experiments using all compared methods on Common tasks and three models (CLIP-RT, OpenVLA and CLIP-RT-Action) on Novel Tasks. The success rate for each task is measured by averaging the results of ten trials. Average success rates of all tasks are shown on the left for both Common and Novel task sets. Tasks are arranged from left to right based on their average number of steps per episode in the training data. The task on the right indicates that it requires more steps in average compared with the task on the left.

IV. Experiments on Real-World Robotic Manipulation

A. Tasks & Dataset

We train and evaluate our models in 18 robotic manipulation tasks, categorized into two groups: *Common* and *Novel*. **Common tasks** consist of nine tasks closely aligned with those in the Open X-Embodiment dataset [41]. These tasks include common manipulation skills, such as "*pick the <obj>*" and "*place the <obj> on the <obj>*". In contrast, **Novel tasks** include nine tasks barely observed during pretraining on the Open X-Embodiment dataset, such as "*stamp on <obj>*", "*play with the toy car*", and "*erase the whiteboard*". This set of tasks serves as a benchmark for evaluating the model's ability to acquire new skills using in-domain data. We first collect in-domain data through language-based teleoperation, gathering 10 episodes per task, resulting in 911 transitions for Common tasks and 1,123 transitions for Novel tasks. Leveraging stochastic trajectory augmentation (STA), we augment each demonstration with 3 additional trajectories across all tasks. This augmentation increases the dataset size to approximately 11K transitions for Common tasks and 10K transitions for Novel tasks. Unless stated otherwise, all the models compared were trained on the same dataset. We provide details of each task, along with visualizations, in Appendix G.

B. Robotic Platform

We perform experiments using a physical robot arm, 6-DoF Universal Robots (UR5) with a two-finger gripper. We provide more details about the robotic platform in the Appendix D.

C. Experiments on Common and Novel Tasks

We train and evaluate CLIP-RT on both Common and Novel tasks, comparing with diverse baselines. We introduce baseline models and then discuss the results in detail.

Baselines. We compare CLIP-RT with four methods, including the state-of-the-art method and ablated versions of our model:

- **CLIP-RT** is our proposed model, pretrained on the Open X-Embodiment (OXE) dataset [41] and further fine-tuned using our in-domain data.
- **OpenVLA** [29] is a state-of-the-art, open-source vision-language-action (VLA) model. This model leverages the 7B-parameter Llama2 language model [55] and a visual encoder that combines pretrained features from DINOv2 [40] and SigLIP [60]. We also fine-tune OpenVLA on the same in-domain data as CLIP-RT by using low-level 7D end-effector actions as supervision.
- **CLIP-RT-Action** is a variant of CLIP-RT where each motion is mapped to existing text tokens that are not frequently used in the vocabulary, similar to existing VLA models [29, 7, 6, 41]. In other words, CLIP-RT-Action represents actions as learned action tokens, rather than representing in natural language. It is also pretrained on the OXE dataset and fine-tuned on in-domain data.
- **CLIP-RT-Passive** is another ablated model of CLIP-RT, which excludes data collected from stochastic trajectory augmentation (STA) and relies solely on data from language-based teleoperation.
- **CLIP-RT-Zero** is an ablated model trained solely on the OXE dataset without accessing any in-domain data.

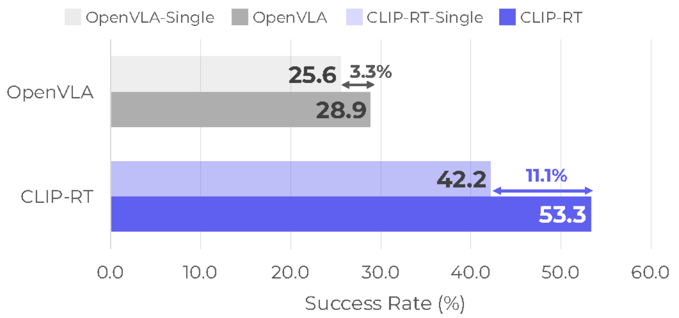


Fig. 5: **A comparison of multi-task and single-task policies on Novel tasks.** The performance of each task is in Figure 12 of Appendix.

Results on Common Tasks. We compare CLIP-RT with all baseline models on Common tasks. The results are summarized in the upper row of Figure 4. CLIP-RT achieves an average success rate of 54%, outperforming all baselines, including OpenVLA and three ablative models. While CLIP-RT outperforms OpenVLA on average, OpenVLA still shows better performance on four basic tasks—Point, Pull, Push, and Move. When comparing CLIP-RT with CLIP-RT-Action, we observe that the use of natural language supervision significantly increases performance on Common tasks (43% → 54%). We hypothesize that CLIP-RT effectively leverages the rich vision-language representations of the pretrained CLIP model [45], allowing it to align language-based motions with semantic concepts. Furthermore, CLIP-RT-Passive, which omits stochastic trajectory augmentation (STA), struggles in most tasks, highlighting the critical role of STA in performance. This suggests that STA enhances robustness and generalization, enabling CLIP-RT to adapt to novel situations. We refer readers to Appendix F-B for a more detailed analysis on the effect of STA. Finally, CLIP-RT-Zero, despite being trained in the large-scale robot learning dataset [41], shows 8% on average success rates, underscoring the need for in-domain fine-tuning.

Results on Novel Tasks. We compare CLIP-RT with OpenVLA and CLIP-RT-Action on 9 Novel tasks. In the lower row of Figure 4, CLIP-RT achieves an average success rate of 53%, outperforming these baselines. Notably, CLIP-RT maintains its average success rates on Novel tasks compared to those of Common tasks, but we observe a significant performance drop of OpenVLA on Novel tasks (51% → 29%). These findings suggest that CLIP-RT generalizes more effectively to tasks that are barely observed in the pretraining dataset. To verify the statistical significance of the performance difference between CLIP-RT and OpenVLA, we conduct a t-test. The resulting p-value is $p = 1.74 \times 10^{-9}$, indicating that CLIP-RT significantly outperforms OpenVLA.

D. In-Depth Analysis of Generalization

We investigate the source of CLIP-RT's improved generalization on Novel tasks. We conduct analyses along three axes: (1) a comparison between multi-task and single-task policies, (2) the effect of natural language supervision, and (3) few-shot generalization.

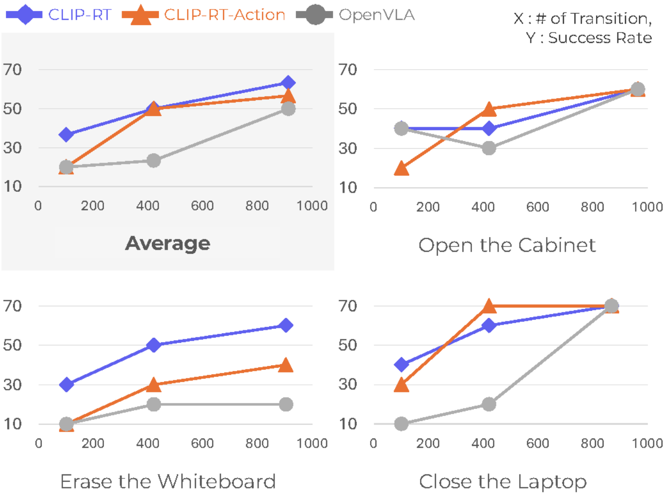


Fig. 6: **Results on few-shot learning.** We report the performance of CLIP-RT, CLIP-RT-Action, and OpenVLA with 1, 5, and 10 demonstrations (from left to right in each graph). The x-axis denotes the number of transitions actuall provided, and the y-axis indicates the task success rate.

Comparison Between Multi-Task and Single-Task Policies. Where does the significant performance gap between CLIP-RT and OpenVLA on Novel tasks come from? One of our hypotheses is that CLIP-RT effectively learns the *shared structure* across diverse robotic tasks by utilizing language-based motion primitives as basic building blocks. To verify this, we train a single-task policy for each Novel task and evaluate the performance of each model. In other words, 9 individual single-task policies for both CLIP-RT and OpenVLA are evaluated. The results are summarized in Figure 5. OpenVLA-Single and CLIP-RT-Single denote the performance of single-task policies for each model. Compared to multi-task policies, both models show performance drops with single-task policies—3.3% drop for OpenVLA and 11.1% drop for CLIP-RT. This suggests that multi-task policy learning benefits both models, but CLIP-RT, with its larger performance gap, benefits more from shared knowledge across tasks. This highlights that CLIP-RT facilitates the learning of more generalizable and transferable policies compared with OpenVLA.

Effect of Natural Language Supervision. In Figure 4, CLIP-RT outperforms CLIP-RT-Action on both Novel and Common tasks. This indicates that the use of natural language supervision also enhances CLIP-RT's generalization capabilities. We visualize the action embeddings of both models to further analyze the impact of natural language supervision in Appendix H.

Few-Shot Generalization. Does CLIP-RT perform effectively with a limited amount of in-domain data? We further investigate this by evaluating learned policies, assuming fewer

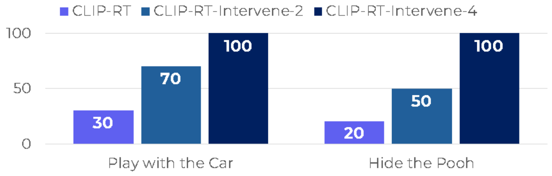


Fig. 7: **Performance on varying numbers of human interventions.** Success rates of two challenging tasks under 0, 2, and 4 human corrections. Each success rate is measured by averaging the results of ten trials.

demonstrations (*i.e.,* 1, 5, and 10) are provided. Specifically, we compare CLIP-RT with OpenVLA and CLIP-RT-Action on three Novel tasks, where all models performs relatively well on average. As shown in Figure 6, CLIP-RT demonstrates improved performance in few-shot policy learning, especially in the single demonstration setting. Such few-shot adaptation is particularly crucial for robotics, where pretraining data (*e.g.,* Padalkar et al. [41]) cannot cover all real-world tasks, necessitating models that can rapidly acquire new skills from minimal demonstrations.

E. Collaborative Capabilities of CLIP-RT

Learning and reasoning about actions in natural language offer an additional benefit: collaborative problem-solving with language-capable entities. In this subsection, we explore how CLIP-RT collaborates with (1) humans by incorporating corrections and (2) large pretrained models via action refinement.
Collaboration with Humans. When CLIP-RT predicts an incorrect motion, humans can easily interpret the predictions and provide a correct motion in a certain state (e.g., "rotate gripper 90 degrees"). We study two tasks in which CLIP-RT achieves its lowest success rates—*Play with the Car* and *Hide the Pooh*—and measure how a small number of human interventions affects performance. We set a maximum limit on the number of corrections per episode humans can provide: 2 and 4. Figure 7 shows the task success rate with varying numbers of human interventions (0, 2, and 4). Without intervention, CLIP-RT's success rates are 30% and 20% on these two tasks. With two interventions, these rates increase to 70% and 50%, and with four interventions, both tasks achieve a 100% success rate. These results demonstrate that even a few human corrections substantially improve CLIP-RT's performance in challenging tasks. Since actions are expressed in language, humans can easily intervene with language corrections

Collaboration with Large Pretrained Models. We also investigate how CLIP-RT can collaborate with a large pretrained model—GPT-4o [24] (GPT for short)—through action refinement. As shown in Figure 8, at each transition, we provide the current image observation and instruction to GPT. GPT then proposes a set of action candidates and labels them as either "appropriate" or "inappropriate". CLIP-RT incorporates this feedback by boosting the scores of actions deemed appropriate and penalizing those labeled as inappropriate. In the example from Figure 8, CLIP-RT initially assigns a high score to "Move arm to the right", but GPT labels this motion as inappropriate and provide positive rewards to the motion, "Move arm to the left", leading to a correct prediction. This GPT-guided approach broadens the range of instructions that CLIP-RT can handle, enabling it to execute instructions that require commonsense knowledge or high-level reasoning. For instance, CLIP-RT can benefit from collaboration with large pretrained models when given instructions like "Stamp below an American corporation founded in July 2003, headquartered in Austin, Texas," as shown in Figure 8. We provide several qualitative examples in Appendix I-B to illustrate how large pretrained models can help perform out-of-distribution instructions requiring commonsense knowledge or complex reasoning. Furthermore, Appendix I-A discusses details about the GPT's text prompt and how exactly GPT's decisions are integrated to CLIP-RT's scores.

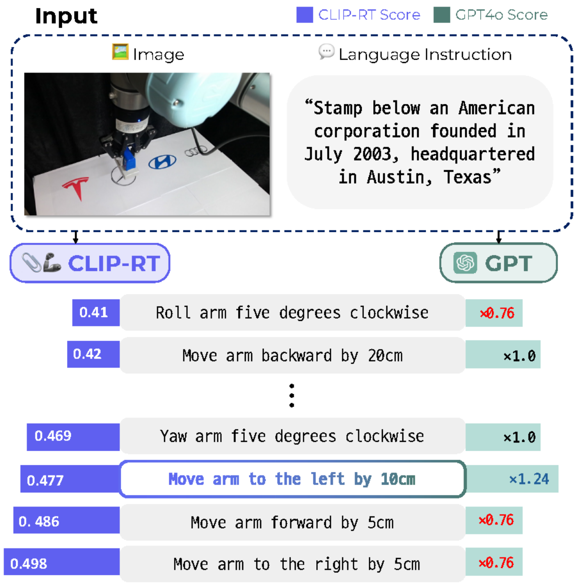
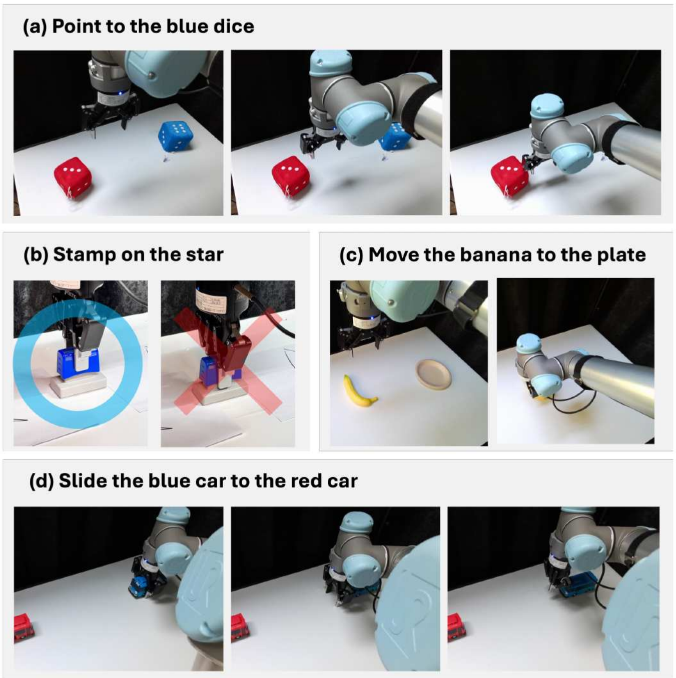

Input
CLIP-RT Score
GPT4o Score
Image
Language Instruction
"Stamp below an American corporation founded in July 2003, headquartered in Austin, Texas"
CLIP-RT
GPT
0.41
Roll arm five degrees clockwise
×0.76
0.42
Move arm backward by 20cm
×1.0
⋮
0.469
Yaw arm five degrees clockwise
×1.0
0.477
Move arm to the left by 10cm
×1.24
0. 486
Move arm forward by 5cm
×0.76
0.498
Move arm to the right by 5cm
×0.76

Fig. 8: **Ensembling CLIP-RT and GPT outputs.** Given an image and language instruction (top), CLIP-RT produces initial scores for candidate actions (left). GPT then supplies multiplicative appropriateness factors for each action (right), which are applied to the CLIP-RT scores to determine the final action, "*Move the arm to the left by 1cm*".

F. Analysis on Failure Cases

We visualize four types of failure cases. First, CLIP-RT has occasionally failed to comprehend the attributes of objects specified in instructions. For example, Figure 9-(a) depicts a scenario in which CLIP-RT is instructed to *point to the blue dice*, but mistakenly pointed at the red dice instead. This examples confirms a need of more precise visual grounding.

Second, CLIP-RT sometimes fail to execute tasks that require fine-grained control, such as *Stamp on* `<obj>`. Figure 9-(b) illustrates an example of such a task. Based on the image observed from the current distance, it may be difficult to precisely determine whether the z-axis of the gripper is



(a) Point to the blue dice

(b) Stamp on the star

(c) Move the banana to the plate

(d) Slide the blue car to the red car

Fig. 9: **Example failure cases of CLIP-RT.** (a) CLIP-RT incorrectly identifies the target, pointing at the red dice instead of the blue dice. It is difficult to detect the correct spatial relationship between the cup and the hanger based on the initial visual input. (b) Failure in executing the "Stamp on the star". The left figure demonstrates a correct grasp of the stamp, whereas the right figure illustrates an incorrect grip that prevents successful task completion. (c) The robot arm completely obstructs the objects of interest, preventing accurate perception and manipulation. (d) The robot slips while attempting to slide the blue car and fails to recover by reopening the gripper and attempting to re-grasp the object.

properly aligned to stamp on <obj>. This limitation is likely due to the reliance on 2D image inputs, which makes it challenging to accurately infer the 3D spatial information necessary for precise manipulation. The models, pretrained on large-scale image-text datasets, may not capture the depth and spatial nuances required for such tasks. Utilizing inputs like RGB-D images or point clouds might alleviate this issue. Third, relying on images from a single viewpoint can lead to occlusions, as visualized in Figure 9-(c), particularly when the robot's arm obstructs the object of interest. Employing multiple camera angles could alleviate this issue by providing a more comprehensive view of the scene.

Fourth, stochastic trajectory augmentation (STA) relies on heuristic algorithms that may not capture the full diversity of possible trajectories. This is particularly evident in scenarios requiring recovery from failure states, such as when an object slips from the gripper, as shown in Figure 9-(d). The heuristics does not adequately represent the multitude of ways a robot might recover or adapt in these situations, potentially hindering the model's ability to generalize to unforeseen circumstances.

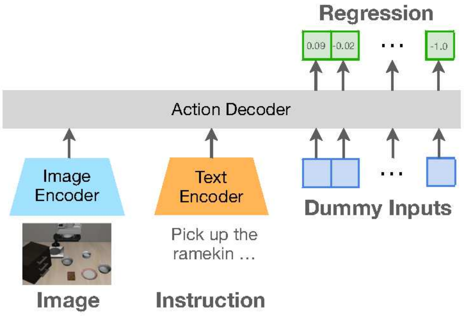


Fig. 10: **Overview of CLIP-RT+ for LIBERO.**

V. Experiments: Adapting CLIP-RT to Simulated Environments

While our primary focus is on training real-world robots through language-guided data collection, we further evaluate CLIP-RT on the LIBERO simulation benchmark [33] to study the following questions:

- **Generality**: Is CLIP-RT applicable to environments with offline, human-teleoperated demonstration data?
- **Performance**: Does CLIP-RT remain effective in a controlled simulation setting?

A. Tasks & Dataset

We evaluate on four task suites of the LIBERO benchmark: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long. These task suites assess policy generalization to diverse spatial relationships, objects, task goals, and long-horizon tasks. Each task suite contains 500 human-teleoperated demonstration data for 10 different tasks. By following the experimental setup in existing studies [30, 29, 21], we train and evaluate CLIP-RT on each task suite individually.

B. Adapting CLIP-RT to the LIBERO Benchmark

Before describing how we adapt CLIP-RT to the LIBERO simulation benchmark, we acknowledge the inherent difficulty of directly representing the fine-grained, continuous human-teleoperated actions in LIBERO using natural language at a comparable level of abstraction. This discrepancy in abstraction levels necessitates the design of an alternative model architecture to enable effective action prediction in this setting. Accordingly, we simply add a 0.3B-parameter action decoder to the original CLIP-RT model to predict continuous actions. We refer to this model as CLIP-RT+. By following Kim et al. [30], we employ action chunking and parallel decoding. As shown in Figure 10, the action decoder takes the image and instruction embeddings vectors from CLIP-RT and zero-valued empty tokens as inputs. We use the L1 regression-based objective to optimize the model. The action decoder shares the same model architecture with the CLIP-RT's text encoder. As a result, CLIP-RT+ is a 1.3B-parameter model. The size of the action chunk is 8, and the dimension of each action is 7. We train CLIP-RT+ using 8 NVIDIA H100 GPUs for 128 epochs with a batch size of 256.

		Inference Efficiency		LIBERO Task Success Rates				
Model	Size	Throughput↑ (Hz)	Latency↓ (Sec)	Spatial↑ (%)	Object↑ (%)	Goal↑ (%)	Long↑ (%)	Average↑ (%)
Octo [38]	93M	-	-	78.9	85.7	84.6	51.1	75.1
DP (scratch) [10]	157M	-	-	78.3	92.5	68.3	50.5	72.4
Dita [21]	334M	-	-	84.2	96.3	85.4	63.8	82.4
OpenVLA [29]	7.5B	4.2	0.240	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT [30]	7.7B	109.7	0.073	**96.2**	98.3	**96.2**	**90.7**	**95.3**
CLIP-RT+ (ours)	1.3B	**163.8**	**0.049**	95.2	**99.2**	94.2	82.6	92.8

TABLE I: **LIBERO task performance and inference efficiency results.** All models, except Diffusion Policy (DP) [10], were fine-tuned. Boldface scores represent the highest score, while underlined scores indicate the runner-up.

C. Results & Discussions

We compare CLIP-RT+ with the state-of-the-art models on the LIBERO simulation benchmarks, including OpenVLA [29], OpenVLA-OFT [30], Dita [21], DP [10], and Octo [38]. As shown in Table I, the recent state-of-the-art VLA model, OpenVLA-OFT [30], achieves the highest average success rate of 95.3%. However, CLIP-RT+ shows comparable performance across all task suites with an average score of 92.8%, while using 6x fewer parameters (1.3B) compared with OpenVLA-OFT (7.7B). Surprisingly, CLIP-RT+ attains a near perfect success rate (99.2%) on the LIBERO-Object task suite, indicating strong generalization to unseen objects in simulation environments. We conjecture that the generalization capabilities of the CLIP model to novel visual categories [45, 17] are successfully transferred to the LIBERO-Object tasks.

We further analyze the inference efficiency of CLIP-RT+. We use two evaluation metrics: (1) throughput (the number of actions predicted per second) and (2) latency (time to predict an action chunk or single action). By following the setup from Kim et al. [30], we measure the throughput and latency on an NVIDIA A100 GPU. As shown in Table I, CLIP-RT+ achieves 39× improved throughput (4.2Hz→163.8Hz) compared with OpenVLA based on its lightweight design and the action chunking technique. When compared to OpenVLA-OFT using the same action chunk size of 8, CLIP-RT+ improves both throughput and latency by approximately 49%.

While LIBERO demonstrations are not compatible with language-based action representations due to their low-level, continuous action space nature, we adapt CLIP-RT by adding a simple action prediction module with an L1 regression objective for continuous action representations. This modification enables us to evaluate the core architectural strengths of CLIP-RT—language-based policy pretraining and lightweight design—on a widely used simulation benchmark (LIBERO). The results demonstrate that CLIP-RT remains effective and generalizable, even when applied beyond the scope of language supervision-based robot learning settings.

VI. Discussion

A. Summary

This paper investigates: (1) how non-experts collect robotic data using natural language supervision and (2) how pretrained vision-language models learn visuomotor policies directly from this supervision. We present CLIP-RT, a new vision-language-action (VLA) model that learns generalizable and transferable policies from natural language supervision. Furthermore, we propose a data collection framework consisting of language-based teleoperation and stochastic trajectory augmentation. Experiments show that CLIP-RT outperforms the state-of-the-art model, OpenVLA by 24%, in acquiring novel manipulation skills, while using 7x fewer parameters. Furthermore, CLIP-RT can collaborate with humans and large pretrained models by using natural language as an interface, improving generalization and decision-making. Finally, we validate the effectiveness of CLIP-RT in simulated environments with offline, human-teleoperated robot data. We believe that our work represents a promising step towards making robot learning more accessible and scalable, enabling non-experts to teach robots directly in their environments.

B. Limitations and Future Work

Inherent Limitations in Human Language Supervision. Human can provide instructions at varying levels of abstraction—from high-level commands like "*Pick up the cup*" to low-level directives such as "*Rotate the second joint by 10 degrees*". Our approach currently assumes that users can offer supervision at an appropriate intermediate level (e.g., *move arm to the right*). This assumption may not hold in real-world scenarios, as non-experts might struggle to calibrate the specificity of their instructions. Addressing this limitation may involve developing adaptive models capable of interpreting instructions across different levels of abstraction or designing a two-stage pipeline that first translates high-level instructions into intermediate commands and subsequently into low-level actions, as demonstrated in [52, 49].

Lack of Temporal Context. Current vision-language-action models, including CLIP-RT, do not predict sequences of actions or consider the history of actions taken. This absence of temporal context limits the models' ability to perform tasks that require an understanding of previous actions or states. For instance, in a task like *Shake the water bottle*, the robot needs to know whether it has already shaken the bottle or how it should continue shaking. Without incorporating action history into the context, the model cannot make informed decisions based on past actions. Future research could explore integrating mechanisms that account for temporal sequences, enabling the model to maintain a memory of prior actions and states, such as hierarchical history encoding [9].

Handling Complex Tasks and Long-Term Planning. The robotic tasks addressed in this paper are relatively short-horizon compared with the complexity and duration of everyday tasks, such as folding laundry [5]. While CLIP-RT successfully demonstrates diverse manipulation skills — such as opening the trash can and closing the laptop — extending these capabilities to long-horizon tasks requires novel approaches that can handle increased task complexity. One promising strategy for long-horizon task execution involves developing a high-level task planner [23, 52, 49] that decomposes complex tasks into sequences of primitive skills. For example, a task planner could break down "set the dinner table" into subtasks like "retrieve plates," "place utensils," and "arrange napkins." Integrating such planners with CLIP-RT's manipulation skills could execute structured, multi-step tasks.

Acknowledgements

We would like to thank Eungseo Kim for his help with robot evaluation in simulated environments. This work was partly supported by the IITP (RS-2021-II212068-AIHub/10%, RS-2021-II211343-GSAI/10%, RS-2022-II220951-LBA/10%, RS-2022-II220953-PICA/15%), NRF (RS-2024-00353991-SPARC/15%, RS-2023-00274280-HEI/10%), KEIT (RS-2024-00423940/10%), KRIT (KRIT-CT-23-003/10%), and Gwangju Metropolitan City (Artificial intelligence industrial convergence cluster development project/10%) grant funded by the Korean government.

References

[1] Pieter Abbeel, Adam Coates, and Andrew Y Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.

[3] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024.

[4] Aude G Billard, Sylvain Calinon, and Florent Guenter. Discriminative and adaptive imitation in uni-manual and bi-manual tasks. *Robotics and Autonomous Systems*, 54 (5):370–384, 2006.

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[8] David Chen and Raymond Mooney. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 859–865, 2011.

[9] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847, 2021.

[10] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[11] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[12] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*, 2023.

[14] Yuqing Du, Ksenia Konyushkova, Misha Denil, Akhil Raju, Jessica Landon, Felix Hill, Nando de Freitas, and Serkan Cabi. Vision-language models as success detectors. *arXiv preprint arXiv:2303.07280*, 2023.

[15] Michael Ahn et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, 2022.

[16] Cem Eteke, Doğancan Kebüde, and Barış Akgün. Reward learning from very few demonstrations. *IEEE Transactions on Robotics*, 37(3):893–904, 2020.

[17] Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander Toshev, and Vaishaal Shankar. Data filtering networks. *arXiv preprint arXiv:2309.17425*,

2023.

[18] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

[19] Jensen Gao, Bidipta Sarkar, Fei Xia, Ted Xiao, Jiajun Wu, Brian Ichter, Anirudha Majumdar, and Dorsa Sadigh. Physically grounded vision-language models for robotic manipulation. *arXiv preprint arXiv:2309.02561*, 2023.

[20] Xingwei He, Zhenghao Lin, Yeyun Gong, Alex Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, Weizhu Chen, et al. Annollm: Making large language models to be better crowdsourced annotators. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, 2024.

[21] Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, et al. Dita: Scaling diffusion transformer for generalist vision-language-action policy. *arXiv preprint arXiv:2503.19757*, 2025.

[22] Yingdong Hu, Fanqi Lin, Tong Zhang, Li Yi, and Yang Gao. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv:2311.17842*, 2023.

[23] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR, 2022.

[24] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[25] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, July 2021. URL https://doi.org/10.5281/zenodo.5143773.

[26] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[27] Gi-Cheon Kang, Junghyun Kim, Jaein Kim, and Byoung-Tak Zhang. Prograsp: Pragmatic human-robot communication for object grasping. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3304–3310. IEEE, 2024.

[28] Junghyun Kim, Gi-Cheon Kang, Jaein Kim, Suyeon Shin, and Byoung-Tak Zhang. Gvcci: Lifelong learning of visual grounding for language-guided robotic manipulation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 952–959. IEEE, 2023.

[29] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[30] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[31] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 259–266. IEEE, 2010.

[32] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020.

[33] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[34] Huihan Liu, Alice Chen, Yuke Zhu, Adith Swaminathan, Andrey Kolobov, and Ching-An Cheng. Interactive robot learning from verbal correction. *arXiv preprint arXiv:2310.17555*, 2023.

[35] Corey Lynch and Pierre Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020.

[36] Guilherme J Maeda, Gerhard Neumann, Marco Ewerton, Rudolf Lioutikov, Oliver Kroemer, and Jan Peters. Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks. *Autonomous Robots*, 41:593–612, 2017.

[37] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

[38] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[39] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[40] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features

without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[41] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[42] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025, 2022.

[43] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[44] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9, 2019.

[45] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[47] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL https://openreview.net/forum?id=M3Y74vmsMcY.

[48] Mingyo Seo, Steve Han, Kyutae Sim, Seung Hyeon Bang, Carlos Gonzalez, Luis Sentis, and Yuke Zhu. Deep imitation learning for humanoid loco-manipulation through human teleoperation. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.

[49] Suyeon Shin, Junghyun Kim, Gi-Cheon Kang, Byoung-Tak Zhang, et al. Socratic planner: Inquiry-based zero-shot planning for embodied instruction following. *arXiv preprint arXiv:2404.15190*, 2024.

[50] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.

[51] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pages 894–906. PMLR, 2022.

[52] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2998–3009, 2023.

[53] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[54] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR, 2020.

[55] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[56] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[58] Ted Xiao, Harris Chan, Pierre Sermanet, Ayzaan Wahid, Anthony Brohan, Karol Hausman, Sergey Levine, and Jonathan Tompson. Robotic skill acquistion via instruction augmentation with vision-language models. In *Proceedings of Robotics: Science and Systems*, 2023.

[59] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

[60] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023.

[61] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5628–5635. IEEE, 2018.