ERRA: An Embodied Representation and Reasoning Architecture for Long-horizon Language-conditioned Manipulation Tasks

Chao Zhao*, Shuai Yuan*, Chunli Jiang, Junhao Cai,
Hongyu Yu, Michael Yu Wang, *Fellow, IEEE,* and Qifeng Chen

***Abstract*—This letter introduces ERRA, an embodied learning architecture that enables robots to jointly obtain three fundamental capabilities (reasoning, planning, and interaction) for solving long-horizon language-conditioned manipulation tasks. ERRA is based on tightly-coupled probabilistic inferences at two granularity levels. Coarse-resolution inference is formulated as sequence generation through a large language model, which infers action language from natural language instruction and environment state. The robot then zooms to the fine-resolution inference part to perform the concrete action corresponding to the action language. Fine-resolution inference is constructed as a Markov decision process, which takes action language and environmental sensing as observations and outputs the action. The results of action execution in environments provide feedback for subsequent coarse-resolution reasoning. Such coarse-to-fine inference allows the robot to decompose and achieve long-horizon tasks interactively. In extensive experiments, we show that ERRA can complete various long-horizon manipulation tasks specified by abstract language instructions. We also demonstrate successful generalization to the novel but similar natural language instructions.**

***Index Terms*—Manipulation, Large Language Model (LLM), Reasoning, Reinforcement Learning, Human-robot interaction**

I. Introduction

If robots are to be widely deployed in workplaces, hospitals, and our homes to assist us, they must understand our needs, discover the underlying causal relations of environments, and interact with the environment appropriately. An example is the case of long-horizon manipulation tasks specified by natural language. For example, when humans hear a request such as "Please put the cosmetic in the drawer", we can simultaneously understand the sentence's semantics and observe the surroundings to determine whether we need to "open the drawer" first or "grasp the cosmetic." We then observe the outcomes of attempted concrete action and plan next. In addition, we can take corrective measures from failure cases (e.g., cosmetic slips from our hands). To operate in our world, robots must replicate such abilities.

This is the motivation for the problem tackled in this paper, which is a robot that has the following abilities: (i) reason abstract nature language instructions and plan with the causal relation of the environment, (ii) develop motor skills to interact with environments, and complete long-horizon manipulation tasks, (iii) detect failures (e.g., accidentally drop an object) and correct them (e.g., grasp the object again). Endowing robots with the combination of abilities (i)-(iii) is a grand challenge because the long-horizon manipulation tasks with abstract language instructions, for example, "clean trash on the table," requires the embodied agent to have semantic knowledge and a reliable interpretation of the environment, to successfully plan and perform a long sequence of motor skills, and to know when to stop (i.e., no trash on the table). While conventional methods, such as symbolic programming or hierarchical reinforcement learning, can plan tasks, most approaches rely on carefully designed representations and analytical transition models, which limit generalization. Recently, a few studies have explored the use of pre-trained large language models (LLMs) to answer questions that require reasoning and planning through prompt design (i.e., hand-crafted text prompts) and utilizing such ability for long-horizon robot manipulation [1]–[3]. However, an important problem with these approaches is that there is no guarantee of what manipulation tasks LLMs can reason about and plan without trying because LLMs lack real-world experience during their original training. Furthermore, small changes in prompts can deteriorate the performance of LLMs, making finding appropriate prompts time-consuming.

To address the above problems and endow robots with abilities (i)-(iii), we propose the ERRA framework based on tightly-coupled probabilistic inferences at two levels of granularity, coarse and fine. An overview of ERRA is shown in Fig. 1. The coarse-resolution inference focuses on high-level reasoning and planning (i.e., what to do in the next step?), and the fine-resolution inference focuses on learning concrete actions (i.e., how to do it?). The results of executing concrete actions in the environment provide feedback for subsequent coarse-resolution inferences. Such coarse-to-fine inferences are invoked repeatedly to decompose long-horizon manipulation tasks as a sequence of concrete actions. Coarse-resolution inference is built on a pre-trained large language model for generating the action language. Motor skills in

Manuscript received: December, 5, 2022; Revised February, 28, 2023; Accepted March, 29, 2023.

This paper was recommended for publication by Editor Hong Liu upon evaluation of the Associate Editor and Reviewers' comments.

*Authors with equal contribution. This work was supported by grants from the Innovation and Technology Commission (project: ITS/036/21FP) of HKSAR. C. Zhao, S. Yuan, C. Jiang, J. Cai, H. Yu, and Q. Chen are with The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong {czhaobb, syuanaf, cjiangab, jcaiaq}@connect.ust.hk and {hongyuyu, cqf}@ust.hk. J. Cai, H. Yu, and M. Wang are also with HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen. M. Wang is with Monash University michael.y.wang@monash.edu

Digital Object Identifier (DOI): see the top of this page.

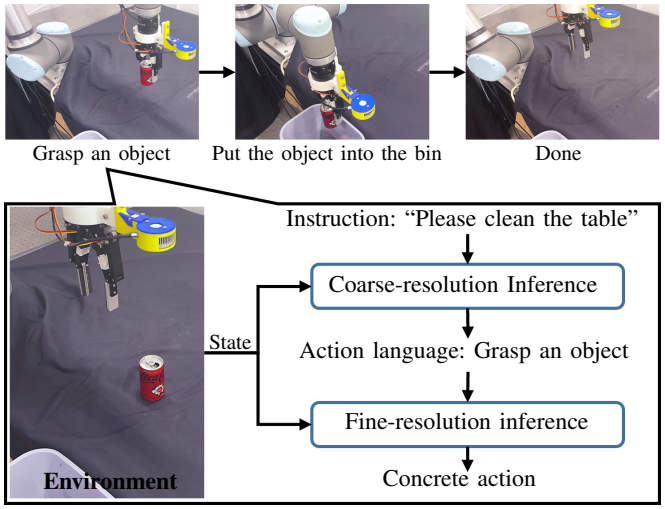

Fig. 1: **ERRA Overview.** The image sequence at the top shows the action language and execution process to complete a task with the instruction "Please clean the table," utilizing ERRA. Given a language instruction, the coarse-resolution inference produces the next step represented by action language, according to the environment state. The action language and state are then processed by fine-resolution inference, which outputs the concrete action to interact with the environment.

fine-resolution inference are learned through reinforcement learning (RL) under self-supervision.

The primary contribution of this work is to suggest a new approach, ERRA, that allows an embodied agent to acquire reasoning, planning, and interaction abilities for solving long-horizon manipulation tasks specified by natural language. Extensive experiments show that ERRA is capable of understanding the semantics in abstract language instructions, reasoning in environments with rich functional relationships between objects, and providing motor skills to complete long-horizon manipulation tasks. We also show that ERRA allows the robot to recover from failure cases and adapt to environmental changes in the real world, significantly improving the robustness of robots in dynamic environments.

II. Related Work

Task and Motion Planning. In robotics, task and motion planning [4] is capable of solving long-horizon (i.e., multi-step) manipulation tasks. Traditional methods rely on symbolic planning [5] or optimization [6] in abstract or symbolic spaces. However, most approaches require manually defined representation spaces and environment kinematics models, which are usually domain-specific and lack generalization ability. More recently, LLMs have demonstrated dawning properties on reasoning and planning under appropriate conditions (e.g., language prompts) [7]–[10]. Several works [3] have studied using LLMs to plan robot manipulation tasks. SayCan [1] uses LLM to infer the entire plans of the manipulation task and estimate the feasibility of each step using a model of action affordance. However, these methods assume that the execution of each planned motor skill is faultless, making them not robust to intermediate failures in task execution. In this aspect, [2] introduces additional modules to incorporate human and environmental feedback to improve the completion of tasks. While prior works have investigated how LLMs plan via prompt design, the ability of LLMs is agnostic, requiring time and human effort to design and experiment with different prompts. We introduce the prompt tuning method [11], enabling the LLM to be a reasoner and planner without using design prompts.

Learning Language-Conditioned Manipulation. Natural language provides a human-interactive interface to link humans to robots, which is important for deploying robots in our lives. Many studies [12]–[16] have explored how robots follow language instructions, in which robots are required to complete tasks specified by the language. Some studies [17]–[19] have learned language-conditioned behaviors through imitation learning. For example, [20] learns a direct mapping from images and natural language instructions to actions using a Transformer network. [21] uses an offline robotics dataset with crowdsourced natural language labels to learn a range of vision-based manipulation tasks. Most of these works focus on learning short-horizon manipulation tasks such as grasping or in-hand manipulation. In contrast, ERRA can understand instructions with abstract semantics and achieve long-horizon tasks by leveraging LLMs' semantic knowledge to interpret instructions and plan tasks.

Reinforcement Learning for Manipulation. Reinforcement learning combined with deep learning has recently made extensive progress in learning skills in different domains, such as beyond human experts at the games of Go [22] and Atari [23]. In robotic manipulation, reinforcement learning offers the robot a way to acquire various manipulation skills through self-exploration [24]–[26]. However, most studies focus on learning narrow and individual tasks. Some works achieve long-horizon task planning by hierarchical reinforcement learning [27], [28], which requires manual task-level design and lacks generalization ability. In our work, ERRA leverages reinforcement learning to acquire low-level motor skills in the simulation and cooperates with the coarse-resolution inference module to perform long-horizon manipulation tasks.

III. Method

In this section, we describe the architecture of ERRA, as shown in Fig. 2. ERRA is based on two inference modules, coarse and fine. The coarse-resolution module infers an action language (e.g., grasp the apple) based on language instruction, environment state, and robot proprioception. The action language corresponds to a motor skill that the robot needs to execute. Subsequently, the fine-resolution inference module generates concrete actions for executing the motor skill, using inputs of the action language inferred by the coarse-resolution inference module and the visual and tactile information. By iteratively invoking the coarse-to-fine inference process, a task can be decomposed into simpler concrete actions and executed. The step-by-step planning and execution processes of ERRA enable feedback to be established and mitigate the challenges of reasoning and planning, resulting in effective and robust performance. In

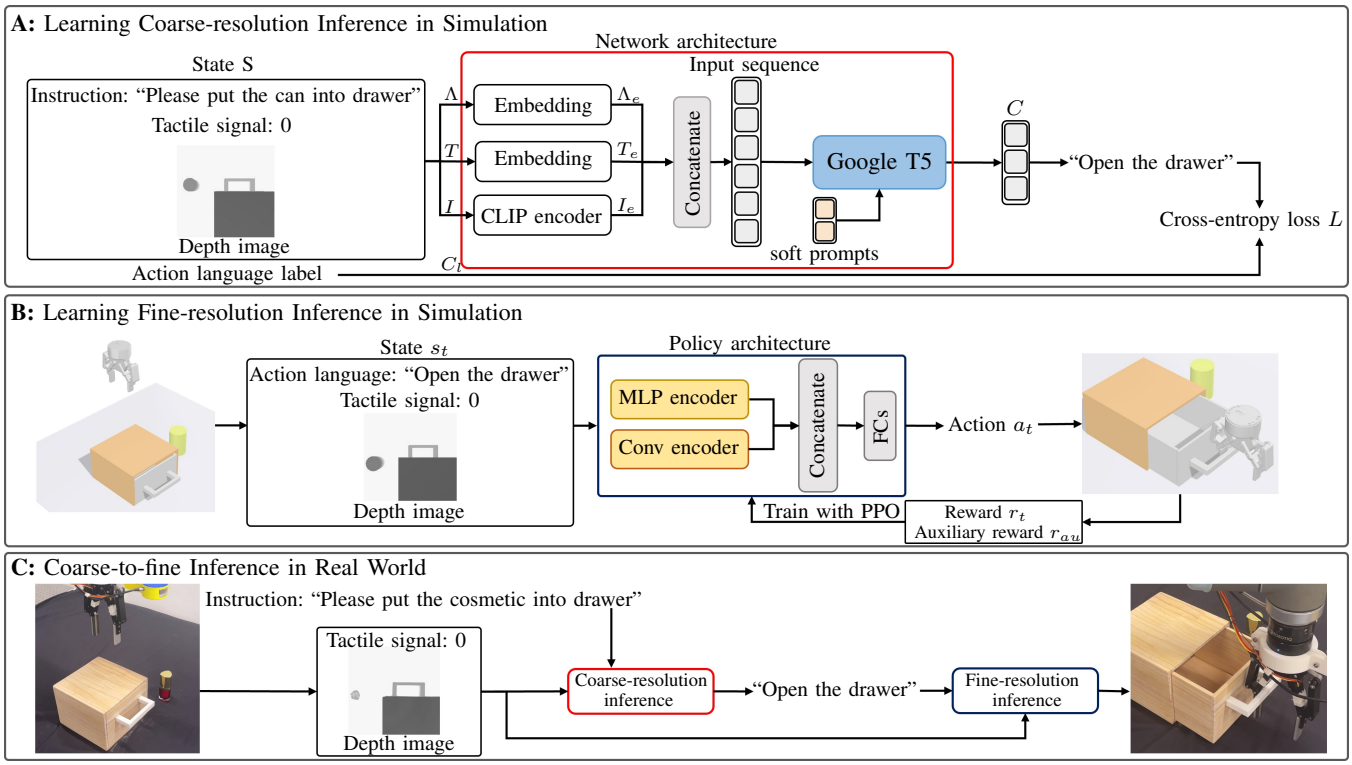

Fig. 2: **System Overview. A:** We generate a set of correspondences (S, C) in the simulation to learn the coarse-resolution inference. State S includes the instruction Λ, depth image I, and tactile signal T. The provided inputs are encoded as three vectors (Λ_e, T_e, I_e), respectively. These vectors are concatenated and subsequently fed to Google T5. At last, the output action language C and label of action language C_l are used to compute loss; **B:** To learn fine-resolution inference, we employ PPO. The RL agent takes the state s_t as input and predicts the action a_t for the robot execution at time step t. The agent then obtains rewards r_t and auxiliary reward r_{au} from the simulation; **C:** We deploy the ERRA in the real world. Given instruction, the coarse-resolution module infers the action language based on current observation and the tactile signal. Then the fine-resolution module predicts actions with the inputs of action language and environment state.

the following sections, we describe the problem formulation of the coarse and fine-resolution inferences and elucidate the supervised learning and reinforcement learning approaches we adopted to train these two inference modules in the simulation.

A. *Coarse-resolution Inference*

The objective of learning the coarse-resolution inference is to obtain a high-level manipulation planning strategy, as shown in Fig. 2A. The coarse-resolution inference is formulated as a sequence-to-sequence text generation task, in which the generated action language guides fine-resolution inference to predict concrete actions performed by the robot.

Problem Formulation: Formally, it is a mapping $p : S \rightarrow C$, where $S = (\Lambda, I, T)$ is the input state and $C = (c_1, c_2, ..., c_m)$ is a sequence of text that represents the action language. The state S consists of three parts: a language instruction $\Lambda = (\lambda_1, \lambda_2, ..., \lambda_n)$, which is a sequence of text words; a depth image I taken by the camera in the environment; and a tactile signal T (i.e., a binary signal indicating the presence or absence of objects between fingers). The coarse-resolution inference is constructed as a neural network that predicts each word $c_i \in C$ given the state S.

Learning Coarse-resolution Inference: To learn the coarse-resolution inference, we generate a synthetic dataset $D = (d_1, d_2, \dots)$. The dataset is collected in simulation leveraging the pre-programmed environments for various language-conditioned manipulation tasks. Each piece of data d_i contains a corresponding relationship: at the current state S, what needs to do next (denoted as $C_l = (c_{l1}, ..., c_{lm})$). For instance, in the first example of Fig. 3(a), the data $d_i \in D$ consists of the language instruction $\Lambda = \{Please\ put\ the\ cosmetic\ into\ the\ drawer\}$, the tactile signal $T = 0$, the depth image I, and the label of the action language $C_l = \{Open\ the\ drawer\}$.

We choose Google T5 [29], a large-scale pre-trained language model, as the backbone of the network, which provides benefits of better contextual understanding and generalization ability for language. As shown in Fig. 2A, the model extracts the information from the language instruction, tactile signal, and image. Then, the model reasons the next action language is "Open the drawer," based on the available information while rejecting other possibilities, such as "Grasp the cosmetic." Operationally, we encode the image I with an image encoder from CLIP [30] as a dense vector I_e. The binary tactile signal T is transformed to a random initialized dense vector $T_e \in R^n$, with the same dimension as the embedding vectors in T5. Subsequently, image embedding I_e and tactile embedding T_e are concatenated with the word embedding Λ_e of the given instruction Λ as the input sequence (Λ_e, T_e, I_e), as shown in Fig. 2A. Given such input, the model is expected to generate the appropriate action language C after training.

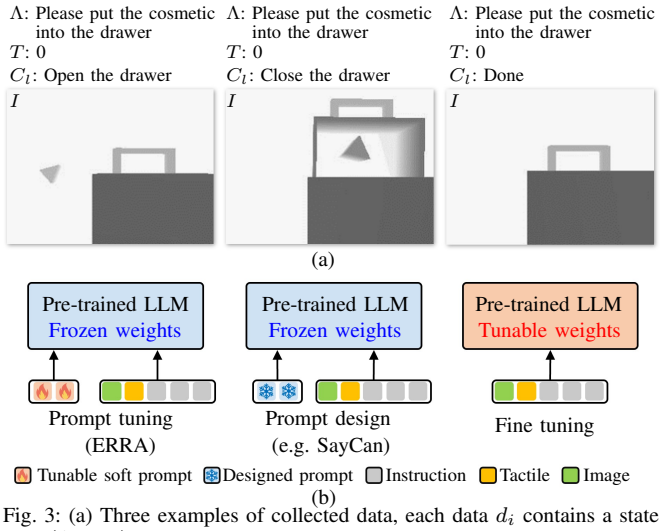

Fig. 3: (a) Three examples of collected data, each data d_i contains a state $S = (\Lambda, I, T)$, and a label of the action language C_l; (b) Difference between our training and others. Instead of adjusting the parameters of LLM or using engineered prompts, our method introduces prompt tuning, which adds a small set of learnable soft prompts and shares the frozen LLM across all tasks.

The network is trained with soft-prompt tuning [11]. Conventional prompt tuning methods freeze all parameters of the pre-trained language model and use a language prompt to probe it to downstream tasks [11]. In soft-prompt, the prompt is replaced by a group of trainable dense vectors, which avoids manually designing prompts and reduces the number of training parameters. Fig. 3(b) shows the difference between our training method and alternatives. We add relevant soft prompts before the transformer layer of T5 to control the behavior of the LLM. Soft prompts are parameterized by using a two-layer feed-forward neural network. During training, we keep the language model parameters constant and only fine-tune the parameters related to these soft prompts. We learn the model with the language model loss:

$$L = -\sum_{i=1}^{m} \log P(c_i = c_{li} | c_{<i}, \Lambda, I, T), \tag{1}$$

where m is the sequence length of C, $c_i \in V$ is the i^{th} word in sequence C, c_{li} is the i^{th} word of sequence C_l, and V is the vocabulary.

B. Fine-resolution Inference

The fine-resolution inference aims to link the action language inferred by the coarse-resolution module to the concrete action that enacts it. The fine resolution module outputs the action parameters of the gripper for the robot to execute, to realize the motor skills corresponding to the action language. These motor skills are limited to four degrees of freedom in order to simplify collision calculations and motion planning.

Problem Formulation: We formulate the problem of learning fine-resolution inference as a Markov Decision Process (MDP). An MDP comprises state space S', action space A, a reward function $R(s_t, s_{t+1})$, and transition probability $P(s_{t+1}|s_t, a_t)$. The RL aims to discover an optimal policy π that selects action a_t to maximize cumulative rewards.

Learning Fine-resolution Inference: We model the policy as a categorical model corresponding to a discrete-domain stochastic policy. The policy is trained with proximal policy optimization (PPO). At time step t, the agent chooses an action a_t according to the probability output by the policy $\pi(a_t|s_t)$, and receives a reward r_t from the environment.

The state is represented by a tuple $s_t = (I, L_e, T)$, where I is the initial depth image of the environment with a resolution of 240×320, T is a binary signal from the tactile sensor on the finger, and L_e is the embedded vector of the inferred action language. L_e and T are concatenated as a vector $g_t = (L_e, T)$.

The policy network architecture comprises a convolutional (Conv) block and a multilayer perceptron (MLP) block, as shown in Fig. 2B. The depth observation I and g_t are embedded into two latent vectors by the Conv block and MLP block, respectively. The resultant vectors are then concatenated and passed to the fully-connected layers (FCs) to produce an output action.

The action a_t consists of two components, namely gripper pose displacement and gripper closure. The gripper pose displacement is constructed as the difference between the current and desired pose of the gripper. It is formed as $(x_t, y_t, z_t, \alpha_t)$, where (x_t, y_t, z_t) represents the relative displacement of the gripper in the workspace, and α_t represents the gripper's rotation about its z-axis. The displacement of the z-axis is executed last by the robot, and we fix the target gripper height during action execution to facilitate learning. The gripper closure control is represented by a one-hot vector β_t is a one-hot vector that the gripper will be closed if $\beta_t = 1$. Thus, the full action is defined as $a_t = (x_t, y_t, z_t, \alpha_t, \beta_t)$, and we discretize each action coordinate according to the workspace. During operation, the robot initiates its movement along the x and y axes before proceeding to the z-axis.

The reward r_t is given at the end of an episode, 1 for successfully completing the required motor skill and 0 otherwise. In addition, we also provide a linear auxiliary reward that encourages the robot to approach the target position. The auxiliary reward r_{au} varies from 0 to 1 based on the distance between the gripper and target positions (the closer, the higher).

C. Training Details

Coarse-resolution and fine-resolution inferences are learned in the Pybullet simulator [31] and then transferred to the real world. In the coarse-resolution inference learning stage, we generate 300 examples per task in the simulation for 17 language-conditional manipulation tasks, as shown in Fig. 5. Specifically, we vary a task from the following three aspects: object types, object positions, and initial setups. For example, in the "Put something into the drawer" task, we consider different objects, such as cosmetics or cans, and randomly position them within the workspace. Additionally, we have two different initial setups where the drawer is either open

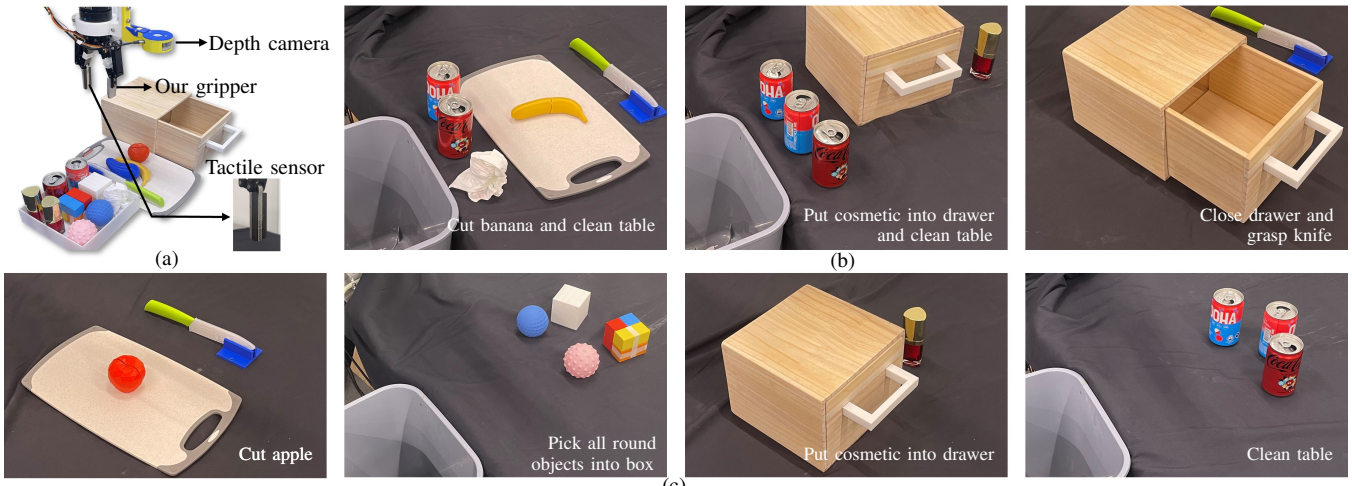

Fig. 4: **Hardware and Scene Setup.** (a) Robot hardware and objects in real-world experiments; (b) Three scene setups for the hybrid tasks; (c) Four scene setups for the Long-horizon tasks. Short-horizon tasks also use these scenes. An example task is shown at the bottom right in each scene setup.

TABLE I: task family and language instruction definitions

Task Family	Num	Task Explanation	Instruction Type	Example Instruction	Example Task
Short-horizon	10	Tasks that require one reasoning step completed by a single motor skill	Straight	"Please grasp the apple"	Robot needs to grasp the apple
Long-horizon	4	Tasks that require many reasoning steps completed by a range of motor skills	Abstract	"Please clean the table"	Robot needs to pick up all trash on the table into the bin
Hybrid	3	Combined long-horizon and short-horizon tasks	Abstract	"Please put the apple into the drawer and clean the table"	Robot needs to place the apple and clean all trash

or closed. The generated synthetic dataset D contains over 12000 corresponding relationships and is used to train the coarse-resolution inference with cross-entropy loss. We follow the implementation in [11] for the soft-prompt tuning. We use the Adam optimizer [32] and a linear learning rate scheduler during training. A default setting trains for ten epochs and uses a learning rate of 5×10^{-5}. Fig. 4 shows seven scene setups, and we also create similar ones in the simulation. In the real world, we deploy ERRA on a UR10 arm equipped with a parallel gripper, an Intel L515 depth camera, and a tactile sensor, as shown in Fig. 4(a).

To learn the fine-resolution inference, 32 robots in simulation environments collect training episodes by obtaining the current policy from the optimizer every eight epochs. In each environment, a manipulation task specified by an action language is procedurally generated, which is randomly selected from substeps in 17 language-conditional manipulation tasks and applies the same random variations as during data collection in the coarse-resolution inference learning stage. The robot in the simulation environment then collects episodes, during which the reward is automatically determined based on whether the task is completed. If the robot completes the task, the environment will be reset, and a new task will be generated again. At last, the collected episodes are returned to the optimizer for learning the policy. During the training, we use Adam optimizer [32] with a learning rate of 10^{-4}. We also randomize the object's physical properties during the task generation and add noise to the depth observation to make the learned policy robust to the various conditions in the real world. Specifically, the object size undergoes a global scaling, which entails resizing object dimensions within a range from 85% (min) to 115% (max) of its original size. Meanwhile, the spatial and visual properties are impacted by adding noise to camera properties. Specifically, we add noise to the camera position, camera pointing position, and field of view. The camera position and pointing are perturbed using three-dimensional vectors, and random noise of each dimension is sampled from a range $\{-2.5\ mm, 2.5\ mm\}$. Similarly, the field of view is perturbed with a noise range of $\{-0.025°, 0.025°\}$. Supplement materials are available at: https://robotll.github.io/ERRA/

IV. Experiments

We design a set of experiments in both simulation and real-world to evaluate the ERRA and other baselines in the language-conditioned manipulation tasks. The hypotheses we want t o validate are as follows:

H1: ERRA can perform long-horizon language-conditioned manipulation tasks and outperforms other baselines.
H2: Robot proprioception is important for completing language-conditioned manipulation tasks.
H3: LLMs with prompt-tuning allow ERRA to generalize to unseen natural language instructions.
H4: ERRA is able to transfer to the real world.
H5: ERRA can respond to environmental changes caused by humans or its own failures.

A. Scenes, Tasks and Evaluation Setup

Scene and task setup: Fig. 4 shows seven scene setups, and we also create similar ones in the simulation. Our

TABLE II: simulation experiments

Method	Short-horizon		Long-horizon		Hybrid		Total	
	Plan*	Task**	Plan	Task	Plan	Task	Plan	Task
Infer-all	**100%**	**94%**	55%	46%	52%	31%	69%	57%
ERRA-w/o touch	83%	79%	48%	41%	42%	35%	58%	52%
ERRA	**100%**	**94%**	**91%**	**81%**	**77%**	**64%**	**89%**	**80%**

* *Plan success rate.* ***Task success rate.*

hardware settings in the real world are also shown in Fig. 4(a). To evaluate ERRA, we test its performance on 17 language-conditioned manipulation tasks from seven scenes in both simulation and real-world. These tasks cover time horizons, language complexity, and variations over the robot and environment. Tab. I details examples for each task family, which fall into the following:

- **Short-horizon**: Short-horizon tasks are decomposed from long-horizon tasks, which involve a straight language instruction that needs to be achieved by a single motor skill. The instruction and the action language have a one-to-one correspondence in such tasks.
- **Long-horizon:** Tasks are specified by abstract natural language instruction and achieved by a long sequence of motor skills. The correspondence between the language instruction and the action language is not one-to-one and is affected by the environment and robot state. This tests the ERRA's ability to reason abstract instructions and to plan with the environment's causal relation.
- **Hybrid:** These tasks are the combination of multiple long-horizon and short-horizon tasks, which have a higher complexity than others.

Baseline comparisons: We compare with the following approaches:

- **Infer-all:** It is similar to the architecture of SayCan [1], in which all action languages are inferred together, and then the robot executes them one by one without feedback during the entire task planning and execution process.
- **ERRA-w/o touch:** An ablated version of the ERRA without the proprioceptive input (i.e., tactile information). Both coarse-resolution and fine-resolution modules only use the camera to observe environments.
- **ERRA:** We deploy the ERRA system to the robot, which is the full non-ablated method we propose in this article.

Metric: We consider two evaluation metrics: *plan success rate* (successful task planning/total attempts) and *task success rate* (completed tasks/total attempts) for validating performance. The *plan success rate* is measured by whether the module of coarse-resolution inference correctly predicts all action languages in a language-conditioned manipulation task, assuming that the execution of motor skills is flawless. The *task success rate* is calculated based on whether the target manipulation task is completed. It requires the coarse-resolution module to successfully plan each step and the fine-resolution module to output the correct actions for the robot to complete corresponding motor skills. For each task, we repeat the test 500 times in simulation experiments and ten times in real-world experiments.

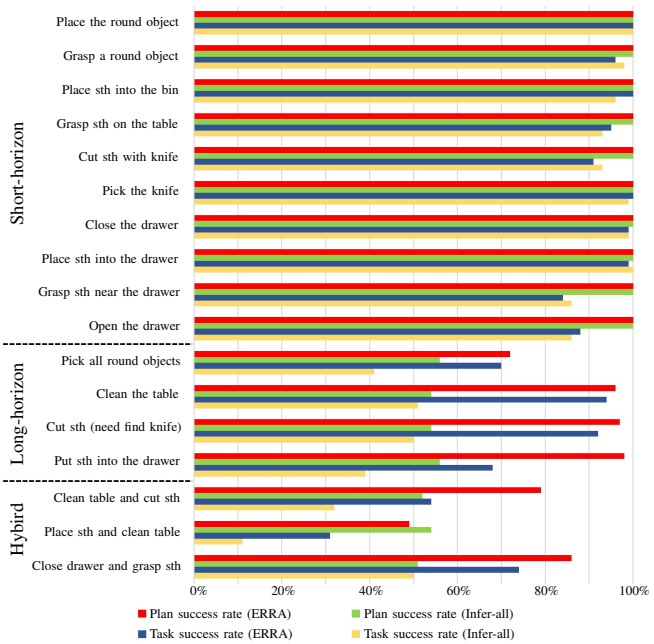

Fig. 5: **Task performance in the simulation.** From top to bottom, there are 14 short-horizon tasks, four long-horizon tasks, and three hybrid tasks.

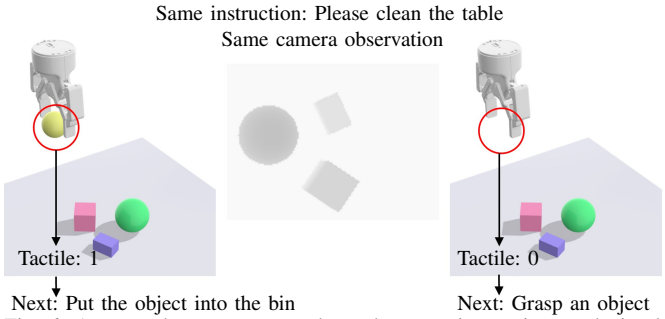

Fig. 6: A case where two scenes have the same instructions and visual observations but different next plans. Language and visual information are insufficient to plan the next step without tactile signals.

B. Simulation Results

Comparison to baselines: Tab. II shows the performance of ERRA on different task families in the simulation. Across all tasks, ERRA achieves a plan success rate of 89% and a task success rate of 80%. In the Long-horizon and Hybrid families, ERRA achieves 91% and 77% plan success rates, respectively. Such results highlight the effectiveness of coarse-resolution inference in enabling ERRA to reason and plan for tasks with longer horizons. The plan and task success rate for each task is fully illustrated in Fig. 5.

To demonstrate the importance of incorporating robot proprioception, we conduct an ablation experiment by excluding the tactile information from inputs during ERRA training (denoted as ERRA-w/o touch in Tab. II). The results show that the planning performance of ERRA -w/o touch is reduced by up to 43% on the Long-horizon family and

TABLE III: generalization to unseen language instructions

Type	Short-horizon		Long-horizon		Hybrid		Total	
	Plan	Task	Plan	Task	Plan	Task	Plan	Task
Unseen Verb	99%	94%	77%	68%	51%	42%	76%	68%
Unseen Noun	100%	94%	80%	72%	55%	40%	78%	69%
Unseen Verb + Noun	99%	94%	52%	47%	34%	25%	62%	55%

35% on the Hybrid family. This decline is attributed to the incomplete information required for reasoning. In certain tasks, the relationship between state and action language is not one-to-one, thereby rendering reasoning impossible. An example of such a scenario is presented in Fig. 4.

We then investigate the effectiveness of the coarse-to-fine inference design in the ERRA by comparing the ERRA with an architecture in which the planning and execution are independent (denoted as infer-all in Tab. II). Our results reveal that ERRA outperforms the Infer-all by over 25% on the Long-horizon and Hybrid families. Infer-all's suboptimal performance is due to its reliance on accurately inferring all action language before executing corresponding motor skills, which increases the difficulty of reasoning and planning. In contrast, the ERRA utilizes closed-loop feedback by inferring the next step only after the robot executes the previous step, leading to more effective and robust task performance.

Generalization to unseen language instructions: We study the ERRA's generalization ability to unseen natural language instructions. Specifically, we test the generalization of ERRA at three levels of rephrased language instructions with novel but similar words. First, we replace nouns in the language instructions of tasks (e.g., "cut the banana" to "chop the banana"), denoted as Unseen Verb in Tab. III. Second, we replace verbs (e.g., "grasp the cola" to "grasp the can"), denoted as Unseen Noun in Tab. III. Finally, we replace both nouns and verbs in instructions(e.g., "close the drawer" to "shut the cabinet"), denoted as Unseen Verb + Noun in Tab. III.

As shown in Tab. III, ERRA's generalization performance degrades as the complexity of the task and the number of unseen words in the instruction increase. Specifically, in the Long-horizon family, changing either the verbs or nouns leads to a 12% decrease in plan success rate, while changing both results in a 40% decline. The decline in planning performance also leads to a corresponding decrease in task success rate. In contrast, ERRA remains stable in the Short-horizon family, owing to the lower complexity of language abstraction and task planning. Notably, we observe that the performance of the Hybrid family drops by up to 35% on novel instructions due to the challenging language instructions and task planning involved. In conclusion, our findings suggest that the ERRA can generalize to novel language instructions, but its generalization performance is affected by the complexity of the task and the number of unseen words in the instruction.

C. Real-world Experiments

We also evaluate ERRA's performance in the real world. ERRA achieves an average task success rate of 77% across three task families. The results show that the model's reasoning, planning, and interaction abilities are transferable to real-world scenes for solving long-horizon language-conditioned manipulation tasks. Similar to the simulation results, ERRA performs best (89% success rate) on Short-horizon tasks among the three task families. The performance of ERRA decreases as task complexity increases, achieving a success rate of 75% on the Long-horizon family and 68% on the Hybrid family. The running time for one step in the task is approximately 12 seconds, which includes the entire cycle time from network inferences (less than 0.2 s) to robot execution.

TABLE IV: real world results

Method	Task Success Rate			
	Short-horizon	Long-horizon	Hybrid	Total
ERRA	89%	75%	68%	77%

Looking back to our initial example in Sec. I, "Please put the cosmetic in the drawer," we have demonstrated that ERRA is able to discover whether the robot needs "Open the drawer" by reasoning the causal relation of the environment (See Fig. 7A) and plan and execute a long sequence in the real world, which include opening the drawer, grasping the cosmetic, putting the cosmetic into the drawer and then closing the drawer. Note also that the robot only has one arm, ERRA necessitates planning the action in reasonable order (e.g., first, open the drawer and then grasp the cosmetic, not the other way around). This requires the ERRA to have strong abilities of long-horizon reasoning and understanding of semantic knowledge in the language instruction.

As shown in Fig. 7, ERRA manifests robustness to dynamic environments. ERRA discovers a new round object added by the human after the last object has been put in the bin and correctly infers that the next action language is "grasp a round object" rather than "Done" (See Fig. 7C). Such behavior is powered by itself, benefiting from the closed-loop feedback provided by the coarse-to-fine inference architecture. Such feedback also allows the robot interactively recover from failure cases. Fig. 7B shows the ERRA response to its failure (object slip from hand during the task execution).

V. Conclusion, Limitation, and Future Work

We have presented a novel solution, ERRA, that utilizes tightly-coupled probabilistic inferences at two granularity levels, coarse and fine, for solving long-horizon language-conditioned manipulation tasks. Through coarse-to-fine inferences, complex manipulation tasks can be decomposed into concrete actions and executed by the robot. Extensive controlled experiments demonstrate the robustness and effectiveness of ERRA on manipulation tasks with long-horizon and abstract semantics. Our work is not without limitations; first, limited by hardware devices, the robot's position is fixed without the need for localization and mapping, suggesting exciting opportunities for extending the current work to the scene of mobile robots. Future research can also benefit from

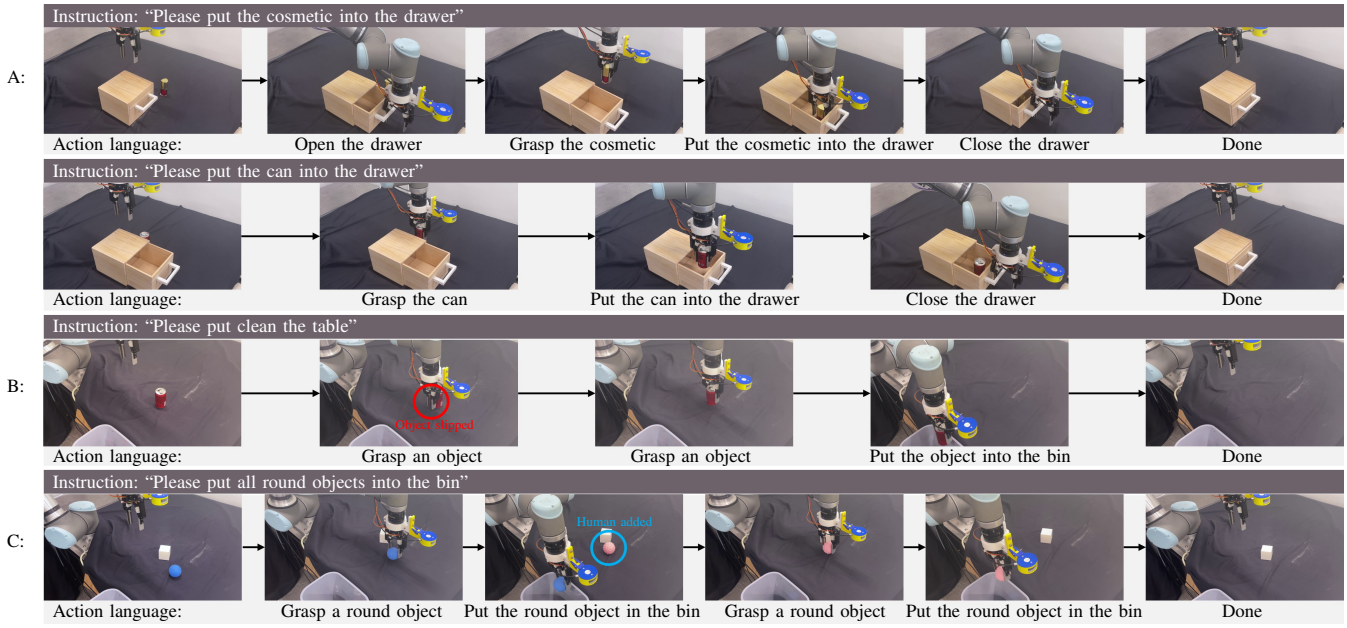

Fig. 7: **Qualitative results of ERRA. A:** Sequences of the robot successfully placing the object into the drawer at different settings (i.e., drawer is closed or open); **B:** A sequence of the robot successfully recovering from its execution failure (i.e., object slip from gripper) and complete table cleaning; **C:** A sequence of the robot adapting to the dynamic environment. (i.e., human places another round object).

the flexibility and efficiency of a dual-arm system. While planning for the dual-arm system may involve evaluating a larger number of potential actions, the increased flexibility and redundancy provided by the additional arm may result in more optimal final plans, compared to our single-arm system. Finally, the proposed work relies on simulated data to learn inference at both coarse and fine resolutions, which is a significant advantage that avoids a more time-consuming process, such as manual labeling. However, it still needs to build simulation scenes carefully. One possible opportunity is to develop a method that is able to learn from online videos in which humans perform manipulation tasks with long-term and abstract semantics.

REFERENCES

[1] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *6th Annual Conference on Robot Learning*, 2022.

[2] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, T. Jackson, N. Brown, L. Luu, S. Levine, K. Hausman, and brian ichter, "Inner monologue: Embodied reasoning through planning with language models," in *6th Annual Conference on Robot Learning*, 2022.

[3] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," *arXiv preprint arXiv:2201.07207*, 2022.

[4] A. Akbari, Muhayyuddin, and J. Rosell, "Knowledge-oriented task and motion planning for multiple mobile robots," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 31, no. 1, pp. 137–162, 2019.

[5] M. Sridharan, M. Gelfond, S. Zhang, and J. Wyatt, "Reba: A refinement-based architecture for knowledge representation and reasoning in robotics," *Journal of Artificial Intelligence Research*, vol. 65, pp. 87–180, 2019.

[6] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," Robotics: Science and Systems Foundation, 2018.

[7] 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," *arXiv preprint arXiv:2210.03094*, 2022.

[8] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022. Survey Certification.

[9] D. Shah, B. Osiński, S. Levine, *et al.*, "Robotic navigation with large pre-trained models of language, vision, and action," in *6th Annual Conference on Robot Learning*.

[10] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[11] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

[12] Y. Chen, R. Xu, Y. Lin, and P. A. Vela, "A joint network for grasp detection conditioned on natural language commands," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4576–4582, IEEE, 2021.

[13] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13139–13150, 2020.

[14] C. Wang, C. Ross, Y.-L. Kuo, B. Katz, and A. Barbu, "Learning a natural-language to ltl executable semantic parser for grounded robotics," in *Conference on Robot Learning*, pp. 1706–1718, PMLR, 2021.

[15] V. Blukis, R. Knepper, and Y. Artzi, "Few-shot object grounding and mapping for natural language robot instruction following," in *Conference on Robot Learning*, pp. 1829–1854, PMLR, 2021.

[16] W. Liu, C. Paxton, T. Hermans, and D. Fox, "Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 6322–6329, IEEE, 2022.

[17] 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*, pp. 991–1002, PMLR, 2022.

[18] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg, "Concept2robot: Learning manipulation concepts from instructions and human demon-

strations," *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1419–1434, 2021.
[19] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13139–13150, 2020.
[20] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," *arXiv preprint arXiv:2209.05451*, 2022.
[21] S. Nair, E. Mitchell, K. Chen, S. Savarese, C. Finn, *et al.*, "Learning language-conditioned robot behavior from offline data and crowd-sourced annotation," in *Conference on Robot Learning*, pp. 1303–1315, PMLR, 2022.
[22] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.
[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[24] H.-G. Cao, W. Zeng, and I.-C. Wu, "Reinforcement learning for picking cluttered general objects with dense object descriptors," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 6358–6364, IEEE, 2022.
[25] C. Zhao, Z. Tong, J. Rojas, and J. Seo, "Learning to pick by digging: Data-driven dig-grasping for bin picking from clutter," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 749–754, IEEE, 2022.
[26] C. Zhao and J. Seo, "Learn from interaction: Learning to pick via reinforcement learning in challenging clutter," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022.
[27] D. Hafner, K.-H. Lee, I. Fischer, and P. Abbeel, "Deep hierarchical planning from pixels," in *Advances in Neural Information Processing Systems* (A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, eds.), 2022.
[28] F. Xia, C. Li, R. Martín-Martín, O. Litany, A. Toshev, and S. Savarese, "Relmogen: Integrating motion generation in reinforcement learning for mobile manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4583–4590, IEEE, 2021.
[29] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer.," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.
[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, pp. 8748–8763, PMLR, 2021.
[31] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning." http://pybullet.org, 2016–2021.
[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.