SAM-E: Leveraging Visual Foundation Model with Sequence Imitation for Embodied Manipulation

Junjie Zhang [1 2] **Chenjia Bai** [2] **Haoran He** [3 2] **Zhigang Wang** [2] **Bin Zhao** [2] **Xiu Li** [1] **Xuelong Li** [2 4]

Abstract

Acquiring a multi-task imitation policy in 3D manipulation poses challenges in terms of scene understanding and action prediction. Current methods employ both 3D representation and multi-view 2D representation to predict the poses of the robot's end-effector. However, they still require a considerable amount of high-quality robot trajectories, and suffer from limited generalization in unseen tasks and inefficient execution in long-horizon reasoning. In this paper, we propose *SAM-E*, a novel architecture for robot manipulation by leveraging a vision-foundation model for generalizable scene understanding and sequence imitation for long-term action reasoning. Specifically, we adopt Segment Anything (SAM) pre-trained on a huge number of images and promptable masks as the foundation model for extracting task-relevant features, and employ parameter-efficient fine-tuning on robot data for a better understanding of embodied scenarios. To address long-horizon reasoning, we develop a novel multi-channel heatmap that enables the prediction of the action sequence in a single pass, notably enhancing execution efficiency. Experimental results from various instruction-following tasks demonstrate that SAM-E achieves superior performance with higher execution efficiency compared to the baselines, and also significantly improves generalization in few-shot adaptation to new tasks.

1. Introduction

Robot manipulation has made significant progress, benefiting from embodied datasets (Walke et al., 2023; Collaboration, 2023; Fang et al., 2023), Imitation Learning (IL) (Jiang et al., 2023; Reed et al., 2022) or Reinforcement Learning (RL) algorithms (Zakka et al., 2023; Hansen et al., 2022; Shi et al., 2024; Bai et al., 2024), and advanced transformer (Chebotar et al., 2023; Zhao et al., 2023a) or diffusion-based networks (Xian et al., 2023; He et al., 2023; 2024). To perform a wide range of complex manipulation tasks in the 3D physical world, it is crucial to understand the 3D scene structure that encompasses object positions, orientations, shapes, occlusions, and the relationships between objects and the environment (Billard & Kragic, 2019). Various methods utilize 3D representations such as voxel patches (James et al., 2022b; Shridhar et al., 2022b), point clouds (Chen et al., 2023; Zhang et al., 2023b) to provide 3D localizations for predicting the end-effector poses. However, learning a 3D representation can be computationally expensive. For instance, the voxel-based method (Shridhar et al., 2022b) achieves state-of-the-art performance while suffering from cubic scaling of the number of voxels with the resolution, making it prohibitive for larger datasets.

To tackle these challenges, recent studies have investigated feature extraction from single-view images and information aggregation using multi-view transformers (Guhur et al., 2022), which provide enhanced efficiency as the scaling of image patches aligns with the input resolution. For example, recently proposed RVT (Goyal et al., 2023) achieves 36 times faster training speeds and better performance than voxel-based approaches. However, learning a multi-view policy still requires a considerable amount of high-quality robot trajectories for imitation, and the resulting policy exhibits limited generalization capabilities for unseen tasks and low execution efficiency in long-horizon reasoning. Motivated by recent research on visual foundation models that leverage web-scale datasets and demonstrate robust zero-shot and few-shot generalization (Radford et al., 2021; Li et al., 2022; Rombach et al., 2022; Hudson et al., 2023), we delve further into the multi-view architecture to enhance the generalization capabilities and execution efficiency of 3D manipulation policies in language-following tasks.

In this paper, we present a novel architecture for robot manipulation that leverages a vision-foundation model for image understanding and sequence imitation for long-horizon reasoning. We name our method **SAM-E**, as we utilize the

[1]Tsinghua Shenzhen International Graduate School, Tsinghua University [2]Shanghai Artificial Intelligence Laboratory [3]Hong Kong University of Science and Technology [4]Institute of Artificial Intelligence (TeleAI), China Telecom, P. R. China. Correspondence to: Chenjia Bai <baichenjia@pjlab.org.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

Segment Anything Model (**SAM**) (Kirillov et al., 2023) as the foundation model for **E**mbodied manipulation. SAM is a prompt-conditioned image segmentation model trained on a large dataset of images and masks. Utilizing SAM as the foundational perception model benefits the scene understanding and generalization in various manipulation scenarios. Moreover, the prompt-conditioned SAM encoder is suitable for language-instructed manipulation by extracting task-relevant visual features according to the task descriptions. Further, we conduct parameter-efficient finetuning for SAM with robot data to enhance the understanding of embodied scenarios. With prompt-guided features, we employ multi-view attention to integrate the view-wise representations with coordinate information for action prediction.

To improve the efficiency of long-horizon action prediction, we propose a novel prediction head that generates multi-channel pose heatmaps for an action sequence. Subsequently, the heatmaps from different views are back-projected into 3D space to generate scores for a discretized set of 3D points, ultimately determining the 3D positions and rotations of actions. During inference, the action sequence can be predicted in a single pass and executed sequentially, resulting in a notable improvement in execution efficiency compared to previous step-by-step prediction methods. We conduct experiments on various 3D instruction-following tasks from RLBench, consisting of 18 tasks with 249 variations (James et al., 2020). The results demonstrate that SAM-E achieves superior performance and higher reasoning efficiency compared to baseline methods. Moreover, the visual foundation model greatly enhances the generalization ability of the learned policy in adapting to new tasks with few-shot demonstrations.

2. Preliminaries

LC-POMDP. The problem of language-conditioned robot manipulation can be modeled as a Language-Conditioned Partial Observable Markov Decision Process (LC-POMDP) formulated as an augmented POMDP $\mathcal{M} := (\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \rho_0, \mathcal{L}, f, T)$, where $\mathcal{S}$ and $\mathcal{A}$ denote state space and action space separately, O denotes the space of observations, $\mathcal{P}(s|s,a) : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ denotes the transition probability or the environment dynamics, ρ_0 represents the initial state distribution, $\mathcal{L}$ denotes the set of all language instructions, $f(o|s) : \mathcal{S} \to \mathcal{O}$ is the observation function, and T represents the episode horizon. We adopt imitation learning without considering the reward function used for RL. For each episode, the robot is given a language instruction $l \in \mathcal{L}$ representing the goal of the current task. At each time step t, the robot is required to take action according to a policy $\pi(a_t|o_t, l)$ given the observation o_t. Since we focus on 3D manipulation, the observation o_t contains multi-view images from cameras at different perspectives.

Imitation Learning. To address the language-conditioned manipulation tasks, imitation learning (Goyal et al., 2023; Li et al., 2024) allows the agent to mimic a set of expert demonstrations denoted as $\mathcal{D} := \{(\tau, l)_i\}_{i=0}^{|\mathcal{D}|}$, where $\tau := (o_0, a_0, \ldots, o_{T-1}, a_{T-1}, o_T)$ is the expert trajectory, and l represents the language instruction. A common imitation learning objective for the policy π_θ is to maximize the likelihood of action conditioned on the language and current state, Formally, the loss function is

$$\mathcal{L}(\theta) := -\mathbb{E}_{(\tau,l)\sim\mathcal{D}} \left[\sum_{t=0}^{T-1} \log \pi_\theta(a_t|o_t, l) \right]. \tag{1}$$

Key-frame Extraction. To improve the utilization of expert demonstrations, we align with the consensus in 3D manipulation algorithms (James & Davison, 2022; James et al., 2022a; Shridhar et al., 2022b; Goyal et al., 2023) by incorporating key-frame extraction for selecting key-frame actions. The key-frame extraction involves a Boolean function $K : \mathbb{R}^{|\mathcal{A}|} \to \{0, 1\}$, which determines whether an action should be identified as a key-frame. For each demonstration τ, a sequence of key-frame actions $\{k_1, k_2, \ldots, k_m\}$ is generated by the function K following two simple conditions: (i) the joint-velocities are near zero (occurs when entering pre-grasp poses or a new phase of task), and (ii) gripper state has changed (occurs when the object is grasped or released). Based on the function K, the imitation objective in Eq. (1) becomes predicting the 'next key-frame action' in the demonstration. In the following, we slightly abuse a_t to represent the next key-frame action of s_t since we adopt the same key-frame extraction process to SAM-E and baselines.

3. Method

The proposed **SAM-E** is a multi-view imitation framework that leverages the pre-trained visual foundation model and action-sequence imitation for multi-task 3D manipulation. The key idea of SAM-E contains two perspectives: (i) leveraging the visual foundation model SAM with the prompt-driven architecture and its strong generalization ability to handle the language-prompt(instructed) tasks in embodied scenarios; (ii) utilizing the temporal smooth properties of actions to perform sequence modeling of actions to enhance coherent planning and execution efficiency. We introduce the visual foundation model for embodied perception in §3.1 and the multi-view architecture in §3.2. Then we give the motivation of sequence imitation in §3.3 and the multi-channel prediction architecture in §3.4.

We illustrate the architecture of SAM-E in Figure 1. Overall, we adopt the SAM encoder (Kirillov et al., 2023) to generate prompt-guided and object-oriented representations, and fine-tune the encoder with embodied data and Low-Rank Adaptation (LoRA) (Hu et al., 2022) technique for manip-

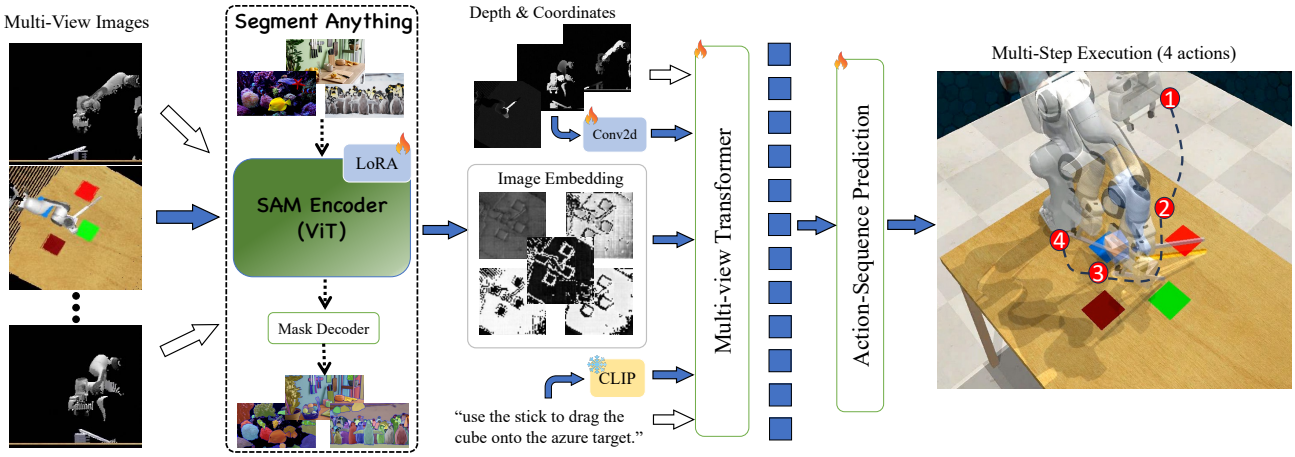

Figure 1. **Overview of SAM-E.** (i) The SAM encoder provides promptable visual embedding of single-view observations after fine-tuning on embodied scenarios with parameter-efficient LoRA. (ii) Multi-view transformer achieves cross-view information integration and vision-language alignment. (iii) The coherent action sequence is predicted via temporal imitation for efficient multi-step execution.

ulation scenarios, which results in a minimal increase in computation requirement. Then, a multi-view transformer is used to integrate cross-view visual information combined with coordinate information and language instruction for multi-view correspondence and vision-language alignment. To address long-horizon action prediction, SAM-E predicts a coherent action sequence in a single pass with a novel action-sequence policy head.

3.1. Perception Foundation and LoRA Finetune

SAM for Promptable Perception. SAM (Kirillov et al., 2023) comprises a powerful image encoder and lightweight mask decoder, structured as a prompt-driven architecture designed for real-world image segmentation. Aiming at achieving promptable segmentation and effective ambiguity awareness, the image encoder of SAM is trained with flexible prompts from the downstream mask decoder. Consequently, after diverse segmentation task training, the SAM encoder is capable of extracting powerful object-centering image embedding rich in semantic information. This also enables SAM to handle unknown prompts arising from various segmentation requirements in robot interactions, including complex object-associated scenarios.

In 3D manipulation, the scene perception is expected to be object-oriented and adaptable, accommodating a range of intentions and shifting focus as tasks progress. For instance, given the task instruction of '*place the apple in the basket*', the agent should first find and focus on the apple to pick it up, followed by finding the basket to place. The perception module should be capable of flexible object-centered attention based on task instructions and allow attention adjustment to other objects as the task progresses (See §C for an example). From this point, the SAM encoder is suitable as a perception foundation model for language-instructed manipulation with rich task variations. The SAM encoder is a Vision Transformer (ViT) (Dosovitskiy et al., 2021) pre-trained with MAE (He et al., 2022), which processes RGB images into $C \times H \times W$ image embedding. In practice, we utilize the ViT-B architecture for the image encoder to showcase the advantages of pre-trained segmentation representations with a low computational cost in manipulation tasks. The image encoder contains 12 layers of transformer blocks and outputs the image embedding of the visual inputs. The proposed SAM-E leverages the SAM encoder as the foundation to generate prompt-guided and object-oriented representations from visual observations, which is essential for language-instructed manipulation.

LoRA with Embodied Data. To effectively adapt the SAM encoder to embodied scenarios at an affordable computing cost, we employ LoRA to finetune the encoder during the policy training. As indicated in LoRA, we freeze the parameters in the image encoder and add a trainable low-rank bypass to each of the transformer encoder blocks as:

$$W_0 + \Delta W = W_0 + BA, \tag{2}$$

where $W_0 \in \mathbb{R}^{d\times k}$ is the pre-trained weight matrix frozen during training, $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ are trainable matrix, and rank $r \ll min(d, k)$. $\Delta W = BA$ represents the accumulated gradient update during adaptation with A initiated by Gaussian initialization and B initiated with zero. We set the rank r to 4 by default. In practice, we apply LoRA to the self-attention modules with query and value projection layers:

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{3}$$

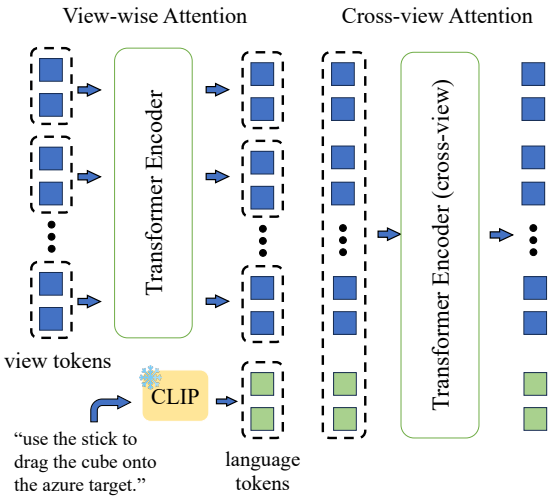

Figure 2. **Multi-view Transformer** has two stages for view-wise information and cross-view information integration.

$$Q = \hat{W}_q X = W_q X + B_q A_q X, \tag{4}$$

$$K = W_k X, \tag{5}$$

$$V = \hat{W}_v X = W_v X + B_v A_v X, \tag{6}$$

where W_q, W_k and W_v are frozen projection weights inherited from SAM encoder, and A_q, B_q, A_v, and B_v are trainable LoRA parameters.

3.2. Multi-View Transformer

After extracting the view-wise representations, we adopt a multi-view transformer to integrate multi-view visual observations, depth information with coordinates, and task-relevant language instructions using an attention mechanism, enabling a comprehensive fusion of the input in multiple modalities. The architecture is shown in Figure 2. The visual observations are processed into image embedding by the previously mentioned SAM encoder, while depth and coordinate information is processed through a Conv2D layer to obtain 3D spatial features. We concatenate the image embeddings with spatial features in the channel dimension along the patch tokens, resulting in a combined representation that we refer to as 'view tokens'. Additionally, we utilize a pre-trained CLIP text encoder to generate language embeddings, from which language tokens are derived. Firstly, view tokens from the same view pass through view-wise attention blocks like ViT to maintain the single-view information. Subsequently, visual tokens across different views and the language tokens are attended to cross-view attention blocks, to integrate cross-view scene information with language instructions. The visual tokens, now enriched with cross-view information and language information are used as input for the action-sequence prediction.

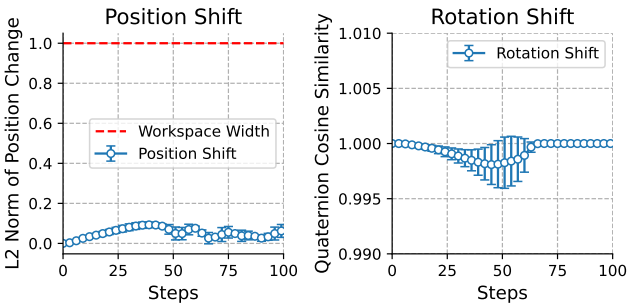

Figure 3. Movement shift in positions and rotations of the end effector in RLBench task *close_jar*, representing smooth changes of positions and rotations in temporally adjacent steps.

3.3. Motivation for Action-Sequence Modeling

In the next, we aim to provide the intuition of action-sequence modeling, attempting to ground the utility of this technique. We start with an assumption about the temporal smooth properties of actions in the robot manipulation.

Assumption 3.1 (Temporal-Smooth Assumption)**.** *Since the actions of the manipulation task are the desired positions and rotations of the end effector, the optimal action sequences* $(a_0^*, a_1^*, a_2^*, \dots, a_T^*)$ *are smooth, formulated as:*

$$\|a_t^* - a_{t+1}^*\| < \epsilon, \;\; 0 \le t \le T-2, \tag{7}$$
$$\forall\, \tau^* := (o_0, a_0^*, o_1, a_1^*, \dots, o_{T-1}, a_{T-1}^*, o_T) \sim \mathrm{P}_{\pi^*}^{\tau}(\cdot),$$

where $\mathrm{P}_{\pi^*}^{\tau}(\cdot)$ *denotes the distribution of trajectories derived from the optimal policy* π^*.

Intuitively, the assumption holds in most embodied manipulation tasks if the actions are the positions and rotations of the end effector. For example, in the common Pick-and-Place tasks, the optimal action sequences are a sequence of points in Euclidean space, which leads the end effector to approach the object and desired goal. Meanwhile, the gripper will rotate smoothly to align with the gripping points of the object. In Figure 3, we show the movement shift of positions and the Quaternion angle of rotations of the end-effector in a manipulation task *close jar* from RLBench (James et al., 2020), which further justifies our assumption. We observe that in certain manipulation tasks, end effector rotations undergo relatively rapid changes, particularly with large keyframe intervals, which weaken the assumption of smooth rotation. However, the end effector positions maintain superior smoothness in Euclidean space, which are more crucial for action-sequence modeling in our method.

The typical approach trains the policy π to predict the action a_t given the multi-view image o_t and the task instruction l, as

$$\pi(l, o_t) \rightarrow a_t.$$

Such a step-by-step process only focuses on predicting ac-

tions of the current situation, which can lead to stagnation and contradictory sequential actions, as observed in experiments. Based on the Assumption 3.1, we can improve the action-prediction process by considering a long-horizon decision process instead of a single action, as

$$\pi^{\text{seq}}(l, o_t) \rightarrow \{a_t, a_{t+1}, \ldots, a_{t+h-1}\},$$

where h is the horizon of the action sequence.

Then we motivate the sequence-prediction procedure based on the assumption. The sequence modeling process tries to predict the optimal action sequence condition on the observation. Intuitively, the learning objective of π^{seq} is more difficult compared to that of π^{step}. However, when we take a closer look at the prediction of action in a sequence (e.g., a_{t+k}), training to predict this action is accompanied by the prediction of former actions (i.e., $(\hat{a}_t, \ldots, \hat{a}_{t+k-1})$) and latter actions (i.e., $(\hat{a}_{t+k+1}, \ldots, \hat{a}_{t+h-1})$). Back to the assumption that the optimal action sequences are smooth, we believe that predicting the former and latter actions can provide *implicit prior* and *constraint* in predicting a_{t+k}. Thus, the smooth properties of action sequences provide an opportunity to perform long-horizon reasoning by predicting the adjacent actions as a whole, thereby reflecting the motion trajectory of the robot's end-effector in completing tasks. In contrast, the action prediction of the traditional policy is only conditioned on the observation without any 'prompt' from the former actions, making the traditional policy inferior to action-sequence modeling in these tasks. Such a technique in 2D manipulation tasks is also called action chunking (Bharadhwaj et al., 2023; Zhao et al., 2023a), while we give a clear motivation by an empirically justified assumption and extend it to 3D scenarios using multi-channel heatmaps.

3.4. Architecture for Action-Sequence Prediction

We introduce a novel multi-channel policy head for the action-sequence prediction, as shown in Figure 4. The policy head takes view tokens from the multi-view transformer (shown in Figure 2) as input, processing view tokens from different views independently, and outputs action sequence prediction in parallel channels within a single view image.

In 3D manipulation, each action in the sequence comprises an 8-dimensional vector dictating the next movement. This vector includes a 6-DoF target end effector pose (3-DoF for position and 3-DoF for rotation), a binary value indicating the gripper state (open or closed), and another binary value determining whether the collision is permissible for the low-level motion planner. (i) For predicting positions, the policy head generates a heatmap from the view tokens corresponding to each view. These heatmaps represent the desired position distribution from the perspective of each view. Then the heatmaps from different views are back-projected to into 3D space to generate scores for a discretized set of 3D points, determining the 3D positions. For action-sequence prediction, we equip the heatmap with time-dimension channels to learn temporal information from demonstrations, which leads to coherent action prediction in the temporal dimension. (ii) For predicting rotations, we follow previous methods (Goyal et al., 2023) to discretize Euler angles into bins of 5° resolution and thus turn rotation prediction into classification as the binary of gripper state and collision indicator. We use heatmap as the weight to extract the view-wise features from the view tokens, which provide higher weights near the desired target position within the view image, and then output the action sequence of rotation, gripper state, and collision indicator using a fully connected network.

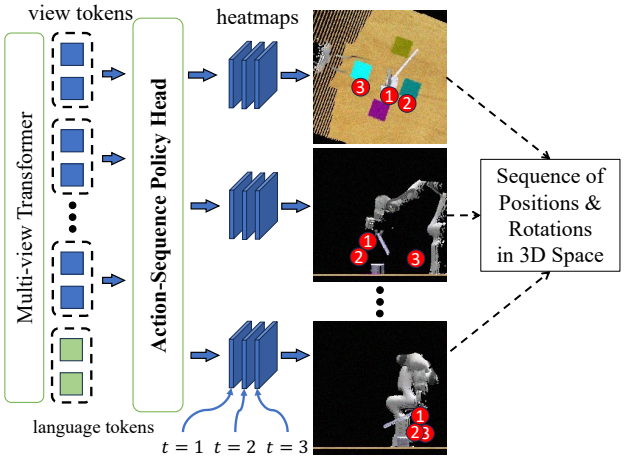

Figure 4. The **Action-Sequence Policy Head** outputs multi-channel pose heatmaps for a sequence of positions and rotations.

4. Related Works

Visual Robot Manipulation. Early research in robot manipulation adopts joint states of the robot arm and geometric information of objects in RL or IL frameworks (Zeng et al., 2017; Deng et al., 2020; Xie et al., 2020; Yu et al., 2020; Xu et al., 2022), assuming the acquisition of pre-perception information and coordinates of objects. In real-world manipulation tasks, visual perception provides more general inputs without additional assumptions (Yuan et al., 2023). Various methods have adopted visual pretraining models for affordance (Goyal et al., 2022; Bahl et al., 2023), representation learning (Khandelwal et al., 2022; Shridhar et al., 2022a; Nair et al., 2023; Ma et al., 2023b;a), and goal generation (Gao et al., 2023; Jia et al., 2023) to facilitate policy learning. Other works incorporate language encoders (Xie et al., 2023) and cross-modal transformers (Brohan et al., 2023b; Guhur et al., 2022) for instruction-following manipulation. However, these methods learn manipulation policies from top-down 2D images and are limited to pick-and-place primitives (Hansen et al., 2023). In contrast, by leverag-

ing 3D perception, the robot is able to take into account object orientations, occlusions, and collisions in complex manipulation tasks. Recent methods utilize 3D representations, such as voxel patches (James et al., 2022b; Shridhar et al., 2022b), point clouds (Chen et al., 2023; Zhang et al., 2023b; Eisner et al., 2022), and feature fields (Gervet et al., 2023), to achieve accurate 3D localizations for action prediction. Another line of research utilizes multi-view images to represent the projections of a 3D environment onto image planes, significantly reducing the computation requirements (Liu et al., 2023a; Seo et al., 2023; Goyal et al., 2023). Our method lies in multi-view architectures and leverages pre-trained foundation models to enhance generalization across various visual scenarios and task descriptions. The technique of action chunking is also employed in 2D manipulation (Bharadhwaj et al., 2023; Zhao et al., 2023a), while we extend it to 3D scenarios using multi-channel heatmaps.

Foundation Models for Embodied Agents. Large Language Models (LLMs) (Touvron et al., 2023; Hu et al., 2023a), Vision Language Models (VLMs) (Liu et al., 2023b; Li et al., 2023), and vision foundation models (Radford et al., 2021) have demonstrated remarkable capabilities (Akyürek et al., 2023) and hold great promise for solving complex embodied tasks. The chain-of-thought capacity (Wei et al., 2022) of LLMs has been effectively utilized in task planning for embodied agents, including EmbodiedGPT (Mu et al., 2023), ReAct (Yao et al., 2023), SayCan (Ahn et al., 2022), and DoReMi (Guo et al., 2023). The commonsense knowledge within LLMs can serve as a world model (Zhao et al., 2023b; Hao et al., 2023; Lin et al., 2023) in text-based environments. Additionally, it can be utilized as a reward designer, as demonstrated by VoxPoser (Huang et al., 2023), Text2Reward (Xie et al., 2024), and Eureka (Ma et al., 2024). GenSim (Wang et al., 2024) and RoboGen (Wang et al., 2023a) leverage LLMs to generate task curricula and simulation environments to augment robot data. VLMs are commonly employed as foundation models for embodied policies, taking visual observations and language instructions as inputs, and generating language plans (Driess et al., 2023) or tokenized actions (Brohan et al., 2023a; Wu et al., 2023) as outputs. Other approaches utilize VLMs for reward generation (Rocamonde et al., 2024) in RL frameworks and self-reflection for task planning (Hu et al., 2023b). RoboFlamingo (Li et al., 2024) is related to our method as it employs OpenFlamingo as a base policy and finetunes this policy using embodied datasets. However, it is limited to 2D manipulation and lacks explicit consideration of 3D geometry, which hinders its capacity to develop highly accurate spatial manipulation skills in robotics.

Segment Anything Model. SAM (Kirillov et al., 2023) is a promptable segmentation model capable of generating masks by receiving various prompts, including points, bounding boxes, and language prompts. Subsequent works have examined the application of SAM for object localization (Zhang et al., 2023a), tracking (Rajič et al., 2023; Cheng et al., 2023), and semantic analysis (Mazurowski et al., 2023). For embodied agents, SAM-G (Wang et al., 2023b) is a concurrent work that utilizes point prompts to establish correspondences and employs SAM to generate masked images for the agent. However, SAM-G focuses on extracting the agent-relevant mask for robust visual representations and mitigating the impact of noise (e.g., colors, backgrounds) in 2D manipulation and locomotion tasks. In contrast, our method adopts SAM to enhance 3D manipulation within a multi-view framework and extracts task-relevant features to facilitate generalization across various manipulation scenarios and language instructions.

5. Experiments

In this section, we evaluate SAM-E in RLBench (James et al., 2020), which is a challenging multi-task 3D manipulation benchmark. To perform a fair comparison to baselines, we use the same settings as the state-of-art method (Goyal et al., 2023) by using 18 tasks with 249 variations in experiments. Moreover, we evaluate the generalization ability of SAM-E via few-shot adaptation in 6 new tasks. The Videos are available at: `https://sam-embodied.github.io/`.

5.1. Experiment Setup

Baselines. We compare SAM-E against off-the-shelf algorithms proved to work on multi-view 3D manipulation, including (i) **RVT** (Goyal et al., 2023), the state-of-the-art multi-view architecture for 3D manipulation by re-rendering visual observations into orthographic projections of cube views and predicting the next move based on these projections; (ii) **PerAct** (Shridhar et al., 2022b), an action-centric method that encodes RGB-D images into voxel grid patches for 3D representation and predicts the action within the 3D voxel space. (iii) We include **R3M** (Nair et al., 2023), the visual representation designed for robot manipulation, as an alternative encoder in our architecture. (iv) We include two more general visual representations CLIP (Radford et al., 2021), DINO (Caron et al., 2021) in our architecture. (v) We include a variant referred to **SAM→RVT** that replaces the SAM encoder with RVT's visual encoder, which is trained from scratch. (vi) Since RVT has been shown to significantly outperform other behavior cloning (BC) baselines including **CNN-BC**, **ViT-BC** (Jang et al., 2021), and **Coarse-to-Fine BC** (James et al., 2022b), we do not include the scores of these methods and we refer to Goyal et al. (2023) for details. (vii) Additionally, we compare SAE-E against **Hiveformer** (Guhur et al., 2022) with same tasks evaluated in their paper (we refer to §E for the results).

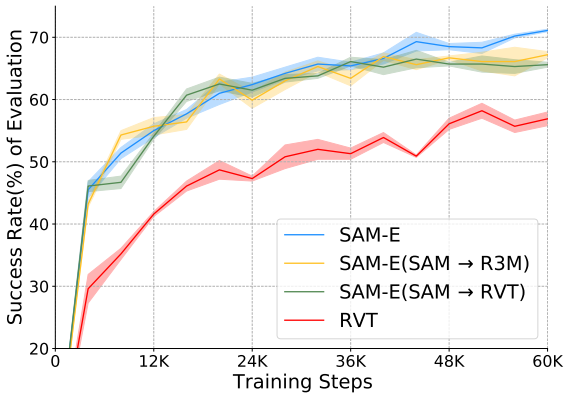

Figure 5. The comparison of training curves from 5 seeds with ± 1 std. We observe that SAM-E achieves a higher success rate than R3M and non-pre-trained baselines. Meanwhile, SAM and its variations achieve a better training efficiency compared to RVT, benefiting from action sequence imitation. The training curve of RVT is from our reproduction by running the official code.

Table 1. A comparison of trainable parameters to baselines.

Models	RVT	SAM-E (SAM → RVT)	SAM-E (SAM → R3M)	SAM-E
Trainable Para.	36.3M	35.6M	35.6M	35.7M

Simulation Environment. We perform experiments in RLBench (James et al., 2020), which is simulated by CoppeliaSim (Rohmer et al., 2013) to control a Franka Panda robot equipped with a parallel gripper. Visual observations are captured by four RGB-D cameras (left shoulder, right shoulder, front, and wrist) with a resolution of 128×128, and target gripper pose is achieved by a sample-based motion planner. In this elaborated simulator, the agent is tested to complete the task within a limited number of timesteps, which is 25 in experiments. The tasks include picking and placing items, executing staged moves for tool usage, and comprehending scenes to solve puzzles (see §A for more detailed descriptions of the tasks). The algorithms are evaluated in a multi-task and multi-modal setting, characterized by a high degree of variation, which necessitates the agent to demonstrate scene understanding, instruction comprehension, and precise action prediction.

Training Datasets. We utilize the same training datasets as RVT and PerAct, comprising 100 expert demonstrations per task. Unlike RVT and PerAct, which slice demonstration episodes into keyframe transitions with empirically crucial duplication for important transitions, we seamlessly decompose demonstrations into multiple sub-episodes of keyframes to facilitate action-sequence prediction. We train SAM-E for 60K steps and choose the last model for evaluation, which is the same as RVT. We use cosine learning rate decay after 2K steps warm-start (see §B for more details).

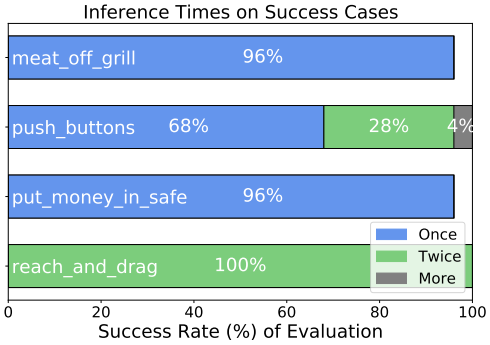

Figure 6. An illustration of the execution efficiency in several tasks. SAM-E completes most tasks in merely once or twice inferences in all success cases. We refer to §D for more examples.

5.2. Main Experiments

Multi-Task Learning. We train all methods in 18 tasks and the comparison of success rate is given in Table 2. SAM-E outperforms PerAct and RVT in 14 out of 18 tasks. SAM-E outperforms PerAct and RVT by an average of **21.2%** and **7.7%** percentage points in success rate across 18 tasks, marking a relative improvement with **43.0%** and **12.2%**, while incurring significantly lower model inference costs. Furthermore, it achieves an improvement exceeding 30% points in several tasks. Eliminating the pre-trained SAM encoder in SAM-E leads to a performance drop but still outperforms RVT, benefiting from the action sequence policy head. Building upon this, the addition of R3M's frozen representation has yielded a marginal performance improvement, however, which is still inferior compared to SAM-E. Similarly, CLIP and DINO representations have mediocre performances compared to SAM-E. Notably, SAM-E has comparable training time and even less trainable parameters compared to RVT, as shown in Table 1. Moreover, Figure 5 shows that SAM-E and its variations exhibit higher training efficiency than RVT, mainly attributed to the action sequence imitation. Further, utilizing SAM as the scalable visual foundation, SAM-E not only achieves the best performance on the current setup, but also shows potential for further enhancing its advantages with more embodied data or update steps.

Different from baselines that predict the next keypoint gripper pose at each timestep, SAM-E generates a sequence of actions for long-term planning and sequential execution, thereby considering the task completion from a higher perspective and has much fewer inference steps. According to Table 2, SAM-E demonstrates an average execution efficiency of more than **5X** greater than that of RVT. In tasks such as *meat_off_grill*, *push_buttons*, and *put_money_in_safe* (see §A for task descriptions), SAM-E can complete the task after merely **a glance** at the initial state, as shown in Figure 6. In contrast, RVT requires, on average, 5.5, 3.8, and 6.0 steps to complete them for its successful cases. For

Table 2. **Multi-task Performances.** SAM-E outperforms state-of-the-art methods in most tasks and on average, with much fewer inference steps in execution. Scores of PerAct and RVT are adopted from Goyal et al. (2023). Mean and std of 5 evaluations are reported.

Models	Put in Drawer	Reach and Drag	Turn Tap	Slide to Target	Open Drawer	Put in Cupboard	Place in Shape Sorter	Put Money in Safe	Push Buttons	Close Jar
PerAct	51.2±4.7	89.6±4.1	88.0±4.4	74.0±13.0	88.0±5.7	28.0±4.4	16.8±4.7	84.0±3.6	92.8±3.0	55.2±4.7
RVT	88.0±5.7	99.2±1.6	93.6±4.1	81.6±5.4	71.2±6.9	49.6±3.2	36.0±2.5	91.2±3.0	**100.0**±0.0	52.0±2.5
SAM-E (SAM → RVT)	87.2±5.9	**100.0**±0.0	**100.0**±0.0	79.2±6.6	**95.2**±3.3	59.2±5.2	35.2±4.4	72.0±4.0	98.4±2.2	83.2±5.9
SAM-E (SAM → R3M)	83.2±5.9	99.2±1.8	**100.0**±0.0	88.8±4.4	**95.2**±3.3	41.6±7.3	31.2±7.7	**95.2**±3.3	96.0±0.0	78.4±2.2
SAM-E (SAM → CLIP)	88.8±3.3	**100.0**±0.0	**100.0**±0.0	78.4±13.4	92.0±4.0	40.0±4.9	**42.4**±6.1	80.8±1.8	**100.0**±0.0	73.6±2.2
SAM-E (SAM → DINO)	78.4±4.6	99.2±1.8	99.2±1.8	88.0±4.9	89.6±5.4	52.0±7.5	30.4±9.2	85.6±2.2	**100.0**±0.0	**89.6**±3.6
SAM-E (ours)	**92.0**±5.7	**100.0**±0.0	**100.0**±0.0	**95.2**±1.8	**95.2**±5.2	**64.0**±2.8	34.4±6.1	**95.2**±3.3	**100.0**±0.0	82.4±3.6

Models	Stack Blocks	Place Cups	Place Wine at Rack	Screw Bulb	Sweep to Dustpan	Insert Peg	Meat off Grill	Stack Cups	**On Average**	**Inference Steps(Sum)**
PerAct	26.4±3.2	2.4±3.2	44.8±7.8	17.6±2.0	52.0±0.0	5.6±4.1	70.4±2.0	2.4±2.0	49.4	-
RVT	28.8±3.9	**4.0**±2.5	91.0±5.2	48.0±5.7	72.0±0.0	11.2±3.0	88.0±2.5	**26.4**±8.2	62.9	6158±64
SAM-E (SAM → RVT)	22.4±3.6	0.0±0.0	92.8±6.6	61.6±9.2	84±0.0	7.2±5.9	95.2±3.3	3.2±3.3	65.3±0.6	1190±19
SAM-E (SAM → R3M)	**32.0**±2.8	1.6±2.2	92.8±3.3	60.0±2.8	96.8±3.3	5.6±6.7	**97.6**±2.2	2.4±2.2	66.5±1.0	1165±63
SAM-E (SAM → CLIP)	22.4±10.8	0.0±0.0	93.6±2.2	59.2±4.4	85.6±2.2	8.0±2.8	96.0±4.0	4.8±3.3	64.8±0.9	1192±17
SAM-E (SAM → DINO)	28.8±7.7	0.8±1.8	93.6±3.6	64.0±9.8	**100.0**±0.0	11.2±3.3	96.0±2.8	1.6±2.2	67.1±0.4	1143±15
SAM-E (ours)	26.4±4.6	0.0±0.0	**94.4**±4.6	**78.4**±3.6	**100.0**±0.0	**18.4**±4.6	95.2±3.3	0.0±0.0	**70.6**±0.7	**1130**±12

Table 3. **Few-shot adaptation.** Mean and std of 5 evaluations are reported.

Models	Meat on Grill	Open Jar	Screw Nail	Toilet Seat Done	TV on	Solve Puzzle	**On Average**
RVT (from scratch)	**80.0**±6.3	**36.0**±4.9	7.2±4.4	99.2±1.8	2.4±3.6	11.2±4.4	39.3±2.3
SAM-E (from scratch, SAM → RVT)	60.0±2.8	12.0±0.0	**36.0**±6.9	96.0±0.0	15.2±3.3	20.8±7.7	40.0±1.8
SAM-E (from scratch, SAM → R3M)	69.6±4.6	16.0±0.0	29.6±6.1	**100.0**±0.0	12.0±2.8	**22.4**±2.2	41.6±1.4
SAM-E (from scratch, SAM → CLIP)	64.0±0.0	12.0±0.0	16.0±4.0	**100.0**±0.0	14.7±6.1	**24.0**±4.0	38.4±1.0
SAM-E (from scratch, SAM → DINO)	53.3±8.3	12.0±4.0	26.7±2.3	**100.0**±0.0	16.0±4.0	**24.0**±4.0	38.7±1.2
SAM-E (from scratch)	75.2±4.4	12.8±1.8	28.0±8.0	**100.0**±0.0	**20.8**±1.8	17.6±2.2	**42.4**±1.5
RVT (adaptation)	68.8±3.3	36.0±0.0	1.6±2.2	**100.0**±0.0	1.6±2.2	14.4±6.7	37.1±1.0
SAM-E (adaptation, SAM → RVT)	69.6±6.1	39.2±3.3	38.4±4.6	99.2±1.8	17.6±2.2	38.4±3.6	50.4±1.1
SAM-E (adaptation, SAM → R3M)	64.8±5.9	37.6±2.2	28.8±5.9	**100.0**±0.0	12.8±1.8	37.6±6.7	46.9±2.3
SAM-E (adaptation, SAM → CLIP)	78.7±2.3	38.7±4.6	28.0±6.9	**100.0**±0.0	16.0±0.0	25.3±8.3	47.8±1.9
SAM-E (adaptation, SAM → DINO)	68.0±4.0	33.3±6.1	50.7±8.3	98.7±2.3	24.0±4.0	22.7±2.3	49.6±1.0
SAM-E (adaptation)	**84.0**±5.7	**56.0**±7.5	**62.4**±4.6	**100.0**±0	**35.2**±1.8	**41.6**±7.3	**63.2**±1.5

reach_and_drag, SAM-E completes it all in two inferences while RVT needs to execute 6.4 times on average.

Few-Shot Adaptation. We evaluate the generalization ability of SAM-E by adapting the trained model to 6 new tasks from RLBench. We use 10X fewer demonstrations and 15X fewer update steps in policy adaptation than that of the multi-task experiments to show the generalization capability of the SAM-E in few-shot adaptation. The results are shown in Table 3. We initialize the models with weights from their multi-task training for adaptation, and also introduce their random initialization variants for training from scratch. We find RVT struggles with transferring knowledge from previous tasks to new ones during adaptation, often resulting in performance drops compared to training from scratch. In contrast, SAM-E significantly benefits from adaptation compared to starting from scratch. Specifically, SAM-E outperforms RVT by **3.1%** points (a **7.9%** relative increase) when trained from scratch. However, during adaptation to new tasks, the performance gap widens dramatically, with SAM-E surpassing RVT by **26.1%** points, a substantial **70.4%** relative improvement. This demonstrates that SAM-E has superior generalization capabilities.

When training from scratch, *SAM-E (SAM → R3M)* achieves a slightly better performance than *SAM-E (SAM → RVT)* that does not have a pre-trained encoder, but results in worse performance in adaptation, which shows R3M has limited few-shot generalization ability. While worse than *SAM-E (SAM → R3M)* in training from scratch, *SAM-E (SAM → CLIP)* and *SAM-E (SAM → DINO)* have better performances in adaptation, showing greater generalization of the representations pre-trained in more general image data. *SAM-E (SAM → RVT)* also significantly outperforms RVT in adaptation over from scratch, demonstrating the enhanced generalization ability gained from the action-sequence prediction. In terms of adapting to new tasks, SAM-E equipped with a SAM encoder demonstrates significant advantages over the methods mentioned above. This highlights the exceptional capabilities of SAM-E to generalize in novel task descriptions.

5.3. Ablations

First, we conduct ablation experiments in multi-task experiments to verify the necessity of components in SAM-E. We

Table 4. Success rate and parameters amount of the variations

Models	Success Rate	Parameters	Trainable Parameters
SAM-E	**70.6**±0.7	122.5M	35.7M
SAM-E (SAM → RVT)	65.3±0.6	35.6M	35.6M
SAM-E (LoRA, QKV)	69.2±0.9	122.6M	35.8M
SAM-E (w/o LoRA)	67.2±1.0	122.4M	35.6M
SAM-E (full finetune)	65.8±1.0	122.4M	122.4M

Table 5. Ablation over action sequence length h

Models	Success Rate	Inference Time (ms)
RVT	62.9	103
SAM-E ($h = 1$)	30.6±1.4	126
SAM-E ($h = 3$)	64.0±0.6	144
SAM-E ($h = 5$)	70.6±0.7	152
SAM-E ($h = 7$)	66.5±1.2	156

include (i) *SAM-E (SAM → RVT)*; (ii) *SAM-E (LoRA, QKV)*, which is a variant of LoRA module additionally including K matrix of attention blocks; (iii) *SAM-E (w/o LoRA)*, a frozen SAM encoder without LoRA fine-tuning, and (iv) *SAM-E (full finetune)*, which performs full-parameter training of the SAM encoder. We give the brief result in Table 4. We find SAM is a crucial visual foundation and a suitable finetune method is required for adaptation to embodied scenarios. Using LoRA to parameter-efficiently finetuning, SAM is better than the variant that trains all parameters, which may lead to failure due to the limited demonstrations. For LoRA, adding the trainable matrix for Q and V is better than all Q, K, and V, which is consistent with previous observations (Hu et al., 2022). (See §F for the complete results)

Additionally, to illustrate the impact of the action sequence length h (refer to §B.2 for details), we conduct an ablation study on the action horizon, examining h values of $\{1,3,5,7\}$. During both the training and evaluation execution of the multi-task experiments, we modify the action horizon h while maintaining consistency in other experimental settings. The outcomes are presented in Table 5 (See §F for the complete results), showing the average success rate across 18 tasks and the computing time for each model inference on our same device during the model evaluation. We observe that $h = 5$ performs the best on the average success rate, while it may not suitable for certain tasks. We can also find that $h = 1$ leads to a drop in performance, which we attribute to the insufficient temporal information to drive SAM foundation training, combined with the lack of empirically crucial duplication for important transitions. Moreover, we can observe that SAM-E's inference time is slightly longer than that of RVT. Nevertheless, SAM-E is even faster in inference considering an action sequence (5 actions) is predicted in 152ms, while RVT requires 5*103ms to predict 5 actions.

5.4. Real-World Experiment

To demonstrate the effectiveness of SAM-E in real-world scenarios, we train and test the model in a real-world setup with a Franka Panda robot arm. As shown in Figure 14, we use two statically mounted RGB-D cameras in a third-person view at the left front and right front to capture the multi-view observation. We calibrate the cameras with the robot base and record the RGB-D streams from the cameras and robot joint pose simultaneously during the data collection. We train SAM-E in 5 tasks with 10 episodes for each, including *put the towel on the cabinet*, *stack the block*, *close the drawer*, *pick up the banana*, and *put the orange into the drawer*. All the episodes are collected by human demonstrators. The results show that SAM-E can perform real-time prediction in real-world scenarios and complete tasks effectively, validating SAM-E's capability in real-world scenarios. See the §G and the videos for more details and model performance.

6. Conclusion

We have introduced **S**egment **A**nything **M**odel for **E**mbodied 3D manipulation (SAM-E), a novel multi-view architecture that adopts SAM as the visual foundation model with parameter-efficient finetuning for promptable perception to embodied scenarios, as well as a novel action-sequence prediction head for efficient planning and coherent execution. We conduct experiments of SAM-E on various 3D instruction-following tasks from RLBench for multi-task experiments and few-show adaptation. We find SAM-E outperforms prior state-of-the-art models on multi-task manipulation and achieves a significant improvement in execution efficiency and few-shot adaptation with great generalization ability. Our work highlights the feasibility of leveraging a visual foundation model and sequence prediction for enhancing generalization and efficiency in 3D manipulation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos.62306242 & 62376222), the STI 2030-Major Projects under Grant 2021ZD0201404, the National Key R&D Program of China (No.2022ZD0160102), and Young Elite Scientists Sponsorship Program by CAST (No.2023QNRC001). We thank Wenke Xia for his excellent assistance in hardware deployment and data collection for real robot experiments.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Ahn, M., Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., Kalashnikov, D., Levine, S., and et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Annual Conference on Robot Learning*, 2022.

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. In *International Conference on Learning Representations*, 2023.

Bahl, S., Mendonca, R., Chen, L., Jain, U., and Pathak, D. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13778–13790, 2023.

Bai, C., Wang, L., Hao, J., Yang, Z., Zhao, B., Wang, Z., and Li, X. Pessimistic value iteration for multi-task data sharing in offline reinforcement learning. *Artificial Intelligence*, 326:104048, 2024.

Bharadhwaj, H., Vakil, J., Sharma, M., Gupta, A., Tulsiani, S., and Kumar, V. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *First Workshop on Out-of-Distribution Generalization in Robotics at CoRL 2023*, 2023.

Billard, A. and Kragic, D. Trends and challenges in robot manipulation. *Science*, 364(6446):eaat8414, 2019.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M. G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., and et al. RT-2: vision-language-action models transfer web knowledge to robotic control. *CoRR*, abs/2307.15818, 2023a.

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., and et al. Rt-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems*, 2023b.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

Chebotar, Y., Vuong, Q., Hausman, K., Xia, F., Lu, Y., Irpan, A., Kumar, A., Yu, T., Herzog, A., Pertsch, K., et al. Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *Conference on Robot Learning*, pp. 3909–3928. PMLR, 2023.

Chen, S., Pinel, R. G., Schmid, C., and Laptev, I. Polarnet: 3d point clouds for language-guided robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023.

Cheng, Y., Li, L., Xu, Y., Li, X., Yang, Z., Wang, W., and Yang, Y. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023.

Collaboration, O. X. Open x-embodiment: Robotic learning datasets and RT-X models. *CoRR*, abs/2310.08864, 2023.

Deng, X., Xiang, Y., Mousavian, A., Eppner, C., Bretl, T., and Fox, D. Self-supervised 6d object pose estimation for robot manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3665–3671. IEEE, 2020.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR*, 2021.

Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., and Florence, P. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning*, volume 202, pp. 8469–8488, 2023.

Eisner, B., Zhang, H., and Held, D. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. In *Robotics: Science and Systems (RSS)*, 2022.

Fang, H.-S., Fang, H., Tang, Z., Liu, J., Wang, C., Wang, J., Zhu, H., and Lu, C. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 3:5, 2023.

Gao, J., Hu, K., Xu, G., and Xu, H. Can pre-trained text-to-image models generate visual goals for reinforcement learning? In *Neural Information Processing Systems*, 2023.

Gervet, T., Xian, Z., Gkanatsios, N., and Fragkiadaki, K. Act3d: 3d feature field transformers for multi-task robotic manipulation. In *Conference on Robot Learning*, 2023.

Goyal, A., Xu, J., Guo, Y., Blukis, V., Chao, Y.-W., and Fox, D. RVT: Robotic view transformer for 3d object manipulation. In *7th Annual Conference on Robot Learning*, 2023.

Goyal, M., Modi, S., Goyal, R., and Gupta, S. Human hands as probes for interactive object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3293–3303, 2022.

Guhur, P.-L., Chen, S., Pinel, R. G., Tapaswi, M., Laptev, I., and Schmid, C. Instruction-driven history-aware policies for robotic manipulations. In *6th Annual Conference on Robot Learning*, 2022.

Guo, Y., Wang, Y.-J., Zha, L., Jiang, Z., and Chen, J. Doremi: Grounding language model by detecting and recovering from plan-execution misalignment. *arXiv preprint arXiv:2307.00329*, 2023.

Hansen, N., Yuan, Z., Ze, Y., Mu, T., Rajeswaran, A., Su, H., Xu, H., and Wang, X. On pre-training for visuo-motor control: Revisiting a learning-from-scratch baseline. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12511–12526, 23–29 Jul 2023.

Hansen, N. A., Su, H., and Wang, X. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, pp. 8387–8406. PMLR, 2022.

Hao, S., Gu, Y., Ma, H., Hong, J. J., Wang, Z., Wang, D. Z., and Hu, Z. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

He, H., Bai, C., Xu, K., Yang, Z., Zhang, W., Wang, D., Zhao, B., and Li, X. Diffusion model is an effective planner and data synthesizer for multi-task reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

He, H., Bai, C., Pan, L., Zhang, W., Zhao, B., and Li, X. Large-scale actionless video pre-training via discrete diffusion for efficient policy learning. *arXiv preprint arXiv:2402.14407*, 2024.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

Hu, Y., Lin, F., Zhang, T., Yi, L., and Gao, Y. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv:2311.17842*, 2023a.

Hu, Y., Lin, F., Zhang, T., Yi, L., and Gao, Y. Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning. *arXiv preprint arXiv:2311.17842*, 2023b.

Huang, W., Wang, C., Zhang, R., Li, Y., Wu, J., and Fei-Fei, L. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Annual Conference on Robot Learning*, 2023.

Hudson, D. A., Zoran, D., Malinowski, M., Lampinen, A. K., Jaegle, A., McClelland, J. L., Matthey, L., Hill, F., and Lerchner, A. Soda: Bottleneck diffusion models for representation learning. *arXiv preprint arXiv:2311.17901*, 2023.

James, S. and Davison, A. J. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):1612–1619, 2022.

James, S., Ma, Z., Arrojo, D. R., and Davison, A. J. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

James, S., Wada, K., Laidlow, T., and Davison, A. J. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13739–13748, 2022a.

James, S., Wada, K., Laidlow, T., and Davison, A. J. Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13739–13748, 2022b.

Jang, E., Irpan, A., Khansari, M., Kappler, D., Ebert, F., Lynch, C., Levine, S., and Finn, C. BC-z: Zero-shot task generalization with robotic imitation learning. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=8kbp23tSGYv.

Jia, Z., Liu, F., Thumuluri, V., Chen, L., Huang, Z., and Su, H. Chain-of-thought predictive control. *arXiv preprint arXiv:2304.00776*, 2023.

Jiang, Y., Gupta, A., Zhang, Z., Wang, G., Dou, Y., Chen, Y., Fei-Fei, L., Anandkumar, A., Zhu, Y., and Fan, L. Vima: Robot manipulation with multimodal prompts. In *International Conference on Machine Learning*, pp. 14975–15022, 2023.

Khandelwal, A., Weihs, L., Mottaghi, R., and Kembhavi, A. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14829–14838, 2022.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022.

Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., and Qiao, Y. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

Li, X., Liu, M., Zhang, H., Yu, C., Xu, J., Wu, H., Cheang, C., Jing, Y., Zhang, W., Liu, H., et al. Vision-language foundation models as effective robot imitators. In *International Conference on Learning Representations*, 2024.

Lin, J., Du, Y., Watkins, O., Hafner, D., Abbeel, P., Klein, D., and Dragan, A. Learning to model the world with language. *arXiv preprint arXiv:2308.01399*, 2023.

Liu, H., Lee, L., Lee, K., and Abbeel, P. Instruction-following agents with jointly pre-trained vision-language models, 2023a. URL https://openreview.net/forum?id=U0jfsqmoV-4.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023b.

Ma, Y. J., Kumar, V., Zhang, A., Bastani, O., and Jayaraman, D. LIV: Language-image representations and rewards for robotic control. In *International Conference on Machine Learning*, volume 202, pp. 23301–23320, 2023a.

Ma, Y. J., Sodhani, S., Jayaraman, D., Bastani, O., Kumar, V., and Zhang, A. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *International Conference on Learning Representations*, 2023b.

Ma, Y. J., Liang, W., Wang, G., Huang, D., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., and Anandkumar, A. Eureka: Human-level reward design via coding large language models. In *International Conference on Learning Representations*, 2024.

Mazurowski, M. A., Dong, H., Gu, H., Yang, J., Konz, N., and Zhang, Y. Segment anything model for medical image analysis: an experimental study. *Medical Image Analysis*, 89:102918, 2023.

Mu, Y., Zhang, Q., Hu, M., Wang, W., Ding, M., Jin, J., Wang, B., Dai, J., Qiao, Y., and Luo, P. EmbodiedGPT: Vision-language pre-training via embodied chain of thought. In *Neural Information Processing Systems*, 2023.

Nair, S., Rajeswaran, A., Kumar, V., Finn, C., and Gupta, A. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pp. 892–909. PMLR, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Rajič, F., Ke, L., Tai, Y.-W., Tang, C.-K., Danelljan, M., and Yu, F. Segment anything meets point tracking. *arXiv preprint arXiv:2307.01197*, 2023.

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

Rocamonde, J., Montesinos, V., Nava, E., Perez, E., and Lindner, D. Vision-language models are zero-shot reward models for reinforcement learning. In *International Conference on Learning Representations*, 2024.

Rohmer, E., Singh, S. P., and Freese, M. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ international conference on intelligent robots and systems*, pp. 1321–1326. IEEE, 2013.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Seo, Y., Kim, J., James, S., Lee, K., Shin, J., and Abbeel, P. Multi-view masked world models for visual robotic manipulation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 30613–30632, 2023.

Shi, J., Bai, C., He, H., Han, L., Wang, D., Zhao, B., Li, X., and Li, X. Robust quadrupedal locomotion via risk-averse policy learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Shridhar, M., Manuelli, L., and Fox, D. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning*, pp. 894–906. PMLR, 2022a.

Shridhar, M., Manuelli, L., and Fox, D. Perceiver-actor: A multi-task transformer for robotic manipulation. In *6th Annual Conference on Robot Learning*, 2022b.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P.,

Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Walke, H. R., Black, K., Zhao, T. Z., Vuong, Q., Zheng, C., Hansen-Estruch, P., He, A. W., Myers, V., Kim, M. J., Du, M., Lee, A., Fang, K., Finn, C., and Levine, S. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pp. 1723–1736, 2023.

Wang, L., Ling, Y., Yuan, Z., Shridhar, M., Bao, C., Qin, Y., Wang, B., Xu, H., and Wang, X. Gensim: Generating robotic simulation tasks via large language models. In *International Conference on Learning Representations*, 2024.

Wang, Y., Xian, Z., Chen, F., Wang, T., Wang, Y., Erickson, Z., Held, D., and Gan, C. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *CoRR*, abs/2311.01455, 2023a.

Wang, Z., Ze, Y., Sun, Y., Yuan, Z., and Xu, H. Generalizable visual reinforcement learning with segment anything model. *arXiv preprint arXiv:2312.17116*, 2023b.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., brian ichter, Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Wu, H., Jing, Y., Cheang, C., Chen, G., Xu, J., Li, X., Liu, M., Li, H., and Kong, T. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, 2023.

Xian, Z., Gkanatsios, N., Gervet, T., Ke, T.-W., and Fragkiadaki, K. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=W0zgY2mBTA8.

Xie, A., Lee, Y., Abbeel, P., and James, S. Language-conditioned path planning. In *Conference on Robot Learning*, pp. 3384–3396. PMLR, 2023.

Xie, C., Xiang, Y., Mousavian, A., and Fox, D. The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation. In *Conference on robot learning*, pp. 1369–1378. PMLR, 2020.

Xie, T., Zhao, S., Wu, C. H., Liu, Y., Luo, Q., Zhong, V., Yang, Y., and Yu, T. Text2reward: Automated dense reward function generation for reinforcement learning. In *International Conference on Learning Representations*, 2024.

Xu, M., Shen, Y., Zhang, S., Lu, Y., Zhao, D., Tenenbaum, J., and Gan, C. Prompting decision transformer for few-shot policy generalization. In *International Conference on Machine Learning*, volume 162, pp. 24631–24645, 17–23 Jul 2022.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020.

Yuan, Z., Yang, S., Hua, P., Chang, C., Hu, K., and Xu, H. Rl-vigen: A reinforcement learning benchmark for visual generalization. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Zakka, K., Wu, P., Smith, L., Gileadi, N., Howell, T., Peng, X. B., Singh, S., Tassa, Y., Florence, P., Zeng, A., and Abbeel, P. Robopianist: Dexterous piano playing with deep reinforcement learning. In *Conference on Robot Learning*, 2023.

Zeng, A., Yu, K.-T., Song, S., Suo, D., Walker, E., Rodriguez, A., and Xiao, J. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 1386–1383. IEEE, 2017.

Zhang, R., Jiang, Z., Guo, Z., Yan, S., Pan, J., Dong, H., Gao, P., and Li, H. Personalize segment anything model with one shot. *arXiv preprint arXiv:2305.03048*, 2023a.

Zhang, T., Hu, Y., Cui, H., Zhao, H., and Gao, Y. A universal semantic-geometric representation for robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023b.

Zhao, T. Z., Kumar, V., Levine, S., and Finn, C. Learning fine-grained bimanual manipulation with low-cost hardware. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023a.

Zhao, Z., Lee, W. S., and Hsu, D. Large language models as commonsense knowledge for large-scale task planning. *arXiv preprint arXiv:2305.14078*, 2023b.

A. RLBench Tasks

We follow the multi-task multi-variation simulated experiments setting of RVT (Goyal et al., 2023) and PerAct (Shridhar et al., 2022b) with 18 RLBench tasks (shown in Figure 7) and 249 unique variations across object placement, color, size, category, count, and shape. Here we give a summary of the 18 RLBench tasks in Table 6. The extra 6 RLBench tasks (shown in Figure 8) for the few-shot adaptation experiment are summarized in Table 7.

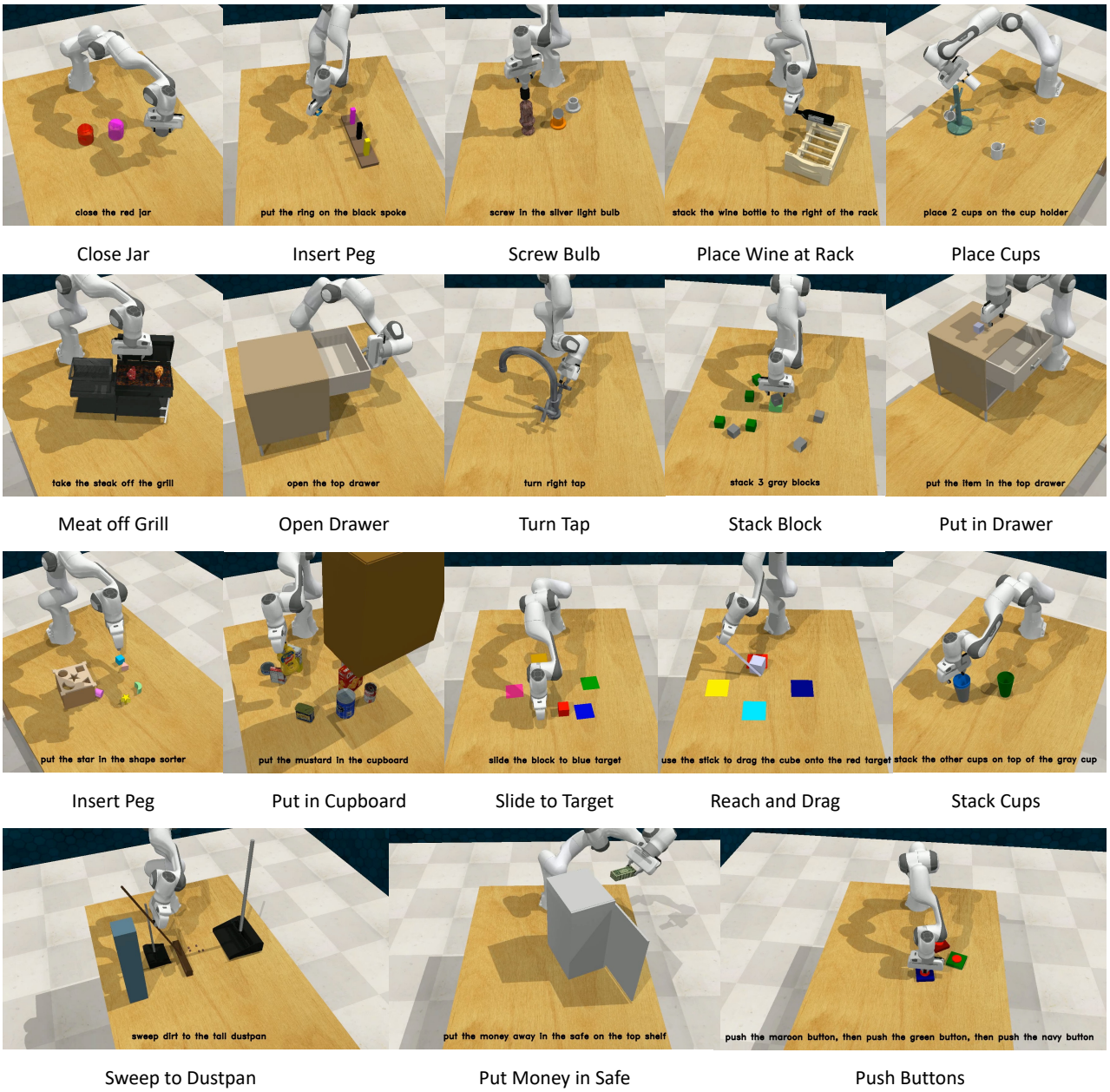

Figure 7. **Language-Conditioned Manipulation Tasks in RLBench**. We conduct multi-task experiments on 18 simulated tasks in RLBench(James et al., 2020). Apart from the language instruction depicted in the figures, there are a total of 249 variations of these tasks, as illustrated in Table 6. During the test, the agent needs to handle the novel object poses, randomly sampled goals, and randomly sampled scenes with different semantic instantiations of object colors, shapes, sizes, and categories within a maximum of 25 execution steps.

Table 6. The 18 RLBench tasks for multi-task experiment

Task name	Language Template	Avg. Keyframes	#of Variations	Variation Type
put in drawer	"put the item in the __ drawer"	12.0	3	placement
reach and drag	"use the stick to drag the cube onto the __ target"	6.0	20	color
turn tap	"turn __ tap"	2.0	2	placement
slide to target	"slide the block to __ target"	4.7	4	color
open drawer	"open the __ drawer"	3.0	3	placement
put in cupboard	"put the __ in the cupboard"	5.0	9	category
place in shape sorter	"put the __ in the shape sorter"	5.0	5	shape
put money in safe	"put the money away in the safe on the __ shelf"	5.0	3	placement
push buttons	"push the __ button, [then the __ button]"	3.8	50	color
close jar	"close the __ jar"	6.0	20	color
stack block	"stack __ __ blocks"	14.6	60	color,count
place cups	"place __ cups on the cup holder"	11.5	3	count
place wine at rack	"stack the wine bottle to the __ of the rack"	5.0	3	placement
screw bulb	"screw in the __ light bulb"	7.0	20	color
sweep to dustpan	"sweep dirt to the __ dustpan"	4.6	2	size
insert peg	"put the ring on the __ spoke"	5.0	20	color
meat off grill	"take the __ off the grill"	5.0	2	category
stack cups	"stack the other cups on top of the __ cup"	10.0	20	color

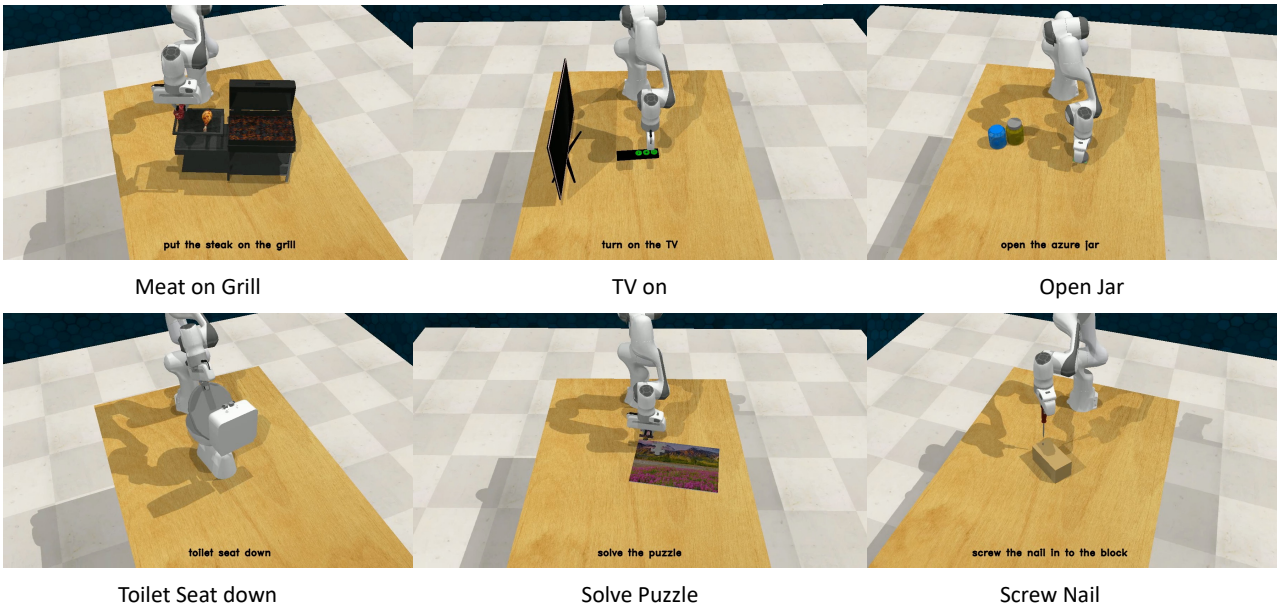

Figure 8. **Language-Conditioned Manipulation Tasks in RLBench**. We conduct few-shot adaptation experiments on 6 simulated tasks in RLBench to evaluate the generalization ability of SAM-E. Task variations are shown in Table 7. The tasks must be completed by the agent within a maximum of 25 steps.

B. Implementation Details

In this section, we provide more implementation details of SAM-E.

B.1. Visual Input

In our experiments of RLBench, the visual observations are captured by four cameras (left shoulder, right shoulder, front, and wrist) with a resolution of 128×128 in RGB-D. We follow the re-render approach introduced by RVT (Goyal et al.,

Table 7. The 6 RLBench tasks used for the few-shot adaptation experiments.

Task name	Language Template	Avg. Keyframes	#of Variations	Variation Type
meat on grill	"put the __ on the grill "	5.0	2	category
open jar	"open the __ jar"	6.0	20	color
screw nail	"screw the nail in to the block"	6.0	1	-
toilet seat down	"toilet seat down"	4.7	1	-
tv on	"turn on the TV"	8.0	1	-
solve puzzle	"solve the puzzle"	5.0	1	-

2023) before feeding visual images to the model. Specifically, the RGB-D images are rerendered to generate virtual images in the form of cube orthographic projection. Then we use the cube orthographic projections as the visual inputs of SAM-E.

B.2. Action Sequence Imitation

We utilize a multi-channel action sequence policy head to predict the action sequence, trained by action sequence imitation. To extract the temporal information of actions from the expert demonstrations, we employ the keyframe extraction on each demonstration, generating a dataset of keyframe sequences. Given observations, SAM-E generates an action sequence with a default action horizon of 5 and is trained to maximize the likelihood objective of imitation learning. Note that the action sequence data may have variable lengths, when the data is shorter than the action horizon, we mask the untrained action head, and when the data is longer, we truncate it accordingly.

B.3. Hyperparameters

In our experiments, the hyperparameters are primarily fixed, as shown in Table 8.

Table 8. Training Hyperparameters

Hyperparameters	Multi-task Training	Few-shot adaptation
batch size	10	10
learning rate	4e-3	4e-3
optimizer	LAMB	LAMB
learning rate schedule	cosine decay	cosine decay
warmup steps	2000	2000
training steps	60K	4K
training epochs	15	1

C. Visualization

We visualize the attention map of the multi-view transformer to show SAM-E's various attention patterns for task comprehension and action sequence prediction. We use task *put_item_in_drawer* as an example, which is completed by three executions.

(i) In the first execution with the initial observation (see Figure 9), SAM-E's attention, from one of its heads, is predominantly focused on the Franka robot, the drawer cabinet, and more specifically, the item on the cabinet and the handle of the top drawer. This observation aligns with the given instruction to '*put the item in the top drawer*', highlighting SAM-E's capability to identify key objects within the scene according to the task description for task execution.

(ii) In the second inference, following an action sequence that results in the opening of the top drawer, SAM-E adapts its focus. It now observes the newly available space within the drawer for placing the item (see Figure 10). Concurrently, another of its attention heads redirects back to the end-effector and the item, strategizing the subsequent action of picking up and placing the item into the drawer(see Figure 11).

(iii) In the final inference (see Figure 12), SAM-E concentrates on the end-effector picking up the item and positioning it accurately into the target position. This phase likely involves precise adjustments and movements, ensuring the successful completion of the task as the language instruction.

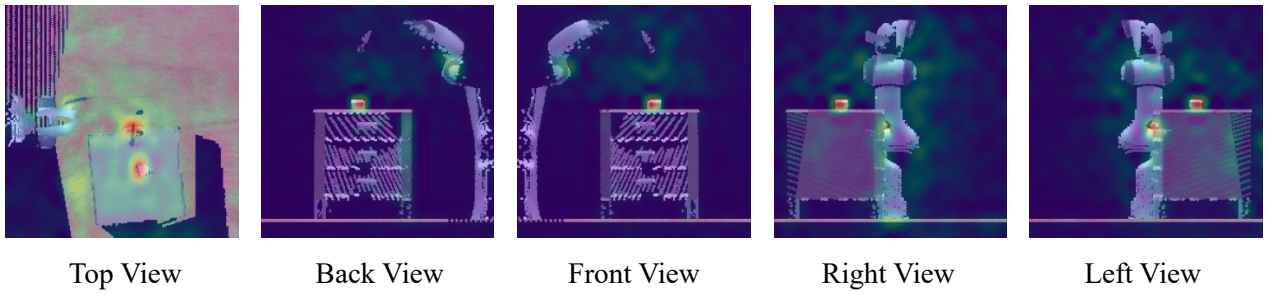

Figure 9. SAM-E's multi-view attention map of the initial inference.

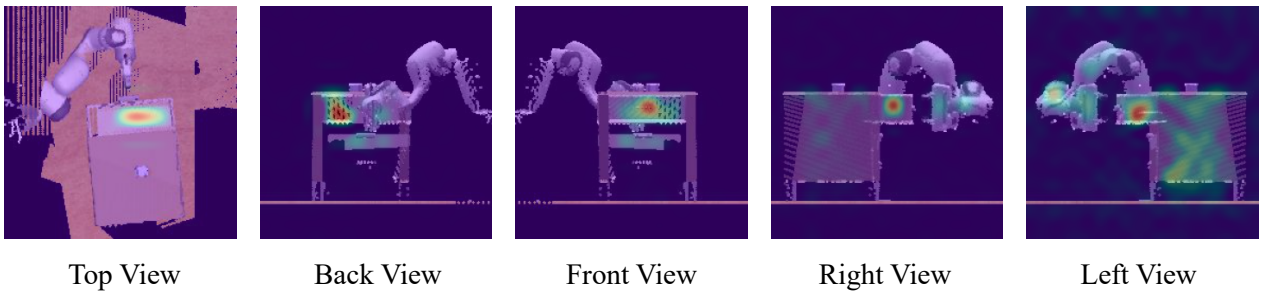

Figure 10. SAM-E's multi-view attention map of the second inference, focusing on the open drawer.

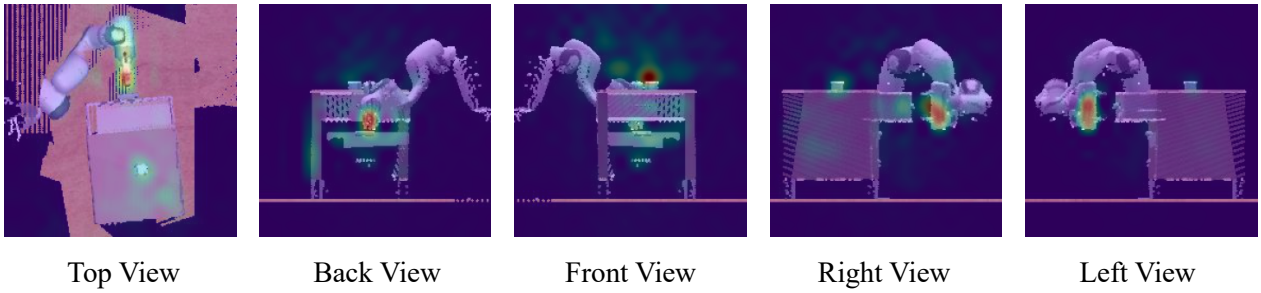

Figure 11. SAM-E's multi-view attention map of the second inference, focusing on the end-effector and the item.

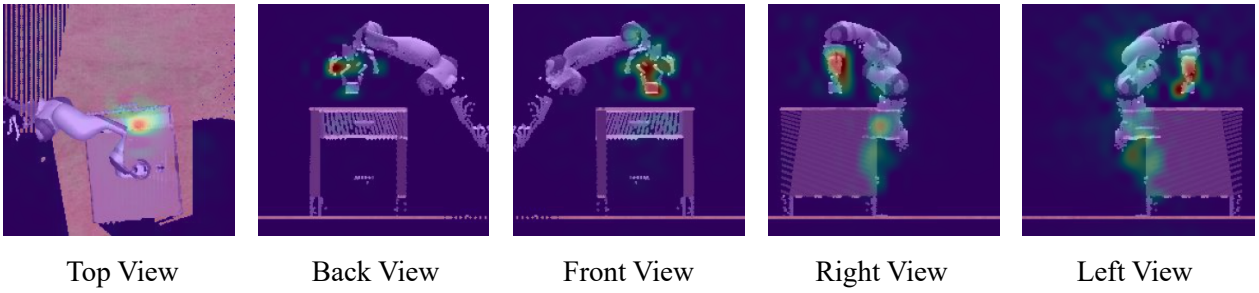

Figure 12. SAM-E's multi-view attention map of the last inference.

D. One Glance Results

Figure 13 shows the results of execution times of SAM-E on success cases of several tasks. Thanks to its promptable perception and efficient action sequence prediction, SAM-E excels in task completion by executing actions coherently, resulting in improved performance and significantly reduced inference requirements. For the following tasks, in comparison, RVT requires an average of $[6.4, 6.0, 3.8, 5.5, 3.7, 4.3, 5.5, 5.0, 4.8]$ execution times of its success cases.

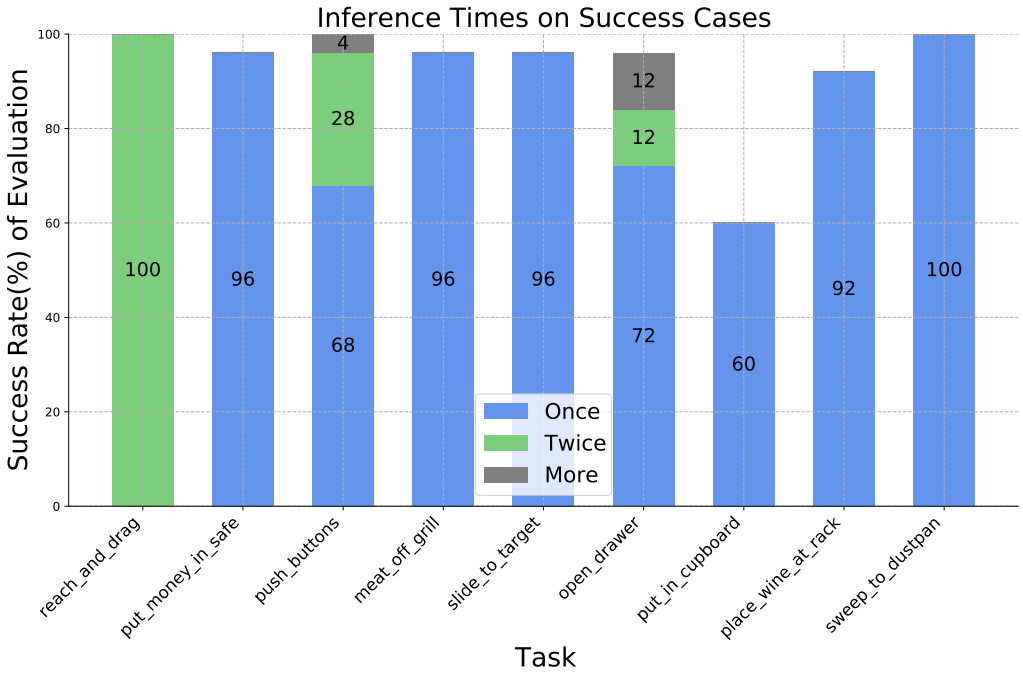

Figure 13. The comparison of execution times on success cases of several tasks.

E. Comparison with Hiveformer

To compare the performance of SAM-E and HiverFormer, we add experiments to train SAM-E with the same 10 tasks evaluated in the Hiveformer paper with 100 demonstrations per task, which is the same as Hiveformer (results are shown in Table 9). The score of Hiveformer is adapted from their original paper. We remark that SAM-E is trained with 10 tasks with **all variations**, which is much more challenging than Hiveformer which is trained with a unique variation for each task.

Table 9. **Comparison with Hiveformer.** Scores of Hiveformer are adopted from Guhur et al. (2022). Mean and std of 5 evaluations are reported.

Models	Pick and Lift	Pick up Cup	Put Knife on Board	Reach Target	Stack Wine	Take Money out Safe	Take Umbrella out Stand	Push Buttons	Put Money in Safe	Slide to Target	**On Average**
Hiveformer	88.9	92.9	75.3	100.0	71.2	79.1	89.2	100.0	58.2	78.7	83.3
SAM-E	87.2±1.8	88.8±5.2	68.0±4.0	100.0±0.0	69.6±7.3	98.4±2.2	96.0±2.8	100.0±0.0	93.6±2.2	84.8±7.7	**88.6**±0.7

F. Ablation

We provide the complete results of the ablation study in Table 10 and Table 11.

G. Real-World Experiments

We conduct real-world experiments on a FranKa Panda robot arm in the real world, equipped with a dual RGB-D camera setup positioned at the left front and right front for multi-view observation, shown in Figure 14. We construct the real-world scene and design 5 tasks for experiments, including *put the towel on the cabinet*, *stack the block*, *close the drawer*, *pick up*

Table 10. **Ablation Performances of SAM-E's variations.** Mean and std of 5 evaluations are reported.

Models	Put in Drawer	Reach and Drag	Turn Tap	Slide to Target	Open Drawer	Put in Cupboard	Place in Shape Sorter	Put Money in Safe	Push Buttons	Close Jar
SAM-E	92.0±5.7	**100.0**±0.0	**100.0**±0.0	**95.2**±1.8	**95.2**±5.2	**64.0**±2.8	34.4±6.1	**95.2**±3.3	**100.0**±0.0	82.4±3.6
SAM-E (LoRA, QKV)	88.8±7.7	**100.0**±0.0	**100.0**±0.0	90.4±6.1	94.4±3.6	57.6±4.6	**37.6**±5.4	92.8±5.2	**100.0**±0.0	78.4±4.6
SAM-E (w/o LoRA)	84.8±7.2	**100.0**±0.0	**100.0**±0.0	92.0±4.0	92.8±3.3	52.0±2.8	31.2±5.2	92.8±1.8	98.4±2.2	**87.2**±5.2
SAM-E (full finetune)	**93.3**±4.6	98.7±2.3	**100.0**±0.0	69.3±11.5	90.7±2.3	52.0±0.0	29.3±11.5	89.3±2.3	**100.0**±0.0	68.0±0.0

Models	Stack Blocks	Place Cups	Place Wine at Rack	Screw Bulb	Sweep to Dustpan	Insert Peg	Meat off Grill	Stack Cups	**On Average**	**Inference Steps(Sum)**
SAM-E	26.4±4.6	0.0±0.0	**94.4**±4.6	**78.4**±3.6	**100.0**±0.0	**18.4**±4.6	95.2±3.3	0.0±0.0	**70.6**±0.7	**1130**±12
SAM-E (LoRA, QKV)	**32.0**±4.9	**3.2**±1.8	93.6±3.6	69.6±4.6	98.4±2.2	6.4±6.1	97.6±2.2	5.6±2.2	69.2±0.9	1142±6
SAM-E (w/o LoRA)	20.8±7.2	0.0±0.0	92.0±4.0	64.0±7.5	96.8±1.8	8.8±6.6	94.4±3.6	0.8±1.8	67.2±1.0	1182±6
SAM-E (full finetune)	20.0±6.9	1.3±2.3	90.7±2.3	69.3±4.6	**100.0**±0.0	0.0±0.0	**98.7**±2.3	**13.3**±4.6	65.8±1.0	1204±18

Table 11. **Ablation Performances with different action sequence length.** Mean and std of 5 evaluations are reported.

Models	Put in Drawer	Reach and Drag	Turn Tap	Slide to Target	Open Drawer	Put in Cupboard	Place in Shape Sorter	Put Money in Safe	Push Buttons	Close Jar
SAM-E ($h = 1$)	0.0±0.0	6.7±2.3	98.7±2.3	45.3±4.6	72.0±6.9	8.0±4.0	14.7±2.3	8.0±0.0	69.3±2.3	12.0±4.0
SAM-E ($h = 3$)	77.6±2.2	84.8±1.8	**100.0**±0.0	72.8±1.8	92.0±2.8	31.2±5.2	**35.2**±3.3	84.8±5.2	99.2±1.8	73.6±3.6
SAM-E ($h = 5$)	**92.0**±5.7	**100.0**±0.0	**100.0**±0.0	**95.2**±1.8	**95.2**±5.2	**64.0**±2.8	34.4±6.1	**95.2**±3.3	**100.0**±0.0	**82.4**±3.6
SAM-E ($h = 7$)	88.8±9.5	99.2±1.8	**100.0**±0.0	80.8±18.6	90.4±4.6	53.6±7.3	28.8±3.3	92.8±1.8	**100.0**±0.0	72.8±3.3

Models	Stack Blocks	Place Cups	Place Wine at Rack	Screw Bulb	Sweep to Dustpan	Insert Peg	Meat off Grill	Stack Cups	**On Average**	**Inference Steps(Sum)**
SAM-E ($h = 1$)	0.0±0.0	1.3±2.3	40.0±6.9	58.7±2.3	24.0±4.0	34.7±14.0	54.7±4.6	2.7±4.6	30.6±1.4	8329±60
SAM-E ($h = 3$)	16.8±3.3	1.6±2.2	76.8±4.4	49.6±6.7	87.2±1.8	**54.4**±2.2	**100.0**±0.0	**13.6**±3.6	64.0±0.6	2026±30
SAM-E ($h = 5$)	**26.4**±4.6	0.0±0.0	**94.4**±4.6	**78.4**±3.6	**100.0**±0.0	18.4±4.6	95.2±3.3	0.0±0.0	**70.6**±0.7	1130±12
SAM-E ($h = 7$)	13.6±5.4	**3.2**±3.3	92.0±4.0	70.4±5.4	**100.0**±0.0	8.8±4.4	97.6±3.6	4.8±1.8	66.5±1.2	**919**±12

the banana, and *put the orange into the drawer*. For data collection, we manually control the robot arm for demonstrations by a controller and collect the RGB-D stream and robot joint pose simultaneously with a data collection pipeline. We collect demonstrations with variations in item placement for all tasks. See https://sam-embodied.github.io/ for videos and performance.

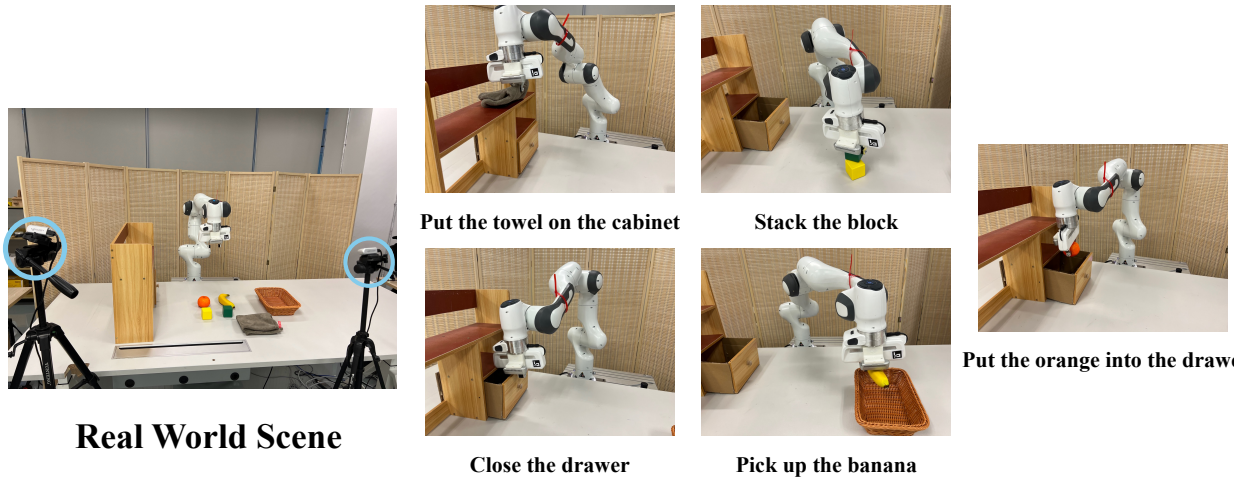

Figure 14. Real-World Scene and tasks.

H. Limitation and Future Work

In this work, we propose SAM-E with SAM as the visual foundation and action-sequence policy head, which outperforms prior state-of-the-art methods. However, we also identify limitations that suggest directions for future research. We employ parameter-efficient fine-tuning on relatively limited robot data to enhance its understanding of embodied manipulation. Future improvements might include leveraging the scalability of the visual foundation through training on larger datasets, such as Open-X (Collaboration, 2023). Additionally, we employed a fixed horizon for the action-sequence policy, which, while generally effective, could be less suitable for certain tasks, such as *stack cups* in our experiments, in which may need to pay more attention to the trade-off between precision and coherence of the action. It would be intriguing to see the action horizon optimized through a mechanism or learned from data.