Towards Fast, Memory-based and Data-Efficient Vision-Language Policy

Anonymous ICCV submission

Paper ID 14405

Abstract

Vision Language Models (VLMs) pretrained on Internet-scale vision-language data have demonstrated the potential to transfer their knowledge to robotic learning. However, the existing paradigm encounters three critical challenges: (1) expensive inference cost resulting from large-scale model parameters, (2) frequent domain shifts caused by mismatched data modalities, and (3) limited capacity to handle past or future experiences. In this work, we propose LiteVLP, a lightweight, memory-based, and general-purpose vision-language policy generation model. LiteVLP is built upon a pre-trained 1B-parameter VLM and fine-tuned on a tiny-scale and conversation-style robotic dataset. Through extensive experiments, we demonstrate that LiteVLP outperforms state-of-the-art vision-language policy on VIMA-Bench, with minimal training time. Furthermore, LiteVLP exhibits superior inference speed while maintaining exceptional high accuracy. In long-horizon manipulation tasks, LiteVLP also shows remarkable memory ability, outperforming the best-performing baseline model by 18.8%. These results highlight LiteVLP as a promising model to integrating the intelligence of VLMs into robotic learning.

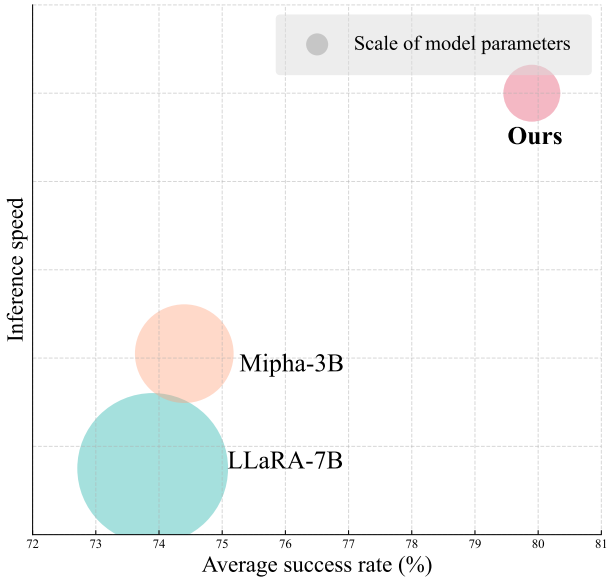

Figure 1. **Comparative performance of vision-language policies.** The x-axis represents the average success rate on VIMA-Bench, and the y-axis represents the inference speed evaluated on same devices. The bubble diameter indicates the number of model parameters

1. Introduction

Integrating pre-trained Large Language Models (LLMs) and VLMs with low-level robotic policies enables context-aware robotic systems and enhances the robot's ability to reason and interact with the environment [11, 13, 15, 29, 44, 57, 61–63]. Recently, the robotics domain has increasingly explored how to leverage LLMs and VLMs for various robotic tasks such as perception, prediction, planning, and control [22]. However, despite the growing interest in these models, fully unlocking the capability of LLMs and VLMs in robotics remains a great challenge.

Due to their remarkable performance in decision making and task reasoning, LLMs and VLMs have been widely adopted for hierarchical task planning [1, 17, 23, 59] and code generation [18, 26, 35, 50], facilitating guidance and interaction with low-level robotic controllers. To construct end-to-end models that generate robot actions directly from observations in natural vision and language, recent research has focused on developing Vision-Language-Action (VLA) models [9, 11, 29, 43, 56–58, 62] that learn from web and robotics data. Some VLA models leverage pre-trained LLMs and VLMs, which represent actions as simple text strings and tokenize them into text tokens with a tokenizer [9, 29, 43], while others utilize LLMs and VLMs to compress multimodal representations and train additional action actors to refine action trajectory outputs [6, 11, 38, 47].

However, there are three key challenges existing in the above-mentioned models. First, fine-tuning pre-trained LLMs and VLMs with robotic data often encounters frequent domin shifts due to the substantial differences be-

tween the pre-training web dataset and the fine-tuning robotic dataset. Second, current models suffer from insufficient memory for future or past experiences, making them achieve poor performance for long-horizon manipulation. Third, the large number of model parameters in the backbone network leads to a high computational time, which limits their real-world deployment.

To solve the above challenges, in this paper, we introduce LiteVLP, a lightweight vision-language policy tailored to handle memory-dependent robotic tasks with a stable fine-tuning strategy that effectively leverages pre-trained knowledge. LiteVLP fine-tunes a pre-trained VLM backbone on a conversation-style robotic dataset via visuomotor instruction tuning training paradigm [33]. Due to the high structural alignment between robotic data and vision-language pre-training data, LiteVLP effectively mitigates frequent domin shifts during fine-tuning, resulting in greater training stability compared to traditional VLA models. To address long-horizon and memory-dependent tasks, LiteVLP supports multi-image input to better leverage past and future experiences. Meanwhile, LiteVLP introduces the RLT [16] in video transformers [3, 5, 20, 34] and modifies it as a multi-observation compression (MOC) module in our model to reduce the spatio-temporal complexity of the model. As a result, the number of image tokens decreases by approximately 60%, while preserving the representational quality of the image patches. Furthermore, LiteVLP uses InternVL2-1B [14] as its backbone, and the small parameters of the model make its training and inference more efficient compared to previous models. All of these characteristics make LiteVLP not only demonstrate computational efficiency, but also have great potential to improve real-time performance and scalability in long-horizon robotic manipulation tasks.

We conduct extensive experiments and demonstrate that LiteVLP can successfully handle a variety of robot manipulation tasks by fine-tuning a lightweight VLM on a small robotic manipulation dataset. It achieves competitive results on the VIMA-Bench [27]. In general, our contributions can be summarized as follows:

- We propose LiteVLP, the first vision-language policy with efficient inference, memory effects, and fine-tuning stability. It employs a lightweight VLM as the backbone, supports future goals or past experiences as input, and fine-tunes with conversation-style training manner.
- We introduce a novel MOC module in LiteVLP to improve training efficiency and accelerate inference speed. This module effectively reduces the number of image tokens, increasing inference speed by 34.1%.
- We also conduct extensive experiments on 17 simulated tasks from VIMA-Bench. With equitable training on the same small dataset, LiteVLP outperforms the state-of-the-art vision-language policy while delivering a 70% reduction of training time and 6.8 times acceleration of inference speed. Furthermore, in long-horizon tasks, LiteVLP achieves a 18.8% higher success rate compared to baselines.

2. Related Work

LLMs and VLMs in Robot Learning Recently, VLMs have made significant progress in various tasks. Models such as InternVL [14], Qwen2-VL [54], Flamingo [2], BLIP-2 [31], Mipha [64] and LLaVA [36] have demonstrated their remarkable potential and application in multimodal learning tasks. With the rise of embodied intelligence, integrating LLMs and VLMs into robotics has become a promising research direction, catalyzing the development of both generalist robot policies [6, 8, 10, 29, 38, 51] and VLA models [25, 29, 32, 41, 46, 49, 56, 58, 62]. However, existing generalist robot policies or VLA models typically have a large number of parameters, as they are fine-tuned from large VLMs, resulting in high training and deployment costs. The method proposed in this paper demonstrates that fine-tuning a small-parameter VLM can achieve comparable performance to large-scale generalist robot policies while significantly reducing computational costs and improving inference efficiency.

Memory-based Robotic Manipulation In robotics domain, enabling robots to think human-like is a long-standing and unsolved challenge. Therein, equipping robots with memory capabilities is particularly crucial, as it allows robots to store and retrieve historical experiences, thus enhancing their ability to perform complex tasks efficiently [28, 53]. Early studies on robotic memory-based manipulation mainly addressed navigation using constrained semantic maps [7, 12], while other studies focused on designing memory models and representations within cognitive architectures. Recent advancements like SAM2Act+ [21] inspired by SAM2 [48], incorporate a memory bank, an encoder, and an attention mechanism to improve spatial memory for robotic tasks. A slight difference between the above methods and our work is that LiteVLP not only integrates the memory of historical experiences but also future goal states, which enables the robot to handle long-horizon tasks more effectively.

Robotic Manipulation Recently, some studies have attempted to integrate visual observations with robotic proprioception to achieve precise predictions for robotic manipulation, such as Where2Act [40], Flowbot3d [19], Partnet [39] and Sparsedff [55]. With the rapid advancement of multimodal large language models (MLLMs) [4, 30, 31, 36], their powerful reasoning capabilities have introduced new ways to solve robotic manipulation. OpenVLA [29] is an open-source VLA model that integrates Llama2 [52] with DINOv2 [45] and SigLIP [60], achieving strong gen-

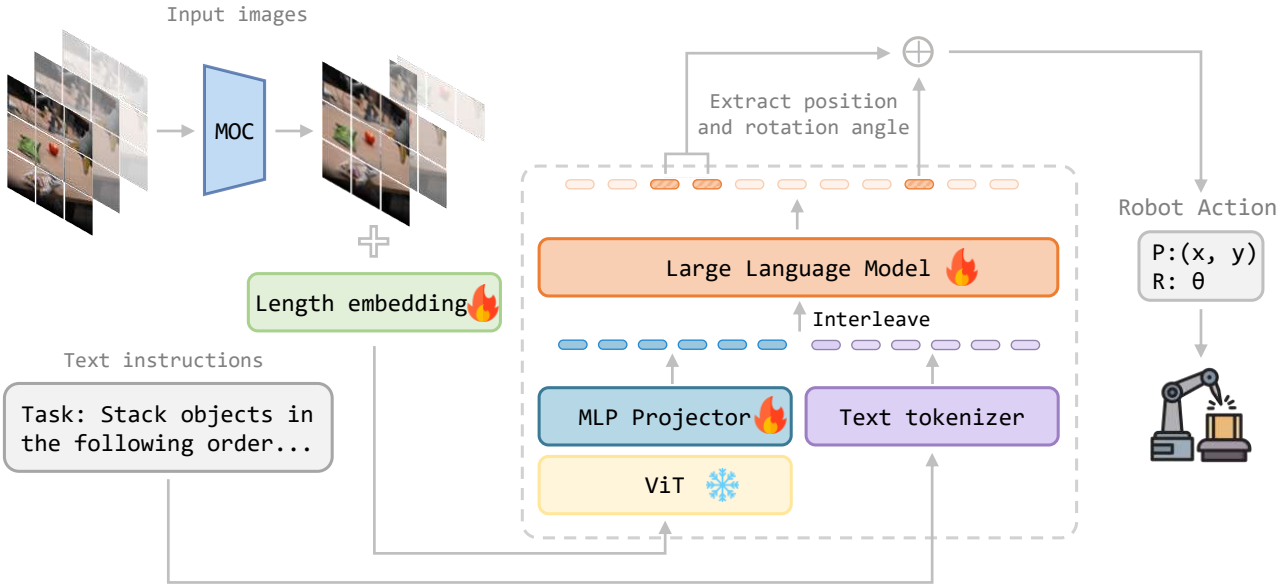

Figure 2. **Overall framework of LiteVLP.** The LiteVLP initiates with multi-observation compression and then projects the image features into the same dimensional space as the text features. Subsequently, the image tokens are interleaved with text tokens and processed by a large language model to generate a text output that includes the end-effector's action. Of note, during the fine-tuning stage, the parameters of the ViT are frozen, while the length embedding, the MLP projector and the large language model are trained.

eralist manipulation capabilities. 3D-VLA [61] introduces a new family of embodied foundation models based on a 3D LLM [24], seamlessly linking 3D perception, reasoning, and action through a generative world model. π_0 [6] is a novel architecture based on a pre-trained VLM and a flow matching [37] action expert, enabling robots to achieve strong generalization capabilities and execute complex and highly dexterous tasks. However, these models typically rely on large-parameter VLM backbones and require training on extremely large-scale datasets. In contrast, our model utilizes only a 1B-parameter pre-trained VLM, achieving comparable or even superior performance to large-scale models while maintaining high data efficiency.

3. Method

3.1. Problem Formulation

The problem of robot manipulation conditioned on vision-language can be formally modeled as a structured decision-making task that generates the current action of the robot based on the visual observations, task instructions, and future goal images given. Let $\mathcal{M} = \langle \mathcal{S}, \mathcal{O}, \mathcal{G}, \mathcal{A} \rangle$ represent the overall problem framework. Here:

- $\mathcal{S}$ is the state space, where a state $s_t \in \mathcal{S}$ represents the configuration of the robot and the environment at timestamp t.
- $\mathcal{O}$ is the observation space, where $o_t \in \mathcal{O}$ denotes the observation at timestep t, consisting of both visual and textual information about the environment.
- $\mathcal{G}$ is the goal condition, which includes a set of goal images and textual instructions that define the desired outcome of the task.
- $\mathcal{A}$ is the action space, where an action $a_t \in \mathcal{A}$ represents the robot action at timestamp t.

In the context of the generation of vision-language policy, the goal is to generate a sequence of actions that enable the robot to perform the task specified by the goal condition $\mathcal{G}$. Our model LiteVLP can be defined as a policy function π that maps the current state s_t, visual observation o_t and goal condition g_t to an action a_t, described as:

$$a_t = \pi(s_t, o_t, g_t; \theta) \tag{1}$$

where θ represents the parameters of the policy model. After executing each action, the new observation o_{t+1} is obtained. The state s_{t+1} is updated by executing the action a_t and the state s_t. The process continues iteratively until the goal condition $\mathcal{G}$ is satisfied or a termination condition is met.

3.2. Model architecture

In this work, our goal is to develop an end-to-end vision-language robot policy that fully utilizes the strong semantic reasoning ability of VLMs to instruct robotic manipulation. The architecture of our model is illustrated in Fig. 2. At timestamp t, our model π accepts multiple images $I_{N,t} \in R^{N \times W \times H \times 3}$ and language instruction L_t as

input, finally generating a pure language answer $L_{a,t}$ that contains the robot's impending actions. Here, N is the number of images, W and H are the width and height of the image respectively. Before $I_{N,t}$ are inserted into the corresponding positions in the text, they first undergo the MOC module. After an extraction process Ext, the 2D image coordinates $C_{i,t}$ and the rotation angle $R_{i,t}$ of the end effector can be extracted from answer $L_{a,t}$, which define a robot's action $A_t(C, R)$. Meanwhile, the 2D image coordinates undergo a transformation into the robot's action space through a predefined mapping process, which can be determined via visual calibration in both simulated and real-world settings.

Once the above round is completed, the robot performs the specified action, capturing a new observation for subsequent processing. The entire process can be represented by the following formulation:

$$L_{a,t} = \pi\left(\text{MOC}\left(I_{N,t}\right), L_t\right) \tag{2}$$

$$A_t(C, R) = Ext\left(L_{a,t}\right) \tag{3}$$

where $A_t(C, R) = \{\mathbf{q}_{i,t} = (x_{i,t}, y_{i,t}, \theta_{i,t}) \mid i = 1, \ldots, N_A\}$. Meanwhile, $C_{i,t} = (x_{i,t}, y_{i,t})$, $R_{i,t} = \theta_{i,t}$, $N_{A,t}$ represents the number of actions generated.

3.3. Memory-based Visuomotor Instruction Tuning

We adopt the visuomotor instruction tuning method proposed in LLaRA [33] to fine-tune LiteVLP, leveraging its effectiveness in addressing distribution shifts when fine-tuning pre-trained VLMs with robotic data. In multi-step tasks, previous actions $A_{1:t-1}$ and their outcomes or historical observations $O_{1:t-1}$ are critical to generate subsequent actions. Notably, I_N already consists of $O_{1:t-1}$. Otherwise, we explicitly append $A_{1:t-1}$ to L_t in the form like "You have finished ...", enabling the model to obtain the whole process of the task. This integration of historical information preserves the model's memory for long-horizon tasks, enhancing its decision-making capability in sequential actions.

3.4. Multi-observations compression

We observe a common characteristic in robot manipulation tasks: there is a significant amount of redundant image information between multiple consecutive observation images. Our goal is to identify whether patches at the same spatial location in two consecutive images are static. These static patches correspond to background elements or objects in the images that do not change over time. The results after the removal of static patches are shown in Fig. 3. To optimize efficiency, we replace subsequent static patches with the patch that first appeared. Consequently, each compressed patch requires a length embedding to indicate how many images it persists across. The process of implementing the above method is as follows: First, we compare image patches at the same spatial location across consecutive images and set a threshold ϵ. If this condition is met, we consider $P^i_{x,y}$ to be static:

$$\left\|P^{i+1}_{x,y} - P^i_{x,y}\right\| < \epsilon \tag{4}$$

Thus, we can obtain an image patch mask M^N_S to indicate static patches and a run-length mask M^N_L to represent the durative length of each static patch. The multi-image compression process can be formulated as follows:

$$\text{MOC}\left(I_{N,t}\right) = M^N_S \circ P\left(I_{N,t}\right) + M^N_L \odot Emb_L \tag{5}$$

Here, P represents the operation of performing position embedding on the image and Emb_L is a learnable length embedding parameter.

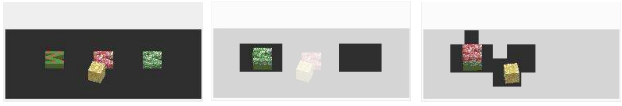

Figure 3. **Simple visualization of MOC's effect.** The light gray image patches indicate unchanged areas between consecutive images, which will be reduced in the sequence of image patches.

4. Experiments

4.1. Experimental Setup

Benchmark VIMA-Bench [27] is a simulated tabletop robotic manipulation environment for evaluating VLMs using multimodal instructions. It includes 17 tasks, each with text and reference images guiding the robot's actions, such as `follow order` and `rearrange`. The robot's action space consists of 2D coordinates for placement and quaternions for rotations. VIMA-Bench uses a four-level protocol to assess generalization: placement (L1), combinatorial (L2), novel object (L3), and novel task generalization (L4). This setup provides a comprehensive evaluation of VLMs in robotic manipulation.

Training datasets We use the dataset introduced in LLaRA, which provides three datasets of varying sizes. For our training, we select the Description-Instruct-BC-8k (D-inBC-8k) dataset. We present an example of D-inBC in Fig. 4. Note that the inputs of LiteVLP-m and LiteVLP-s are diverse. The input of LiteVLP-m includes complete images and the corresponding object detection results. The complete images comprise historical observations, current observations, and future goal images. However, the input of LiteVLP-s only includes current observation. And the way LiteVLP-s obtains the information of complete images is through the object detection results.

Evaluation tasks VIMA-Bench provides tasks with difficulty levels ranging from L1 to L4. We select L1 to

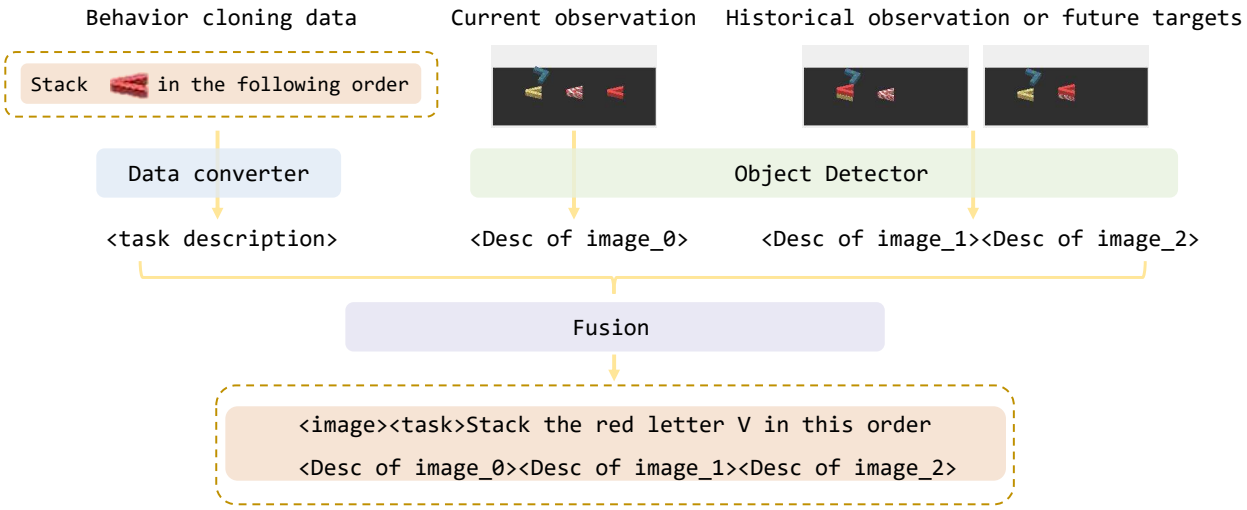

Figure 4. **Example of D-inBC dataset format.** The D-inBC dataset includes the task description and the description of reference images. A data converter processes instructions, while an object detector extracts the locations of each object to form the image descriptions.

L3 as test tasks to evaluate the performance of our model. The reason we abandon the L4 task is the mismatch of spatula's rotation data between training and testing phases. This inconsistency can significantly affect the success rate of the L4 task, so we decided to only use L1 to L3 as our test tasks [33].

Implementation details During the fine-tuning stage, we freeze the vision encoder parameters and train the MLP layers, the large language model, and the newly introduced length embedding layers in MOC module. The hyperparameters are listed in Tab. 1. We train LiteVLP on D-inBC-8k for 4 epochs. We use the AdamW optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$, a learning rate of 2e-5, a weight decay of 0.01, and a warm-up ratio of 0.03. The learning rate schedule is CosineAnnealingLR. Furthermore, in the MOC module, we set the threshold ϵ to 1e-5 during the fine-tuning and inference stages to maximize the retention of informative image patches from observations.

Hyperparameter	Value
learning rate	2e-5
epochs	4
optimizer	AdamW
weight decay	0.01
warming up	2e-5
lr schedule	CosineAnnealingLR
ϵ	1e-5

Table 1. **Hyperparameters used during fine-tuning.** This table presents the hyperparameters during model fine-tuning. Here, the parameter ϵ represents the threshold in the MOC module.

4.2. Evaluation Results and Analysis

Evaluation results We evaluate our method on VIMA-Bench and compare its performance against other models trained on datasets of different sizes. The results are shown in Tab. 2. We demonstrate that LiteVLP achieves highly competitive performance, achieving a success rate of 84.2% on L1, 78.1% on L2, and 75.4% on L3, using only 1.2% of the VIMA dataset. This performance is comparable to the state-of-the-art model VIMA, which achieves 81.5% on L1, 81.5% on L2, and 78.7% on L3 when trained on the full dataset. Additionally, our model significantly outperforms other VLMs fine-tuned on our small dataset, such as LLaRA-7B and Mipha-3B. These results successfully indicate that LiteVLP can rapidly adapt to robotic manipulation and demonstrate highly competitive performance when fine-tuned with visuomotor instruction in a small robotic dataset. The success rates of all tasks at three difficulty levels are shown in Fig. 5.

Robustness to object location We simulate the scenario of inaccurate object localization in real-world applications to test the model's performance stability, with the results shown in Fig. 6. By adding random noise of varying sizes to the object location coordinates, we assess the average decline in task success rate. As shown in the table, with a noise level of 0.2, LiteVLP-m drops by 5.3% and LiteVLP-s by 7.8%. And when the noise level increases to 0.8, the drops of LiteVLP-m and LiteVLP-s rise to 9.6% and 22.2% each. Although the average performance drops for both models as the noise level increases, compared to LiteVLP-s, LiteVLP-m drops much smaller in task success rate. This is because LiteVLP-s can only obtain object location information from the object detection text description, while

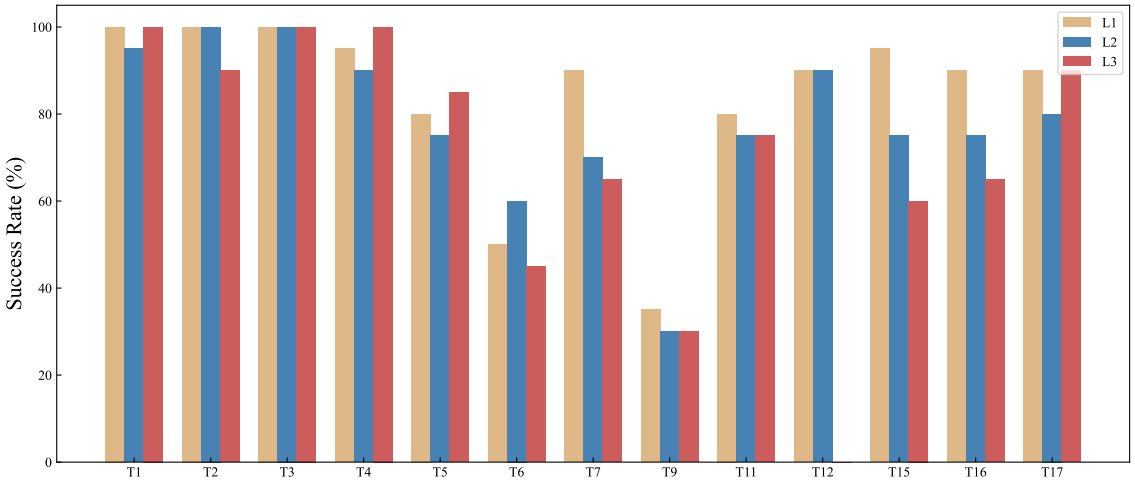

Figure 5. **Success rates of all tasks.** Note that the difficulty level L3 doesn't include `sweep without exceeding` task.

Model	Data	L1 (%)	L2 (%)	L3 (%)
VIMA	100%	81.5	**81.5**	**78.7**
VIMA	10%	76.3	75.8	73.2
VIMA	1%	36.3	34.3	15.4
LLaRA-7B-m	1.2%	44.6	27.7	34.2
LLaRA-7B-s	1.2%	78.1	73.8	70.0
Mipha-3B	1.2%	78.8	72.3	72.1
LiteVLP-s	1.2%	83.1	**81.5**	77.5
LiteVLP-m	1.2%	**84.2**	78.1	75.4

Table 2. **Evaluation results on VIMA-Bench.** Comparison of success rates over different models trained on datasets of varying sizes. For brevity, the suffix '-m' in model names denotes multi-image input, while '-s' indicates single-image input.

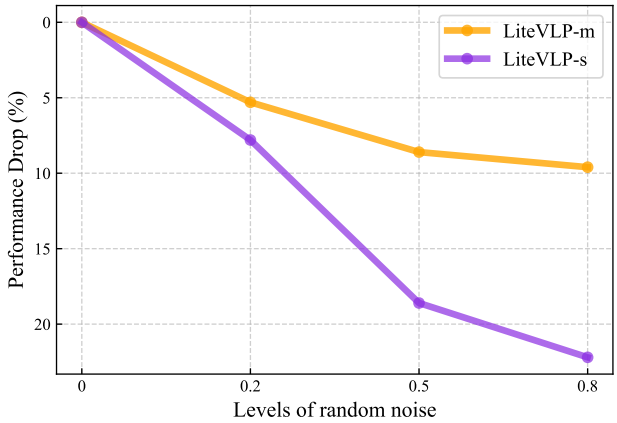

Figure 6. **Robustness performance.** The figure illustrates the performance drop under noise levels of 0.2, 0.5, and 0.8.

LiteVLP-m can obtain location information both from the text description and the image data across multiple views, effectively enhancing the model's robustness to location inaccuracies.

Long-horizon manipulation performance In long-horizon manipulation tasks, LiteVLP-m significantly outperforms other baseline models , the results are shown in Tab 3. We refer to CoTDiffusion [42] to select three representative long-horizon manipulation tasks in VIMA-Bench —`visual rearrangement`, `visual reasoning`, and `visual constraints`. CoT-Diffusion is a model specifically designed to improve performance in long-horizon manipulation tasks. However, our LiteVLP-m demonstrates superior performance in long-horizon manipulation tasks, achieving an average improvement of 18.8% over CoTDiffusion in three types of tasks. The effectiveness of our model can be attributed to the sufficient memory for past and future experiences, making it more capable of long-horizon manipulation.

Model	Rearrange (%)	Reasoning (%)	Constraints (%)	Avg (%)
Gato	6.4	2.5	25.2	11.4
Flamingo	17.5	3.0	36.1	18.9
SuSIE	37.7	39.0	52.3	43.0
VIMA	43.1	38.2	67.2	49.5
CoTDiffusion	59.0	51.7	83.1	64.6
LiteVLP-m	**80.0**	**86.7**	**83.4**	**83.4**

Table 3. **The evaluations on three typical long-horizon tasks.** We refer to CoTDiffusion to select three representative long-horizon manipulation tasks in VIMA-Bench.

4.3. Efficiency Analysis

In this section, we evaluate the efficiency of our method in both training and inference. Our approach strikes a balance between performance and computational cost, achieving competitive accuracy with significantly reduced training time and faster inference speed.

Analysis of training time Based on the results present in Tab. 4, our method demonstrates a significant advantage on training efficiency. Specifically, we achieve an average success rate of 80.7% in just 6.1 hours of training, using 4 NVIDIA RTX 3090 GPUs. In comparison, VIMA, which is trained on 8 NVIDIA V100 GPUs, takes 24 hours and achieves an average success rate of 80.6%, while LLaRA-7B, trained on 4 NVIDIA RTX 3090 GPUs, requires 21 hours and achieves 74% on average. These results highlight the efficiency of our approach, which not only reduces training time significantly by 17.9 hours compared to VIMA and 14.9 hours compared to LLaRA-7B but also performs excellently even on less powerful GPU setups.

Model	Training time	Device (%)	Avg (%)
VIMA	24h	8*NVIDIA V100	80.6
LLaRA-7B	21h	4*NVIDIA RTX 3090	74.0
LiteVLP	**6.1h**	4*NVIDIA RTX 3090	**80.7**

Table 4. **Training time when achieving the highest success rate.** Comparison of training time over different models trained on GPUs with different performance levels.

Fast inference speed With a lightweight design, our model not only significantly reduces training time, but also accelerates inference speed, demonstrating a huge advantage on low latency. We fairly compare our LiteVLP-m with Mipha-3B and LLaRA on VIMA-Bench tasks, with the same NVIDIA RTX 3090. As shown in fig 1, our LiteVLP-m achieves a superior performance with 6.8 times lower inference latency than LLaRA. This result can be attributed to two main factors. First, our model contains only 1B parameters, which is smaller than the 7B parameters of LLaRA, greatly reducing computational overhead. Second, our method of multi-observation compression effectively reduces the number of image tokens, thereby shortening the input sequence length and thus accelerating inference speed. Fig. 7 shows four visualization examples of images processed by the MOC modeule.

4.4. Ablation Studies

In this section, we conduct several additional experiments to research the effect of multi-observation compression and the position of multiple image tokens.

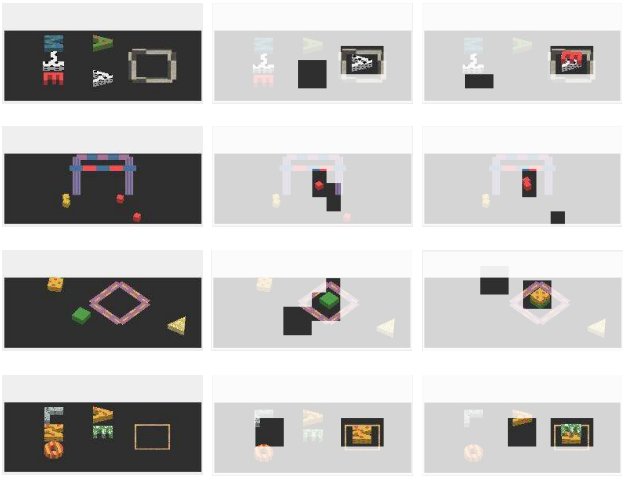

Figure 7. **Visualization of MOC results.** Here we present more visualization results of using MOC in different manipulations.

The effect of multi-observation compression We analyze the impact of multi-observation compression on training time and inference speed by comparing its use versus non-use. The results are shown in Tab 5. We observe that adopting multi-observation compression has minimal impact on task success rate but significantly reduces training time by 47.5% and increases inference speed by 34.1%. This can be explained by the change in input sequence length. In multi-image input data, image tokens constitute the majority of the sequence. The visualization results of the MOC module are shown in Fig. 7. The MOC module effectively reduces the number of these tokens, leading to a shorter sequence length, which in turn decreases training time and enhances inference speed.

Model	L1(%)	L2(%)	L3(%)	avg(%)
LiteVLP (w/ MOC)	84.2	**78.1**	75.4	79.2
LiteVLP (w/o MOC)	**84.6**	75.0	**79.3**	**79.6**

Table 5. **Ablation studies on the use of multi-observation compression.** This table illustrates the comparative analysis of task success rates with and without the activation of the MOC module.

Position of multiple image tokens Tab 6 illustrates the effect of the positions of multiple image tokens. Meanwhile, "Collection" means that all image tokens are gathered at the beginning of the prompt, while "Interleaved" indicates that image tokens are dispersed throughout the text. From the result, we can easily know that compared to "Collection", the "Interleaved" method results in a significantly lower average success rate of 71.2%, representing a 10.1% performance drop. The performance gap suggests that aggregating image tokens at the beginning provides a more

structured input representation, allowing the model to process visual information more effectively. Conversely, interleaving image tokens within the text may introduce discontinuities that hinder the model's ability to integrate multimodal information efficiently. These findings indicate that positioning image tokens at the beginning of the prompt is a more effective strategy to improve task success rates.

Position	L1(%)	L2(%)	L3(%)	avg(%)
Collection	**84.2**	**78.1**	**75.4**	**79.2**
Interleaved	72.7	69.6	71.2	71.2(10.1%↓)

Table 6. **Ablation studies on the position of multiple image tokens.** Position represents the positional relationship of image tokens inserted among text tokens. This table presents a comparison of task success rates under different insertion positional relationships.

5. Limitations and Future Work

5.1. Limitations

The primary limitations of LiteVLP include two aspects: its inability to operate effectively in 3D environments and its difficulty in handling contact-rich manipulation tasks. In the following, we discuss these challenges in detail.

Limitations in 3D robotic manipulations Since our model relies solely on 2D RGB images as input, it inherently lacks depth perception, which is crucial for understanding spatial structures in 3D environments. Additionally, because its capability is partially constrained by the coordinates of 2D object detection to identify target objects, extending its applicability to 3D scenes poses significant challenges. The absence of depth cues makes it difficult for the model to generalize beyond planar understanding, limiting its effectiveness in scenarios that require full spatial awareness.

Incapability of completing contact-rich manipulation tasks The output of our model includes two parts: the position and the rotation of the end effector. However, contact-rich manipulation tasks, such as assembling components, inserting objects, or handling deformable materials, require fine-grained force control and real-time adaptation to unpredictable physical interactions. Since our model relies primarily on visual and text input without explicit force or tactile feedback, it lacks the necessary sensory information to regulate interaction forces effectively. In addition, precise manipulation often requires compliance control, friction estimation, and dynamic adaptation, which are challenging to achieve with purely vision-language-conditioned policies.

5.2. Future work

In future work, we aim to enhance the model's performance in 3D environments and contact-rich manipulation. Integrating depth perception from depth sensors improves the spatial understanding, while fusing 2D and 3D data with self-supervised or contrastive learning may enhance reasoning and generalization. For contact-rich tasks, multi-modal feedback from force-torque and tactile sensors can enable adaptive interaction. Reinforcement learning in physics-based simulations may refine compliance control and dexterous manipulation. Additionally, hybrid policies combining vision-language reasoning with force-aware control could bridge perception and physical interaction for more robust real-world manipulation.

6. Conclusion

In this work, we introduce LiteVLP, a lightweight vision-language policy designed to address the challenges of inference efficiency, domain adaptation, and memory-based planning in robotic manipulation tasks. By leveraging a pre-trained 1B-parameter VLM and fine-tuning it on conversation-style robotic datasets, LiteVLP achieves state-of-the-art performance on VIMA-Bench while maintaining superior inference speed and data efficiency.

Our proposed multi-observation compression module significantly reduces the number of image tokens, leading to 47.5% lower training time and a 34.1% improvement in inference speed without compromising task success rates. Moreover, LiteVLP integrates historical memory and future goal images, enabling more effective long-horizon manipulation. Extensive evaluations demonstrate that LiteVLP outperforms the best-performing baseline by 18.8% while operating under significantly lower computational constraints.

These results highlight LiteVLP as a promising step toward fast, memory-based and data-efficient vision-language policy generation models for robotic manipulation, paving the way for the integration of compact multi-modal models into real-world robotic applications.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 1

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 23716–23736, 2022. 2

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vi-

sion transformer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6836–6846, 2021. 2

[4] Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, et al. Openflamingo: An open-source framework for training large autoregressive vision-language models. *arXiv preprint arXiv:2308.01390*, 2023. 2

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning (ICML)*, number 3, page 4, 2021. 2

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 1, 2, 3

[7] Sean L Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J Pappas. Probabilistic data association for semantic slam. In *International Conference on Robotics and Automation (ICRA)*, pages 1722–1729, 2017. 2

[8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 2

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

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

[11] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 1

[12] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4247–4258, 2020. 2

[13] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 1

[14] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, page 220101, 2024. 2

[15] An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024. 1

[16] Rohan Choudhury, Guanglei Zhu, Sihan Liu, Koichiro Niinuma, Kris Kitani, and László Jeni. Don't look twice: Faster video transformers with run-length tokenization. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 28127–28149, 2025. 2

[17] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. In *International Conference on Machine Learning (ICML)*, pages 8469–8488, 2023. 1

[18] Jiafei Duan, Wentao Yuan, Wilbert Pumacay, Yi Ru Wang, Kiana Ehsani, Dieter Fox, and Ranjay Krishna. Manipulate-anything: Automating real-world robots using vision-language models. *arXiv preprint arXiv:2406.18915*, 2024. 1

[19] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. *arXiv preprint arXiv:2205.04382*, 2022. 2

[20] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6824–6835, 2021. 2

[21] Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. Sam2act: Integrating visual foundation model with a memory architecture for robotic manipulation. *arXiv preprint arXiv:2501.18564*, 2025. 2

[22] Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Applications, challenges, and the future. *International Journal of Robotics Research (IJRR)*, page 02783649241281508, 2023. 1

[23] Muzhi Han, Yifeng Zhu, Song-Chun Zhu, Ying Nian Wu, and Yuke Zhu. INTERPRET: interactive predicate learning from language feedback for generalizable task planning. In *Robotics: Science and Systems (RSS)*, 2024. 1

[24] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 20482–20494, 2023. 3

[25] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023. 2

[26] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023. 1

[27] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: Robot manipulation with multimodal prompts. 2023. 2, 4

[28] Sascha Jockel, Martin Weser, Daniel Westhoff, and Jianwei Zhang. Towards an episodic memory for cognitive robots. pages 68–74, 2008. 2

[29] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1, 2

[30] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, pages 19730–19742, 2023. 2

[31] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, 2023. 2

[32] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023. 2

[33] Xiang Li, Cristina Mata, Jongwoo Park, Kumara Kahatapitiya, Yoo Sung Jang, Jinghuan Shang, Kanchana Ranasinghe, Ryan Burgert, Mu Cai, Yong Jae Lee, et al. Llara: Supercharging robot learning data for vision-language policy. *arXiv preprint arXiv:2406.20095*, 2024. 2, 4, 5

[34] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4804–4814, 2022. 2

[35] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *International Conference on Robotics and Automation (ICRA)*, pages 9493–9500, 2023. 1

[36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2

[37] Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022. 3

[38] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024. 1, 2

[39] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 909–918, 2019. 2

[40] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6813–6823, 2021. 2

[41] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 2

[42] Fei Ni, Jianye Hao, Shiguang Wu, Longxin Kou, Jiashun Liu, Yan Zheng, Bin Wang, and Yuzheng Zhuang. Generate subgoal images before act: Unlocking the chain-of-thought reasoning in diffusion model for robot manipulation with multimodal prompts. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13991–14000, 2024. 6

[43] Dantong Niu, Yuvan Sharma, Giscard Biamby, Jerome Quenum, Yutong Bai, Baifeng Shi, Trevor Darrell, and Roei Herzig. Llarva: Vision-action instruction tuning enhances robot learning. *arXiv preprint arXiv:2406.11815*, 2024. 1

[44] Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 1

[45] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2

[46] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. 2

[47] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025. 1

[48] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2

[49] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In *Conference on Robot Learning (CoRL)*, pages 894–906, 2022. 2

[50] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *International Conference on Computer Vision (ICCV)*, pages 11888–11898, 2023. 1

[51] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey

Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 2

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

[53] David Vernon. Cognitive architectures. 2022. 2

[54] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2

[55] Qianxu Wang, Haotong Zhang, Congyue Deng, Yang You, Hao Dong, Yixin Zhu, and Leonidas Guibas. Sparsedff: Sparse-view feature distillation for one-shot dexterous manipulation. *arXiv preprint arXiv:2310.16838*, 2023. 2

[56] Junjie Wen, Minjie Zhu, Yichen Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Chengmeng Li, Xiaoyu Liu, Yaxin Peng, Chaomin Shen, et al. Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression. *arXiv preprint arXiv:2412.03293*, 2024. 1, 2

[57] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2409.12514*, 2024. 1

[58] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024. 1, 2

[59] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022. 1

[60] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11975–11986, 2023. 2

[61] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 1, 3

[62] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. Tracevla: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. *arXiv preprint arXiv:2412.10345*, 2024. 1, 2

[63] Zhongyi Zhou, Yichen Zhu, Minjie Zhu, Junjie Wen, Ning Liu, Zhiyuan Xu, Weibin Meng, Ran Cheng, Yaxin Peng, Chaomin Shen, and Feifei Feng. Chatvla: Unified multimodal understanding and robot control with vision-language-action model. *arXiv preprint arXiv:2502.14420*, 2025. 1

[64] Minjie Zhu, Yichen Zhu, Xin Liu, Ning Liu, Zhiyuan Xu, Chaomin Shen, Yaxin Peng, Zhicai Ou, Feifei Feng, and Jian Tang. Mipha: A comprehensive overhaul of multimodal assistant with small language models. *arXiv preprint arXiv:2403.06199*, 2024. 2