ControlVLA: Few-shot Object-centric Adaptation for Pre-trained Vision-Language-Action Models

Puhao Li[1,2], **Yingying Wu**[1,2], **Ziheng Xi**[1], **Wanlin Li**[2], **Yuzhe Huang**[2], **Zhiyuan Zhang**[2], **Yinghan Chen**[2,3], **Jianan Wang**[4], **Song-Chun Zhu**[1,2,3], **Tengyu Liu**[2,†], **Siyuan Huang**[2,†]

[1]Tsinghua University [2]State Key Lab of General Artificial Intelligence, BIGAI
[3]Peking University [4]Astribot [†]Corresponding author

https://controlvla.github.io

Abstract: Learning real-world robotic manipulation is challenging, particularly when limited demonstrations are available. Existing methods for few-shot manipulation often rely on simulation-augmented data or pre-built modules like grasping and pose estimation, which struggle with sim-to-real gaps and lack extensibility. While large-scale imitation pre-training shows promise, adapting these general-purpose policies to specific tasks in data-scarce settings remains unexplored. To achieve this, we propose ControlVLA, a novel framework that bridges pre-trained VLA models with object-centric representations via a ControlNet-style architecture for efficient fine-tuning. Specifically, to introduce object-centric conditions without overwriting prior knowledge, ControlVLA zero-initializes a set of projection layers, allowing them to gradually adapt the pre-trained manipulation policies. In real-world experiments across 6 diverse tasks, including pouring cubes and folding clothes, our method achieves a 76.7% success rate while requiring only 10-20 demonstrations — a significant improvement over traditional approaches that require more than 100 demonstrations to achieve comparable success. Additional experiments highlight ControlVLA's extensibility to long-horizon tasks and robustness to unseen objects and backgrounds.

Keywords: Robotic manipulation, Imitation learning, Few-shot learning

1 Introduction

Robotic manipulation in the real world remains a fundamental challenge, particularly when learning skills from limited demonstrations. While recent advances in robotic manipulation [1–16] have shown promise, current methods still demand extensive training data and struggle to efficiently adapt to new tasks and environments with few demonstrations. This limitation significantly hinders the deployment of robots in diverse real-world scenarios, where large amounts of task-specific training data are often impractical or prohibitively expensive.

To tackle this, previous works [10, 17–19] augment expert demonstrations in simulation to enhance policy learning. However, these approaches typically assume *a priori* knowledge of object and environment CAD models, as well as precise 3D pose estimations—requirements often impractical in real-world scenarios. Currently, imitation learning has achieved impressive success in manipulation [1, 3, 8, 11, 20–23], primarily due to its scalability and capability to acquire skills across a wide range of scenarios without relying on external *a priori* knowledge.

In particular, **pre-trained general-purpose Vision-Language-Action models (VLA model)** [3, 11, 22–25] has emerged as a promising approach for enabling generalizable robot behaviors across various tasks and environments. However, fine-tuning these VLA model for downstream tasks remains data-intensive, as substantial amounts of task- and environment-specific data are still required to adapt to the visual and action domains of the target task [3, 22, 23, 26]. On a separate front,

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

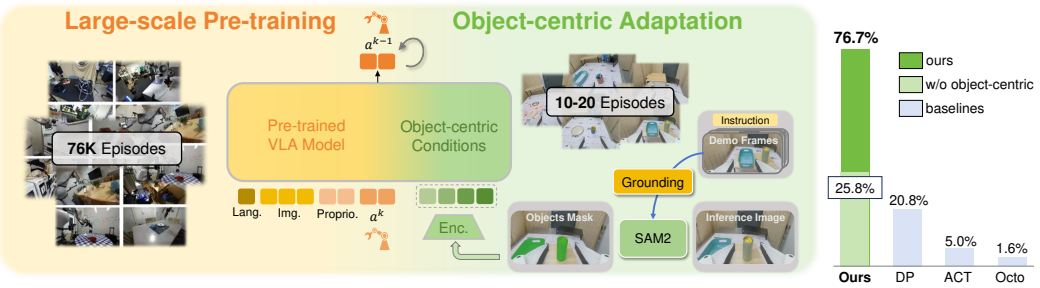

Fig. 1: **ControlVLA bridges pre-trained manipulation policies with object-centric representations via ControlNet-style efficient fine-tuning.** ControlVLA requires only 10–20 demonstrations to achieve 76.7% task success rate, significantly surpassing baseline's 20.8% success rate.

object-centric representations has shown potential in improving data efficiency for learning expert policies [4, 7, 27–29]. By focusing on relevant object properties (*e.g.*, shape, size, and position), object-centric representations reduce the complexity of the input observation space. This approach enhances policy robustness to changes in object pose and instance, while also making the policies less susceptible to real-world noise compared to pixel-level features. Despite this, existing methods still require hundreds of demonstrations to learn a task [4, 27, 29], mainly due to the lack of an action prior from VLA model in data-scarce scenarios.

To this end, we introduce **ControlVLA**, a novel learning framework that combines pre-trained VLA model with object-centric representations for efficient few-shot learning. By integrating object-centric representations into a pre-trained VLA model, our approach leverages both the rich prior knowledge from large-scale pre-training and the data efficiency of object-centric learning. Inspired by Zhang *et al*. [30], ControlVLA introduces **additional cross-attention layers with zero-initialized key-value (KV) projection weights**, allowing expert policies to acquire task-specific skills while progressively integrating object-centric representations. This design ensures that the policy focuses on task-relevant concepts without compromising the generalization or action quality of the pre-trained VLA model. The zero-initialization of additional KV projections stabilizes fine-tuning by mitigating the introduction of harmful noise, thereby enabling a seamless integration of task-specific object-centric representations with general-purpose VLA model pre-training. As a result, ControlVLA significantly reduces data requirements for task-specific adaptation, enhancing the efficiency of robotic manipulation deployment in the real world.

We demonstrate the efficacy of ControlVLA across **8 diverse real-world tasks**, achieving robust performance **with only 10-20 demonstrations**. The evaluation tasks span diverse manipulation challenges: pick-and-place tasks with rigid, soft, and precision-critical objects, as well as complex manipulations including articulated object operation, object pouring, and deformable cloth folding. Empirically, ControlVLA attains an impressive **76.7%** success rate across all tasks with limited demonstrations, significantly surpassing baseline methods that achieve a mere 20.8% success rate. Additionally, we demonstrate the extensibility of ControlVLA on **long-horizon tasks** and its robustness on **unseen objects and backgrounds**. Ablation studies confirm the necessity of three key components: (1) VLA model pre-training for skill priors, (2) object-centric representation for efficient task grounding and learning, and (3) ControlNet-style conditioning for stable adaptation. In summary, ControlVLA bridges the gap between **large-scale VLA model pre-training** and **efficient object-centric adaptation**, enabling robots to acquire complex skills from minimal demonstrations.

2 Related Works

2.1 Few-shot Learning for Manipulation

Reducing reliance on costly demonstrations while ensuring robust performance remains a key challenge in robotic manipulation. Early approaches synthesized training data through simulation-based augmentation [17, 18] or combined imitation with reinforcement learning for robustness [19], yet

these methods depend heavily on accurate object poses and CAD models, limiting real-world applicability. To overcome sim-to-real gaps, especially in contact-rich or deformable tasks, recent methods turn to *human video priors*, using representations like R3M [21], VIP [9] or Ag2Manip [2] to guide policy learning. However, these largely rely on 2D cues and struggle with precise spatial reasoning. Concurrently, few-shot learning aims to generalize from minimal demonstrations, with DenseMatcher [31] and SPOT [7] addressing transferable skill learning via 3D correspondences and object-centric planning. Yet, most approaches still require handcrafted grasping routines and accurate pose pipelines, and few leverage *pre-trained VLA model* as priors. Our work addresses these gaps by introducing ControlNet-style conditioning for object-centric policy modulation, enabling efficient few-shot adaptation in unstructured settings.

2.2 Object-centric Representation Learning

Object-centric representation decomposes complex scenes into manipulable entities, enabling more efficient reasoning for robot learning. Early methods represent objects by poses [32–34] or by bounding boxes [35, 36], but their dependence on known categories or instance labels limits generalization to novel objects and dynamic environments. Unsupervised discovery techniques segment visual inputs into object-like regions without requiring labels [37, 38], yet they struggle in cluttered or occluded scenes and often produce inconsistent object identities [39, 40]. Furthermore, most existing approaches fail to leverage large-scale pre-trained models effectively, aligning instead to category-specific or pose-based features [41–44], which necessitates extensive task-specific tuning and undermines the benefits of transfer learning. To address these challenges, we introduce a unified framework that integrates object-centric decomposition with large-scale, data-driven representations, thereby improving adaptability and transferability in real-world manipulation tasks.

2.3 ControlNet-style Fine-tuning

ControlNet [30] enhances large-scale pre-trained Stable Diffusion [45] by efficiently incorporating additional conditional inputs, such as sketch, normal map, depth map or human pose, through zero-initialized convolution layers. These layers start with zero weights and bias, ensuring no initial impact on outputs while progressively learning to integrate new conditions. This methodology has been extensively applied in various domains, such as controllable image generation [30, 46, 47], video generation [48–50], and human motion generation [51, 52]. Despite its widespread adoption in these areas, the application of ControlNet in the context of robotic manipulation has not yet been investigated. Our study represents the first effort to adapt ControlNet-style fine-tuning to this field, unifying large-scale VLA model pre-training with fine-tuning of object-centric representations to enable few-shot robotic manipulation learning.

3 Method

Given a pre-trained general-purpose policy $\boldsymbol{\pi_g} : \boldsymbol{\mathcal{O}} \rightarrow \boldsymbol{\mathcal{A}}$, where $\boldsymbol{\mathcal{O}}$ and $\boldsymbol{\mathcal{A}}$ denote observation and action spaces, our goal is to efficiently adapt it into a task-specific expert policy $\boldsymbol{\pi_e}$ using limited expert demonstrations. To this end, we propose ControlVLA, which employs a ControlNet-style fine-tuning that integrates object-centric representations with a pre-trained VLA model (see Fig. 2). We first pre-train the general-purpose policy on a large-scale, multi-task manipulation dataset (Sec. 3.1), then extract object-centric features to focus learning on task-relevant elements (Sec. 3.2), and finally fine-tune the policy by gradually incorporating these features (Sec. 3.3). Details are in the appendix.

3.1 VLA Model Pre-training

We begin by pre-training a general-purpose policy $\boldsymbol{\pi_g}$, using the public large-scale manipulation datasets $\boldsymbol{\mathcal{D}_g} = \left\{ (\boldsymbol{o_t}, \boldsymbol{a_t})_{t=1}^{T_i} \right\}_{i=1}^{N_g}$ across a diverse range of tasks and scenes, where N_g represents the total number of episodes. Formally, we use $\boldsymbol{\pi_g} : \boldsymbol{\mathcal{O}} \rightarrow \boldsymbol{\mathcal{A}}$ to model the conditional action distribution $p\left(\boldsymbol{A_t} \mid \boldsymbol{O_t}\right)$, where $\boldsymbol{A_t}$ represents the future action sequence and $\boldsymbol{O_t}$ denotes the history

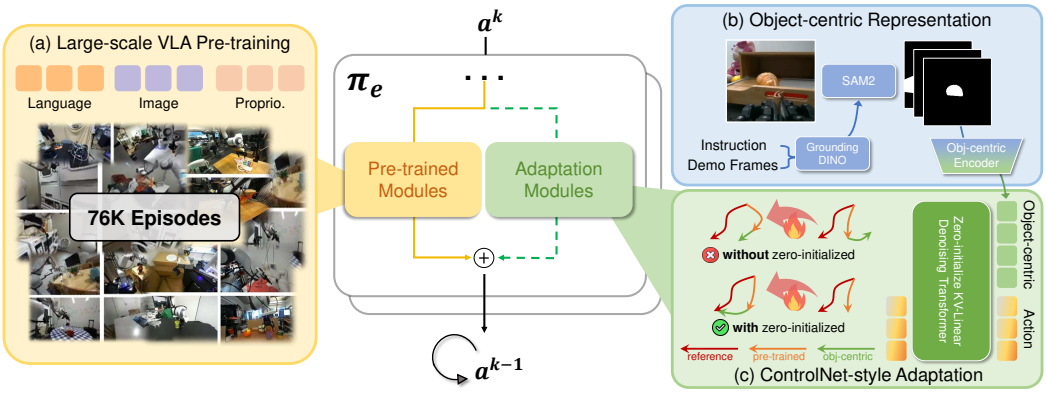

Fig. 2: **Overview of ControlVLA.** ControlVLA leverages a ControlNet-style fine-tuning strategy to integrate object-centric representations with the pre-trained VLA model. The zero-initialized weights and biases preserve the rich prior knowledge of the pre-trained policy while progressively grounding it in object-centric representation.

of the observations. The observation at time t consists of a single-view RGB image $\boldsymbol{I_t}$, a language instruction ℓ_t, and the robot's proprioceptive state $\boldsymbol{q_t}$, such that $\boldsymbol{o_t} = [\boldsymbol{I_t}, \ell_t, \boldsymbol{q_t}]$. The image $\boldsymbol{I_t}$ and language instruction ℓ_t are tokenized via pre-trained encoders and projected into a shared embedding space through a linear projection layer, while the proprioceptive state $\boldsymbol{q_t}$ is similarly embedded using Multi-layer Perceptrons (MLPs).

The $\boldsymbol{\pi_g}$ model adopts a diffusion transformer architecture to capture the multimodal conditional action distribution. During training, the action sequence is supervised using a conditional denoising loss. At inference, actions are sampled by iteratively denoising started from pure Gaussian noise $\boldsymbol{A_t^K} \sim \mathcal{N}(0, I)$ into the desired action $\boldsymbol{A_t^0} \sim q_\theta\left(\boldsymbol{A_t^0} \mid \boldsymbol{O_t}\right)$, with the process accelerated using Denoising Diffusion Implicit Model (DDIM) [53] for real-time control.

3.2 Object-centric Representations

This section outlines the process of building object-centric representations $\boldsymbol{Z} \in \boldsymbol{\mathcal{Z}}$ as additional action conditions, which enable task-specific expert policy $\boldsymbol{\pi_e}$ to explicitly identify the key concepts of the task and efficiently learn from few-shot demonstrations. The process involves two key steps: (i) segment and track task-relevant objects, and (ii) learn to extract object-centric representations.

Segment and Track task-relevant objects. To build object representations that consistently attend to task-relevant objects, it is essential to inform the model about their locations and local geometry in the RGB image observation $\boldsymbol{I}$. Our goal is to automatically access fine-grained instance masks $\left\{\boldsymbol{M^i}\right\}_{i=1}^{N_{\text{obj}}}$ corresponding to task-relevant objects $\left\{\mathbf{obj^i}\right\}_{i=1}^{N_{\text{obj}}}$, where N_{obj} denotes the number of objects. To achieve this, we extract N_{img} frames from demonstration and language instruction as prompts, and leverage GroundingDINO [54] and SAM2 [55] to consistently segment and track task-relevant objects, both in the training data and during real-time inference.

Learn to extract object-centric representations. We aim to learn a $\boldsymbol{f_\varphi}$ to extract the object-centric representations $\boldsymbol{z^i}$ from $i-\text{th}$ object mask $\boldsymbol{M^i}$ as additional conditions for the expert policy $\boldsymbol{\pi_e}$. Inspired by [28], we follow the idea of encoding *where* and *what* relevant objects are, by designing *positional feature* and *geometrical feature* for each object. For positional feature $\boldsymbol{z^i_{\text{pos}}}$, we encode the mean coordinates of the object mask on images using sinusoidal positional encoding [56]. For geometrical feature $\boldsymbol{z^i_{\text{geo}}}$, we obtain a spatial feature map with a Convolutional Neural Network (CNN) [57] that runs on the mask of each task-relevant object. Similar to the approach of Zhu *et al*. [4], we train the spatial network from scratch rather than using a pre-trained model, as we require actionable visual features that are specifically informative for continuous control tasks. Finally, we concatenate the positional and geometrical features to form the object-centric representation $\boldsymbol{z^i} = \left[\boldsymbol{z^i_{\text{pos}}}, \boldsymbol{z^i_{\text{geo}}}\right]$ and $\boldsymbol{Z} = \left\{\boldsymbol{z^i}\right\}_{i=1}^{N_{\text{obj}}} \in \boldsymbol{\mathcal{Z}}$.

3.3 ControlNet-style Fine-tuning

Given a small set of task-specific dataset $\boldsymbol{\mathcal{D}_e} = \left\{(\boldsymbol{o_t}, \boldsymbol{z_t}, \boldsymbol{a_t})_{t=1}^{T_i}\right\}_{i=1}^{N_e}$, where N_e represents the number of demonstrations, we aim to efficiently fine-tune an expert policy $\boldsymbol{\pi_e} : \boldsymbol{\mathcal{O}} \times \boldsymbol{\mathcal{Z}} \rightarrow \boldsymbol{\mathcal{A}}$ from $\boldsymbol{\pi_g} : \boldsymbol{\mathcal{O}} \rightarrow \boldsymbol{\mathcal{A}}$ with the object-centric representations.

In our context, the pre-trained policy is a transformer-based architecture that utilizes cross-attention blocks to model actions conditioned on observations. Specifically, the cross-attention mechanism computes the relationship between actions $\boldsymbol{A}$ and observations $\boldsymbol{O}$ as:

$$\text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right)\boldsymbol{V}, \tag{1}$$

where $\boldsymbol{Q} = \mathbf{W_a}\boldsymbol{A} + \mathbf{B_a}$ represents the query projection, and $\boldsymbol{K}, \boldsymbol{V} = \mathbf{W_o}\boldsymbol{O} + \mathbf{B_o}$ represent the key and value projections, respectively. To incorporate the object-centric representation $\boldsymbol{Z} \in \boldsymbol{\mathcal{Z}}$, we extend the cross-attention mechanism by introducing a dual-attention structure.

$$\text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d}}\right)\boldsymbol{V} + \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}_{\boldsymbol{z}}^T}{\sqrt{d}}\right)\boldsymbol{V_z}, \tag{2}$$

where $\boldsymbol{K_z}, \boldsymbol{V_z} = \mathbf{W_z}\boldsymbol{Z} + \mathbf{B_z}$ are the key and value projections for the object-centric observations.

Inspired by Zhang *et al*. [30], we *zero-initialize* the additional KV-projection layers to ensure the expert policy $\boldsymbol{\pi_e}$ model behaves similarly to the pre-trained general-purpose policy $\boldsymbol{\pi_g}$ during the early stage of fine-tuning, preserving the model's prior knowledge. Since the weight $\mathbf{W_z}$ and bias $\mathbf{B_z}$ are initialized to $\mathbf{0}$, the key and value projections for $\boldsymbol{Z}$ are zero:

$$\mathbf{K_z} = \mathbf{W_z}\boldsymbol{Z} + \mathbf{B_z} = \mathbf{0}, \quad \mathbf{V_z} = \mathbf{0}. \tag{3}$$

Thus, the dual-attention in Eq. (2) reduces to the original cross-attention in Eq. (1), preserving the pre-trained policy's behavior at the first fine-tuning step. We provide further explanation in Appx. B.

4 Experiments

To evaluate the efficiency of ControlVLA, we conduct 8 various real-world tasks using only 10-20 demonstrations. Empirically, our findings indicate that our method consistently and significantly improves success rates across all short-horizon tasks, achieving an overall success rate of **76.7%**, which markedly surpasses the baseline of 20.8%. For long-horizon tasks, we evaluate ControlVLA on two challenging scenarios, where it outperforms the state-of-the-art method with approximately 3x higher success rates. We further assess policy performance on data scaling. Our results show that our method rapidly converges to a high success rate with as few as 20 demonstrations, while baselines require more than 100 demonstrations to achieve comparable performance. Additionally, we also demonstrate the robustness of our method over unseen objects and backgrounds.

4.1 Experimental Setup

Tasks. We develop a suite of various real-world tasks to evaluate the efficacy of our proposed method. These tasks are designed to cover a wide range of manipulation challenges, including pick-and-place of various types of objects like rigid `RearrangeCup`, soft `OrganizeToy`, and precise `OrganizeScissors`, as well as articulated `OpenCabinet`, deformable `FoldClothes`, and fluid-like `PourCubes` manipulations. To further evaluate the performance in long-horizon scenarios, we introduce 2 additional tasks: `OrganizeMultiObjs` and `ReplaceObjInDrawer`, which entail sequential decision-making and handling of temporally extended goals. A detailed illustration of each task is given in Tab. 1 while visualization is provided in Fig. 3.

Baselines and Ablations. We compare our method against Octo [22], π_0 [26], VIOLA [4], ACT [58], and Diffusion Policy [20]. Octo and π_0 are pre-trained foundation VLA model, employing a diffusion-based model to decode the action tokens. VIOLA is a widely recognized

Tab. 1: **Illustrations of Evaluation Tasks.** We develop a suite of real-world tasks for evaluation, including rigid, soft, articulated, deformable, and fluid-like objects. Two tasks are long-horizon, involving temporally extended behaviors with multiple sub-goals and sustained policy control.

TASK NAME	TASK TYPE	#DEMOS	LANGUAGE DESCRIPTION
`RearrangeCup`	Rigid Pick&Place	14	Rearrange the cup and set it on the light plate.
`OrganizeToy`	Soft Pick&Place	20	Pick up the green toy and place it in the blue bowl.
`OrganizeScissors`	Precise Pick&Place	15	Pick up the scissors from the pen holder into the blue basket.
`OpenCabinet`	Articulated Manipulation	11	Open the cabinet with the black handle.
`FoldClothes`	Deformable Manipulation	16	Fold up the sleeves of the pink clothing on the table.
`PourCubes`	Pouring Behavior	19	Pour the blocks from the green cup into the blue box.
`OrganizeMultiObjs`	Long Horizon	25	Organize eggplant, strawberry, and carrot into the woven basket.
`ReplaceObjInDrawer`	Long Horizon	25	Open the drawer, take out the bread, and put the carrot in.

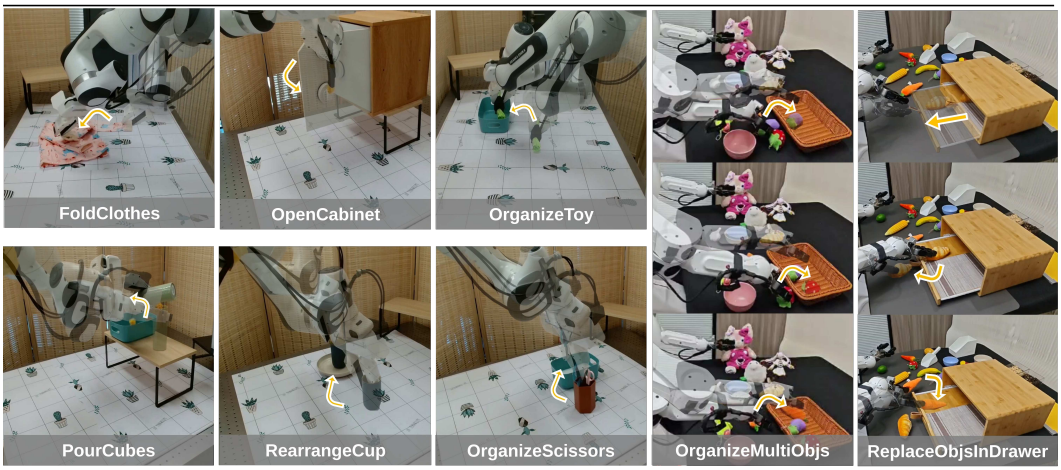

Fig. 3: **Task Visualization.** The initial and target states are shown as transparent and solid layers, respectively. The yellow arrow highlights the desired transition.

2D object-centric transformer-based policy learning framework that utilizes Region Proposal Network (RPN) [59] to extract object-centric representations. ACT and Diffusion Policy are among the most extensively studied and widely applied imitation visuomotor policies. ACT models the actions using a Variational Autoencoder (VAE), while Diffusion Policy leverages Denoising Diffusion Probabilistic Model (DDPM) to capture more multimodal action distributions.

In our ablation study, we systematically remove individual components from our method to investigate their independent contributions. We ablate the pre-training phase by training an object-centric Diffusion Policy from scratch, denoted as "w/o pretrain". We eliminate the object-centric representations by directly fine-tuning the pre-trained model, denoted as "w/o object-centric". To assess the significance of ControlNet-style fine-tuning in integrating object-centric representations into pre-trained policies, we omit the zero-initialization of the projection layers for additional object-centric conditions, denoted as "w/o zero-init".

We provide **Data Collection** and **Evaluation Setup and Protocol** details in the appendix.

4.2 Comparative and Ablation Results

For full comparison and ablation studies, we evaluate each model over 20 trials across 6 short-horizon tasks. All policies are trained or fine-tuned from a shared, limited demonstration dataset for each task. As illustrated in Fig. 4, the task success rates are presented within and across all evaluation tasks, providing a comprehensive overview of our findings.

Our method, ControlVLA, achieves an impressive **overall task success rate of 76.7%**, significantly outperforming the baselines. The strongest baseline, Diffusion Policy, attains only a 20.8% success rate and struggles to precisely manipulate target objects or learn complex behaviors such as pouring. Octo and ACT only achieve $2/20$ and $6/20$ success on the `OpenCabinet` task, with no successes in other tasks, resulting in the overall success rate of just 1.6% and 5.0%. Despite Octo's advantage from large-scale pre-training, its regression-based backbone fails to model action distributions with

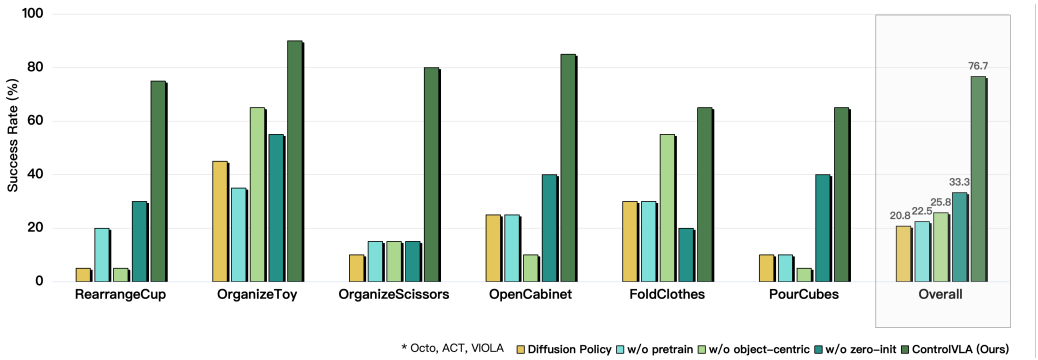

Fig. 4: **Main Comparison and Ablation Study.** All policies are trained or fine-tuned from a shared, limited demonstration dataset for each task. *Octo, ACT, and VIOLA are omitted due to very low success rates, with overall success rates of 1.6%, 5.0%, and 0.0%, respectively.

Tab. 2: **Performance on Long-horizon Tasks.**

	`OrganizeMultiObjects`				`ReplaceObjectsInDrawer`			
	Stage-1	Stage-2	Stage-3	Overall	Stage-1	Stage-2	Stage-3	Overall
DP [20]	14/30	7/14	3/7	10.0%	11/30	5/11	2/5	6.7%
π_0 [26]	18/30	10/18	7/10	23.3%	19/30	9/19	5/9	16.7%
Ours	26/30	23/26	17/23	**56.7%**	25/30	22/25	19/20	**63.3%**

Tab. 3: **Performance on Generalization.**

	obj-1	obj-2	obj-3	bg
DP	16.7%	20.0%	13.3%	6.7%
π_0	26.7%	23.3%	20.0%	16.7%
Ours	**76.7%**	**63.3%**	**70.0%**	**60.0%**

multiple distinct modes. While ACT leverages a VAE to represent action diversity, it is hindered by posterior collapse, especially in the low-data regime. VIOLA fails primarily due to the lack of action pre-training and its object-centric representation extraction, which relies on a large number of task demonstrations. In contrast, ControlVLA demonstrates robust and efficient learning across various real-world tasks even when only limited demonstrations are available.

Ablation studies reveal the critical role of key components of ControlVLA. Removing the pre-training phase and directly incorporating object-centric representations ("w/o pretrain") results in severe jitter problems when policies deploy to a real robot, causes severe execution jitter and significant drops in performance across tasks (details are provided in the appendix). We hypothesize that the object-centric features may provide deceptive low-loss pathways during the early training stage, incentivizing the policy to bypass learning from the more stable visual features. Eliminating object-centric representations during fine-tuning ("w/o object-centric") provides only a marginal improvement over training Diffusion Policy from scratch, highlighting the importance of object-centric representations. Finally, removing zero-initialization for additional object-centric conditions ("w/o zero-init") causes a drastic drop in success rate, emphasizing the role of proper initialization in stabilizing training and improving task performance. Notably, our complete methodology degenerates to the Diffusion Policy baseline when stripping all three components (pre-training, object-centric representations, and zero-initialization).

4.3 Performance on Long-horizon Tasks

We evaluate the ability of ControlVLA to perform long-horizon tasks, each trained with only 25 demonstrations. These tasks require the robot to execute multiple sub-goals in sequence, making them particularly challenging under limited supervision. Specifically, `OrganizeMultiObjects` involves placing three objects—eggplant, strawberry, and carrot—into a woven basket, corresponding to three sequential stages. `ReplaceObjectsInDrawer` requires opening a drawer, taking out a bread item, and placing a carrot inside, similarly decomposable into three dependent stages.

As shown in Tab. 2, ControlVLA significantly outperforms baselines under this low-data regime, achieving an average 60.0% success rate on two long-horizon tasks. The stage-wise breakdown further demonstrates ControlVLA's consistent performance throughout long action sequences. These results highlight the robustness of ControlVLA in long-horizon settings, even with minimal demonstrations, and its ability to reduce compounding errors during sequential execution.

Tab. 4: Task success rates of π_0 and ControlVLA@π_0.

	Organize Toy	Organize Scissors	Open Cabinet	Fold Clothes	Overall
π_0 [26]	55.0%	15.0%	45.0%	40.0%	38.6%
ControlVLA@π_0	85.0%	80.0%	85.0%	75.0%	**81.3%**

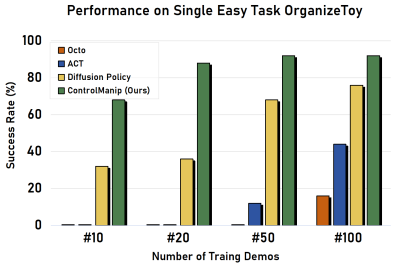

Fig. 5: Effect of Data Scaling on Performance in the `OrganizeToy` Task.

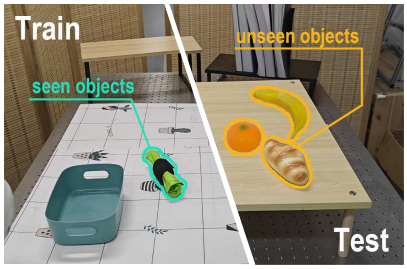

Fig. 6: Generalization over object and background appearance changes.

4.4 Performance on Data Scaling

We evaluate ControlVLA's data efficiency through controlled scaling experiments on the `OrganizeToy` task, benchmarking against established baseline methods. Each approach is tested across demonstration set sizes of 10, 20, 50, and 100 episodes with 25 trials per condition, with results shown in Fig. 5. While all methods improve as the amount of training data increases, ControlVLA achieves a high success rate of 80% with only 20 demonstrations — a level unattained by baselines even at 100 demonstrations. This highlights the efficiency of ControlVLA in learning from limited data.

4.5 Generalization on Object and Background

We evaluate the generalization and robustness of ControlVLA on the `OrganizeToy` task by testing it with unseen objects and backgrounds. Trained on 20 demonstrations using a green toy and uniform background, ControlVLA achieves a 70.0% average success rate across three novel objects (bread, banana, orange) and a 60.0% success rate on a previously unseen background—requiring only an object prompt for mask extraction (Tab. 3). Although these results fall short of the 90% in-domain success rate, they highlight ControlVLA's capacity to adapt to dynamic and diverse environments without additional training.

4.6 Enhancing Performance of π_0 with ControlVLA

We further adapt and evaluate our ControlVLA method on a more general pre-trained VLA model, π_0 [26], referred to as ControlVLA@π_0. Implementation details are in Appx. D.3. We conduct the comparison studies over 20 trials across 4 sub-tasks, each fine-tuned with limited demonstrations, consistent with the setup used in the main paper.

As shown in Tab. 4, our adaptation method ControlVLA@π_0 consistently outperforms the fine-tuned π_0, demonstrating that ControlVLA can serve as a plug-in module to enhance performance across a broader range of pre-trained VLA model models. The improvement is most pronounced on the `OrganizeScissors` task, highlighting ControlVLA's ability to provide more precise guidance for fine-grained manipulation in data-scarce scenarios.

5 Conclusion

This work introduces ControlVLA, a framework that bridges VLA model pre-training with object-centric representations to enable efficient few-shot adaptation for robotic manipulation. By integrating a ControlNet-style fine-tuning, ControlVLA injects task-specific object-centric conditions into a pre-trained VLA model through zero-initialized key-value projection layers. This design preserves the rich prior knowledge of the base policy while progressively grounding it in structured object properties, achieving stable fine-tuning with minimal demonstrations. Across 8 diverse real-world tasks—ranging from rigid object pick-and-place to deformable cloth folding and long-horizon tasks — ControlVLA achieves a 76.7% success rate with only 10–20 demonstrations, outperforming the baselines. By reducing demonstration requirements to practical levels, ControlVLA lowers barriers to deploying robots in diverse scenarios.

6 Limitations

Our evaluation tasks are currently constrained to single-arm manipulation, with experiments conducted primarily in controlled indoor environments. While these settings provide a reliable testbed, they fall short of capturing the full range of challenges faced in real-world deployment. To improve the generalization of ControlVLA, it is essential to explore bi-manual and in-the-wild manipulation tasks. Nevertheless, ControlVLA is a general framework, and future work could explore larger pre-trained bi-manual foundation VLA model, along with more efficient and task-relevant modalities and representations to tackle these challenges.

Acknowledgments

We thank Yuyang Li (BIGAI, PKU) for his technical support and contributed discussion, Yuwei Guo (CUHK) for his discussion, and Ziyuan Jiao (BIGAI) for his assistance in setting up the real-world environment.

References

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

[2] P. Li, T. Liu, Y. Li, M. Han, H. Geng, S. Wang, Y. Zhu, S.-C. Zhu, and S. Huang. Ag2manip: Learning novel manipulation skills with agent-agnostic visual and action representations. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 573–580. IEEE, 2024.

[3] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[4] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. In *Conference on Robot Learning*, pages 1199–1210. PMLR, 2023.

[5] T. Chen, Y. Mu, Z. Liang, Z. Chen, S. Peng, Q. Chen, M. Xu, R. Hu, H. Zhang, X. Li, et al. G3flow: Generative 3d semantic flow for pose-aware and generalizable object manipulation. *arXiv preprint arXiv:2411.18369*, 2024.

[6] Y. Wang, G. Yin, B. Huang, T. Kelestemur, J. Wang, and Y. Li. Gendp: 3d semantic fields for category-level generalizable diffusion policy. In *8th Annual Conference on Robot Learning*, volume 2, 2024.

[7] C.-C. Hsu, B. Wen, J. Xu, Y. Narang, X. Wang, Y. Zhu, J. Biswas, and S. Birchfield. Spot: Se (3) pose trajectory diffusion for object-centric manipulation. *arXiv preprint arXiv:2411.00965*, 2024.

[8] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

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

[10] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. Fan, and Y. Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. *arXiv preprint arXiv:2410.24185*, 2024.

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

[12] S. Yang, Y. Du, S. K. S. Ghasemipour, J. Tompson, L. P. Kaelbling, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. In *The Twelfth International Conference on Learning Representations*, 2024.

[13] K. Li, P. Li, T. Liu, Y. Li, and S. Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2025.

[14] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang. An embodied generalist agent in 3d world. In *Proceedings of the 41st International Conference on Machine Learning*, pages 20413–20451, 2024.

[15] P. Li, T. Liu, Y. Li, Y. Geng, Y. Zhu, Y. Yang, and S. Huang. Gendexgrasp: Generalizable dexterous grasping. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8068–8074. IEEE, 2023.

[16] Y. Li, B. Liu, Y. Geng, P. Li, Y. Yang, Y. Zhu, T. Liu, and S. Huang. Grasp multiple objects with one hand. *IEEE Robotics and Automation Letters*, 2024.

[17] M. Torne, A. Simeonov, Z. Li, A. Chan, T. Chen, A. Gupta, and P. Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024.

[18] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

[19] Y. Mu, T. Chen, S. Peng, Z. Chen, Z. Gao, Y. Zou, L. Lin, Z. Xie, and P. Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version). *arXiv preprint arXiv:2409.02920*, 2024.

[20] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

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

[22] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[23] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

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

[25] A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[26] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*, 2024.

[27] Y. Zhu, Z. Jiang, P. Stone, and Y. Zhu. Learning generalizable manipulation policies with object-centric 3d representations. In *Conference on Robot Learning*, pages 3418–3433. PMLR, 2023.

[28] J. Shi, J. Qian, Y. J. Ma, and D. Jayaraman. Composing pre-trained object-centric representations for robotics from" what" and" where" foundation models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15424–15432. IEEE, 2024.

[29] J. Qian, Y. Li, B. Bucher, and D. Jayaraman. Task-oriented hierarchical object decomposition for visuomotor control. In *8th Annual Conference on Robot Learning*, 2024.

[30] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[31] J. Zhu, Y. Ju, J. Zhang, M. Wang, Z. Yuan, K. Hu, and H. Xu. Densematcher: Learning 3d semantic correspondence for category-level manipulation from a single demo. *arXiv preprint arXiv:2412.05268*, 2024.

[32] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.

[33] S. Tyree, J. Tremblay, T. To, J. Cheng, T. Mosier, J. Smith, and S. Birchfield. 6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13081–13088. IEEE, 2022.

[34] T. Migimatsu and J. Bohg. Object-centric task and motion planning in dynamic environments. *IEEE Robotics and Automation Letters*, 5(2):844–851, 2020.

[35] D. Wang, C. Devin, Q.-Z. Cai, F. Yu, and T. Darrell. Deep object-centric policies for autonomous driving. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8853–8859. IEEE, 2019.

[36] C. Devin, P. Abbeel, T. Darrell, and S. Levine. Deep object-centric representations for generalizable robot learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7111–7118. IEEE, 2018.

[37] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. *Advances in neural information processing systems*, 33:11525–11538, 2020.

[38] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[39] C. Wang, R. Wang, A. Mandlekar, L. Fei-Fei, S. Savarese, and D. Xu. Generalization through hand-eye coordination: An action space for learning spatially-invariant visuomotor control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8913–8920. IEEE, 2021.

[40] N. Heravi, A. Wahid, C. Lynch, P. Florence, T. Armstrong, J. Tompson, P. Sermanet, J. Bohg, and D. Dwibedi. Visuomotor control in multi-object scenes using object-aware representations. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9515–9522. IEEE, 2023.

[41] A. Didolkar, A. Zadaianchuk, A. Goyal, M. Mozer, Y. Bengio, G. Martius, and M. Seitzer. Zero-shot object-centric representation learning. *arXiv preprint arXiv:2408.09162*, 2024.

[42] J. Yoon, Y.-F. Wu, H. Bae, and S. Ahn. An investigation into pre-training object-centric representations for reinforcement learning. *arXiv preprint arXiv:2302.04419*, 2023.

[43] N. Gao, V. A. Ngo, H. Ziesche, and G. Neumann. Sa6d: Self-adaptive few-shot 6d pose estimator for novel and occluded objects. In *7th Annual Conference on Robot Learning*, 2023.

[44] Q. Yi, R. Zhang, J. Guo, X. Hu, Z. Du, Q. Guo, Y. Chen, et al. Object-category aware reinforcement learning. *Advances in Neural Information Processing Systems*, 35:36453–36465, 2022.

[45] Stability. Stable diffusion v1.5 model card. https://huggingface.co/runwayml/stable-diffusion-v1-5, 2022. Accessed: 2024-09-05.

[46] M. Li, T. Yang, H. Kuang, J. Wu, Z. Wang, X. Xiao, and C. Chen. Controlnet++: Improving conditional controls with efficient consistency feedback: Project page: liming-ai. github. io/controlnet_plus_plus. In *European Conference on Computer Vision*, pages 129–147. Springer, 2024.

[47] D. Zavadski, J.-F. Feiden, and C. Rother. Controlnet-xs: Designing an efficient and effective architecture for controlling text-to-image diffusion models. *arXiv preprint arXiv:2312.06573*, 2023.

[48] Y. Guo, C. Yang, A. Rao, Z. Liang, Y. Wang, Y. Qiao, M. Agrawala, D. Lin, and B. Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

[49] T. Wang, L. Li, K. Lin, Y. Zhai, C.-C. Lin, Z. Yang, H. Zhang, Z. Liu, and L. Wang. Disco: Disentangled control for realistic human dance generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9326–9336, 2024.

[50] O. Bar-Tal, H. Chefer, O. Tov, C. Herrmann, R. Paiss, S. Zada, A. Ephrat, J. Hur, G. Liu, A. Raj, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024.

[51] W. Dai, L.-H. Chen, J. Wang, J. Liu, B. Dai, and Y. Tang. Motionlcm: Real-time controllable motion generation via latent consistency model. In *European Conference on Computer Vision*, pages 390–408. Springer, 2024.

[52] Y. Xie, V. Jampani, L. Zhong, D. Sun, and H. Jiang. Omnicontrol: Control any joint at any time for human motion generation. In *The Twelfth International Conference on Learning Representations*, 2024.

[53] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[54] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[55] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[56] A. Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[57] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

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

[59] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting twenty-thousand classes using image-level supervision. In *European Conference on Computer Vision*, pages 350–368. Springer, 2022.

[60] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[61] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024.

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

[63] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021.

[64] Meta Platforms, Inc. Meta Quest: Virtual Reality Headset. https://www.meta.com/quest/, n.d. Accessed: 2025-05-08.

[65] Astribot, Inc. Astribot S1: AI Robotic Partner. https://www.astribot.com/product-en, 2024. Accessed: 2025-05-06.

A Preliminary of Diffusion Policy

Diffusion policy [20] formulates the visuomotor policy $\boldsymbol{\pi}$ as the DDPM [60], which can model complex multimodal action distributions and facilitate a stable training behavior. DDPM performs K iterations of a denoising process, starting from a Gaussian noise $\boldsymbol{x}^K \sim \mathcal{N}(0, I)$ and evolving toward the desired output $\boldsymbol{x}^0 \sim q_\theta(\boldsymbol{x}^0)$. The denoising process is described by the following equation:

$$\boldsymbol{x}^{k-1} = \alpha\left(\boldsymbol{x}^k - \beta\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}^k, k\right)\right) + \sigma\mathcal{N}(0, I), \tag{4}$$

where α, β, and σ are functions of the timestep k, collectively known as the noise schedule, and the $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ is the distribution shift prediction network with the trainable parameter θ.

The training objective is to minimize the variational lower bound of KL-divergence between the given data distribution $p(\boldsymbol{x}^0)$ and the θ-parameterized distribution $q_\theta(\boldsymbol{x}^0)$. As shown in [60], the loss function can be simplified as:

$$\mathcal{L} = \mathbb{E}_{t\sim[1,K],\boldsymbol{x}^0,\epsilon^k}\left[\|\epsilon^k - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{x}^0 + \epsilon^k, k\right)\|^2\right]. \tag{5}$$

Diffusion policy represents the robot actions $\boldsymbol{a}_{\boldsymbol{t:t+T_a}}$ as the model output x and conditions the denoising process on the robot observations $\boldsymbol{o}_{\boldsymbol{t:t-T_o}}$, where $\boldsymbol{a_t} \in \boldsymbol{\mathcal{A}}$, $\boldsymbol{o_t} \in \boldsymbol{\mathcal{O}}$, T_a and T_o denote the horizon lengths of the action and observation sequences. For convenience, we use $\boldsymbol{A_t}$ and $\boldsymbol{O_t}$ to represent the action and observation sequences in the following discussion. The DDPM is naturally extended to approximate the conditional distribution $p(\boldsymbol{A_t} \mid \boldsymbol{O_t})$ for planning. To capture the conditional actions distribution, the denoising process is modified from Eq. (4):

$$\boldsymbol{A_t}^{k-1} = \alpha\left(\boldsymbol{A_t}^k - \beta\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{A_t}^k, k\right)\right) + \sigma\mathcal{N}(0, I). \tag{6}$$

The training loss is modified from Eq. (5):

$$\mathcal{L} = \mathbb{E}_{t\sim[1,K],\boldsymbol{A_t}^0,\epsilon^k}\left[\|\epsilon^k - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\boldsymbol{A_t}^0 + \epsilon^k, \boldsymbol{O_t}, k\right)\|^2\right]. \tag{7}$$

In practice, we exclude observation features from the denoising process for better accommodation of real-time robot control, while the formulation remains the same.

B Further Explanation of ControlNet-style Fine-tuning

A common misunderstanding with zero-initialized weights and biases is that they produce zero gradients and are, therefore, untrainable. We demonstrate that the additional KV-projection layers $(\mathbf{W_z}, \mathbf{B_z})$ and the object-centric representations $\boldsymbol{Z}$ can be optimized despite their zero initialization, which is similar to the case in ControlNet [30].

Let $\frac{\partial\mathcal{L}}{\partial\mathbf{V_z}}$ denote the upstream gradient from the loss $\mathcal{L}$. The gradients for $\mathbf{W_z}$ and $\mathbf{B_z}$ are:

$$\begin{cases} \dfrac{\partial\mathcal{L}}{\partial\mathbf{W_z}} = \displaystyle\sum_{p,i} \frac{\partial\mathcal{L}}{\partial\boldsymbol{V_z}_{p,i}} \cdot \boldsymbol{Z}_{p,i} \\ \dfrac{\partial\mathcal{L}}{\partial\mathbf{B_z}} = \displaystyle\sum_{p,i} \frac{\partial\mathcal{L}}{\partial\boldsymbol{V_z}_{p,i}} \cdot 1 \end{cases} \tag{8}$$

Since $\boldsymbol{Z}$ is non-zero, $\frac{\partial\mathcal{L}}{\partial\mathbf{W_z}} \neq \mathbf{0}$ if $\frac{\partial\mathcal{L}}{\partial\boldsymbol{V_z}} \neq \mathbf{0}$. Similarly, $\frac{\partial\mathcal{L}}{\partial\mathbf{B_z}}$ accumulates non-zero gradients. After one gradient step:

$$\begin{cases} \mathbf{W_z}^* = \mathbf{W_z} - \beta_l \cdot \dfrac{\partial\mathcal{L}}{\partial\mathbf{W_z}} \neq \mathbf{0} \\ \mathbf{B_z}^* = \mathbf{B_z} - \beta_l \cdot \dfrac{\partial\mathcal{L}}{\partial\mathbf{B_z}} \neq \mathbf{0} \end{cases} \tag{9}$$

This ensures $\boldsymbol{K_z}^*$ and $\boldsymbol{V_z}^*$ become non-zero, allowing the dual-attention to incorporate $\boldsymbol{Z}$.

Considering $\boldsymbol{Z}$ is learnable, its gradient is:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{Z}} = \mathbf{W_z}^T \cdot \frac{\partial\mathcal{L}}{\partial\boldsymbol{V_z}}. \tag{10}$$

Since $\mathbf{W_z}^* \neq \mathbf{0}$, $\boldsymbol{Z}$ receives non-zero gradients and is updated accordingly. This aligns with the zero-convolution principle, where gradients persist despite zero-initialized parameters. We fine-tune the expert policy using the conditional denoising loss as defined in Eq. (7).

With the ControlNet-style fine-tuning, we efficiently integrate additional object-centric conditions into the pre-trained visuomotor policy. This approach ensures that when the KV-projection layers are zero-initialized in the dual cross-attention module, the deep neural features remain unaffected prior to any optimization. The capabilities, functionality, and output action quality of the pre-trained visuomotor modules are perfectly preserved, while further optimization becomes as efficient as standard fine-tuning. This allows ControlVLA to simultaneously leverage the advantages of large-scale pre-training and object-centric representations, accelerating real-world robot adoption by significantly reducing the data requirements for task deployment.

C Implementation Details of ControlVLA

In the main paper Sec. 4.1, we pre-train the policy $\boldsymbol{\pi_g}$ on the full DROID dataset [61], using the wrist camera image $\boldsymbol{I_t}$, end-effector poses and gripper widths $\boldsymbol{q_t}$, and episode language descriptions ℓ_t. The observation and action horizons are set to $T_o = 2$ and $T_a = 16$. The pre-trained policy, implemented as a Diffusion Transformer [20] with 29M parameters, uses a CLIP [62] ViT-B/16 vision encoder and a Transformer text encoder. We pre-train $\boldsymbol{\pi_g}$ with AdamW (learning rates: 1×10^{-4} for denoising model, 3×10^{-5} for vision; text encoder frozen) on 4 NVIDIA A800 GPUs for 3 days. In Sec. 4.2, we extract object-centric representations from raw images. In Sec. 4.3, we fine-tune $\boldsymbol{\pi_e}$ on evaluation tasks, adding $\sim$5M parameters. Fine-tuning uses the same settings as pre-training and runs on a single NVIDIA A800 GPU for 12 hours.

D Details of Experiments

D.1 Data Collection

We collect a small set of demonstrations for each evaluation task. For short-horizon tasks, we use UMI [8], an arm-agnostic data collection system with a hand-held gripper for efficient demonstration gathering. UMI features a wrist-mounted GoPro camera that captures RGB images and 6D end-effector pose trajectories using visual SLAM [63] fused with onboard IMU data. For long-horizon tasks, we use Meta Quest [64] to teleoperate the AstriBot-S1 [65], enabling immersive, low-latency 6DoF control via VR motion tracking. The operator's hand movements are mapped to the robot in real time, allowing intuitive and precise demonstrations. AstriBot-S1 is a more human-like robot with spherical joints at the shoulder and elbow, closely mimicking the range and fluidity of human articulation. The number of demonstrations per evaluation task is detailed in the main paper Tab. 1.

D.2 Evaluation Setup and Protocol

For short-horizon tasks, we deploy a Franka Emika FR3 arm with a Panda gripper and the same GoPro camera used during data collection for policy inference. For long-horizon tasks, we execute the policy on the AstriBot-S1 robot, which is equipped with a wrist-mounted RealSense camera for capturing RGB observations. Task success rate serves as the primary evaluation metric. Each trial is terminated if the policy shows no sign of progress, the robot enters a potentially unsafe interaction with the environment, or the task is completed. All evaluations are conducted in the same environment used for data collection, but with randomized initial configurations of both the robot and the objects to ensure robustness and generalization. Fig. 7 provides an overview of the evaluation setup, and Fig. 8 shows the initial objects and robot states distribution of policy evaluation.

D.3 Implementation Details of ControlVLA@π_0

We further adapt and evaluate our ControlVLA method on a more general pre-trained VLA model, π_0 [26], referred to as ControlVLA@π_0. The π_0 model leverages a pre-trained vision-language

Fig. 7: **Evaluation Setup.** The evaluation uses two robot platforms: the Franka Panda (left), for 6 short-horizon tasks; and the AstriBot-S1 (right), for 2 long-horizon tasks.

Fig. 8: **Initial state distribution of policy evaluation.**

backbone and introduces a separate *action expert* that outputs continuous actions using flow matching. ControlVLA@π_0 extends this architecture by incorporating an additional *object expert* and a set of zero-convolution layers to progressively inject object-centric representation guidance. These zero-convolution layers ensure that the object-centric cues are integrated as additional conditions without disrupting the learned action prior, enabling robust skill learning from limited demonstrations. We conduct the comparison studies over 20 trials across 4 sub-tasks, each fine-tuned with limited demonstrations, consistent with the setup used in the main paper.