

## 1 A More Experiments on $\pi_0$

2 We further adapt and evaluate our ControlVLA method on a more general pre-trained Vision-  
 3 Language-Action models (VLA model),  $\pi_0$  [1], referred to as ControlVLA@ $\pi_0$ . The  $\pi_0$  model  
 4 leverages a pre-trained vision-language backbone and introduces a separate *action expert* that out-  
 5 puts continuous actions using flow matching. ControlVLA@ $\pi_0$  extends this architecture by in-  
 6 corporating an additional *object expert* and a set of zero-convolution layers to progressively inject  
 7 object-centric representation guidance. These zero-convolution layers ensure that the object-centric  
 8 cues are integrated as additional conditions without disrupting the learned action prior, enabling ro-  
 9 bust skill learning from limited demonstrations. We conduct the comparison studies over 20 trials  
 10 across 4 sub-tasks, each fine-tuned with limited demonstrations, consistent with the setup used in  
 11 the main paper.

12 As shown in Tab. 1, our adaptation method ControlVLA@ $\pi_0$  consistently outperforms the fine-tuned  
 13  $\pi_0$ , demonstrating that ControlVLA can serve as a plug-in module to enhance performance across  
 14 a broader range of pre-trained VLA model models. The improvement is most pronounced on the  
 15 OrganizeScissors task, highlighting ControlVLA’s ability to provide more precise guidance  
 for fine-grained manipulation in data-scarce scenarios.

Tab. 1: Task success rates of  $\pi_0$  and ControlVLA@ $\pi_0$  across various tasks.

|                     | Organize<br>Toy | Organize<br>Scissors | Open<br>Cabinet | Fold<br>Clothes | Overall      |
|---------------------|-----------------|----------------------|-----------------|-----------------|--------------|
| $\pi_0$ [1]         | 55.0%           | 15.0%                | 45.0%           | 40.0%           | 38.6%        |
| ControlVLA@ $\pi_0$ | 85.0%           | 80.0%                | 85.0%           | 75.0%           | <b>81.3%</b> |

## 17 B Preliminary of Diffusion Policy

18 Diffusion policy [2] formulates the visuomotor policy  $\pi$  as the Denoising Diffusion Probabilistic  
 19 Model (DDPM) [3], which can model complex multimodal action distributions and facilitate a stable  
 20 training behavior. DDPM performs  $K$  iterations of a denoising process, starting from a Gaussian  
 21 noise  $\mathbf{x}^K \sim \mathcal{N}(0, I)$  and evolving toward the desired output  $\mathbf{x}^0 \sim q_\theta(\mathbf{x}^0)$ . The denoising process  
 22 is described by the following equation:

$$\mathbf{x}^{k-1} = \alpha(\mathbf{x}^k - \beta\epsilon_\theta(\mathbf{x}^k, k)) + \sigma\mathcal{N}(0, I), \quad (1)$$

23 where  $\alpha, \beta$ , and  $\sigma$  are functions of the timestep  $k$ , collectively known as the noise schedule, and the  
 24  $\epsilon_\theta$  is the distribution shift prediction network with the trainable parameter  $\theta$ .

25 The training objective is to minimize the variational lower bound of KL-divergence between the  
 26 given data distribution  $p(\mathbf{x}^0)$  and the  $\theta$ -parameterized distribution  $q_\theta(\mathbf{x}^0)$ . As shown in [3], the  
 27 loss function can be simplified as:

$$\mathcal{L} = \mathbb{E}_{t \sim [1, K], \mathbf{x}^0, \epsilon^k} [\|\epsilon^k - \epsilon_\theta(\mathbf{x}^0 + \epsilon^k, k)\|^2]. \quad (2)$$

28 Diffusion policy represents the robot actions  $\mathbf{a}_{t:t+T_a}$  as the model output  $x$  and conditions the  
 29 denoising process on the robot observations  $\mathbf{o}_{t:t-T_o}$ , where  $\mathbf{a}_t \in \mathcal{A}$ ,  $\mathbf{o}_t \in \mathcal{O}$ ,  $T_a$  and  $T_o$  denote  
 30 the horizon lengths of the action and observation sequences. For convenience, we use  $\mathbf{A}_t$  and  
 31  $\mathbf{O}_t$  to represent the action and observation sequences in the following discussion. The DDPM is  
 32 naturally extended to approximate the conditional distribution  $p(\mathbf{A}_t | \mathbf{O}_t)$  for planning. To capture  
 33 the conditional actions distribution, the denoising process is modified from Eq. (1):

$$\mathbf{A}_t^{k-1} = \alpha(\mathbf{A}_t^k - \beta\epsilon_\theta(\mathbf{A}_t^k, k)) + \sigma\mathcal{N}(0, I). \quad (3)$$

34 The training loss is modified from Eq. (2):

$$\mathcal{L} = \mathbb{E}_{t \sim [1, K], \mathbf{A}_t^0, \epsilon^k} [\|\epsilon^k - \epsilon_\theta(\mathbf{A}_t^0 + \epsilon^k, \mathbf{O}_t, k)\|^2]. \quad (4)$$

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

## 37 C Further Explanation of ControlNet-style Fine-tuning

38 A common misunderstanding with zero-initialized weights and biases is that they produce zero  
 39 gradients and are, therefore, untrainable. We demonstrate that the additional KV-projection layers  
 40 ( $\mathbf{W}_z, \mathbf{B}_z$ ) and the object-centric representations  $\mathbf{Z}$  can be optimized despite their zero initialization,  
 41 which is similar to the case in ControlNet [4].

42 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:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \mathbf{W}_z} = \sum_{p,i} \frac{\partial \mathcal{L}}{\partial \mathbf{V}_{z_{p,i}}} \cdot \mathbf{z}_{p,i} \\ \frac{\partial \mathcal{L}}{\partial \mathbf{B}_z} = \sum_{p,i} \frac{\partial \mathcal{L}}{\partial \mathbf{V}_{z_{p,i}}} \cdot 1 \end{cases} \quad (5)$$

43 Since  $\mathbf{Z}$  is non-zero,  $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_z} \neq \mathbf{0}$  if  $\frac{\partial \mathcal{L}}{\partial \mathbf{V}_z} \neq \mathbf{0}$ . Similarly,  $\frac{\partial \mathcal{L}}{\partial \mathbf{B}_z}$  accumulates non-zero gradients. After  
 44 one gradient step:

$$\begin{cases} \mathbf{W}_z^* = \mathbf{W}_z - \beta_l \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{W}_z} \neq \mathbf{0} \\ \mathbf{B}_z^* = \mathbf{B}_z - \beta_l \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{B}_z} \neq \mathbf{0} \end{cases} \quad (6)$$

45 This ensures  $\mathbf{K}_z^*$  and  $\mathbf{V}_z^*$  become non-zero, allowing the dual-attention to incorporate  $\mathbf{Z}$ .

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

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Z}} = \mathbf{W}_z^T \cdot \frac{\partial \mathcal{L}}{\partial \mathbf{V}_z}. \quad (7)$$

47 Since  $\mathbf{W}_z^* \neq \mathbf{0}$ ,  $\mathbf{Z}$  receives non-zero gradients and is updated accordingly. This aligns with the  
 48 zero-convolution principle, where gradients persist despite zero-initialized parameters. We fine-tune  
 49 the expert policy using the conditional denoising loss as defined in Eq. (4).

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

## 58 D Implementation Details of ControlVLA

59 In the main paper Sec. 4.1, we pre-train the policy  $\pi_g$  on the full DROID dataset [5], using the  
 60 wrist camera image  $\mathbf{I}_t$ , end-effector poses and gripper widths  $\mathbf{q}_t$ , and episode language descriptions  
 61  $\ell_t$ . The observation and action horizons are set to  $T_o = 2$  and  $T_a = 16$ . The pre-trained policy,  
 62 implemented as a Diffusion Transformer [2] with 29M parameters, uses a CLIP [6] ViT-B/16 vision  
 63 encoder and a Transformer text encoder. We pre-train  $\pi_g$  with AdamW (learning rates:  $1 \times 10^{-4}$  for  
 64 denoising model,  $3 \times 10^{-5}$  for vision; text encoder frozen) on 4 NVIDIA A800 GPUs for 3 days.  
 65 In Sec. 4.2, we extract object-centric representations from raw images. In Sec. 4.3, we fine-tune  $\pi_e$   
 66 on evaluation tasks, adding  $\sim 5$ M parameters. Fine-tuning uses the same settings as pre-training and  
 67 runs on a single NVIDIA A800 GPU for 12 hours.

## 68 E Details of Experiments

### 69 E.1 Data Collection

70 We collect a small set of demonstrations for each evaluation task. For short-horizon tasks, we use  
 71 UMI [7], an arm-agnostic data collection system with a hand-held gripper for efficient demonstration



Fig. 1: **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.

72 gathering. UMI features a wrist-mounted GoPro camera that captures RGB images and 6D end-  
 73 effector pose trajectories using visual SLAM [8] fused with onboard IMU data. For long-horizon  
 74 tasks, we use Meta Quest [9] to teleoperate the AstriBot-S1 [10], enabling immersive, low-latency  
 75 6DoF control via VR motion tracking. The operator’s hand movements are mapped to the robot in  
 76 real time, allowing intuitive and precise demonstrations. AstriBot-S1 is a more human-like robot  
 77 with spherical joints at the shoulder and elbow, closely mimicking the range and fluidity of human  
 78 articulation. The number of demonstrations per evaluation task is detailed in the main paper Tab. 1.

## 79 E.2 Evaluation Setup and Protocol

80 For short-horizon tasks, we deploy a Franka Emika FR3 arm with a Panda gripper and the same  
 81 GoPro camera used during data collection for policy inference. For long-horizon tasks, we execute  
 82 the policy on the AstriBot-S1 robot, which is equipped with a wrist-mounted RealSense camera  
 83 for capturing RGB observations. Task success rate serves as the primary evaluation metric. Each  
 84 trial is terminated if the policy shows no sign of progress, the robot enters a potentially unsafe  
 85 interaction with the environment, or the task is completed. All evaluations are conducted in the same  
 86 environment used for data collection, but with randomized initial configurations of both the robot  
 87 and the objects to ensure robustness and generalization. Fig. 1 provides an overview of the evaluation  
 88 setup, and Fig. 2 shows the initial objects and robot states distribution of policy evaluation.



Fig. 2: **Initial state distribution of policy evaluation.**

## References

- [1] 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.
- [2] 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.
- [3] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] Meta Platforms, Inc. Meta Quest: Virtual Reality Headset. <https://www.meta.com/quest/>, n.d. Accessed: 2025-05-08.
- [10] Astribot, Inc. Astribot S1: AI Robotic Partner. <https://www.astribot.com/product-en>, 2024. Accessed: 2025-05-06.