arXiv:2503.10631v1 [cs.CV] 13 Mar 2025

HybridVLA: Collaborative Diffusion and Autoregression in a Unified Vision-Language-Action Model

Jiaming Liu[1,2*], Hao Chen[3*], Pengju An[1,2†], Zhuoyang Liu[1†], Renrui Zhang[3‡], Chenyang Gu[1,2], Xiaoqi Li[1], Ziyu Guo[3], Sixiang Chen[1,2], Mengzhen Liu[1,2], Chengkai Hou[1,2], Mengdi Zhao[2], KC alex Zhou[1], Pheng-Ann Heng[3], Shanghang Zhang[1,2] ✉

[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University;
[2]Beijing Academy of Artificial Intelligence (BAAI); [3]CUHK
* Equal contribution, † Equal technical contribution, ‡ Project lead, ✉ Corresponding author

Project web page: hybrid-vla.github.io

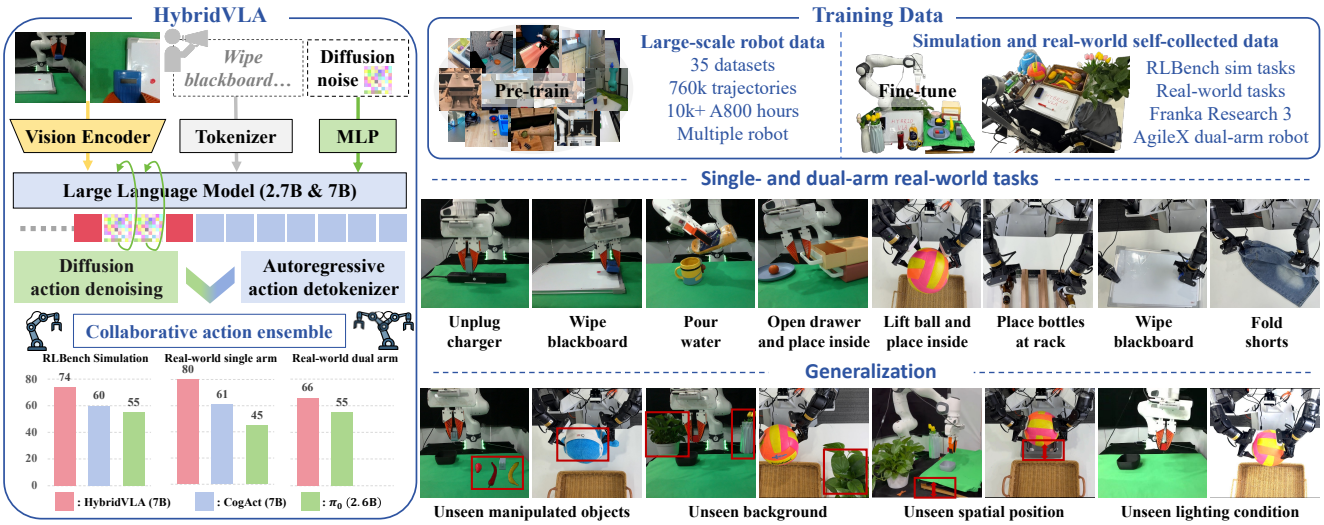


Figure 1. HybridVLA innovatively integrates diffusion and autoregressive action prediction within a single LLM, fully leveraging the continuity and probabilistic nature of diffusion alongside the reasoning capabilities of autoregressive modeling. It undergoes pretraining on large, diverse, cross-embodied real-world robotic datasets and is further fine-tuned on both simulation and self-collected real-world data. HybridVLA achieves remarkable performance across various tasks, demonstrating strong generalization to unseen manipulated objects, backgrounds, spatial positions, and lighting conditions.

Abstract

Recent advancements in vision-language models (VLMs) for common-sense reasoning have led to the development of vision-language-action (VLA) models, enabling robots to perform generalized manipulation. Although existing autoregressive VLA methods leverage large-scale pretrained knowledge, they disrupt the continuity of actions. Meanwhile, some VLA methods incorporate an additional diffusion head to predict continuous actions, relying solely on VLM-extracted features, which limits their reasoning capabilities. In this paper, we introduce HybridVLA, a unified framework that seamlessly integrates the strengths of both autoregressive and diffusion policies within a single large language model, rather than simply connecting them. To bridge the generation gap, a collaborative training recipe is proposed that injects the diffusion modeling directly into the next-token prediction. With this recipe, we find that these two forms of action prediction not only reinforce each other but also exhibit varying performance across different tasks. Therefore, we design a collaborative action ensemble mechanism that adaptively fuses these two predictions, leading to more robust control. In experiments, HybridVLA outperforms previous state-of-the-art VLA methods across various simulation and real-world tasks, including both single-arm and dual-arm robots, while demonstrating stable manipulation in previously unseen configurations.

1. Introduction

Recently, vision-language models (VLMs) [3, 5, 20, 49, 56, 107–109] have demonstrated exceptional abilities in instruction-following and common-sense reasoning, driven by pretraining on internet-scale image-text pairs. Build-

ing on this success, several studies have extended VLMs into vision-language-action (VLA) models, enabling them to generate action plans[1, 7, 17, 32] or predict SE(3) poses [10, 45, 54]. VLA models enable robots to interpret visual observations and language conditions, generating generalizable actions for control. Therefore, effectively harnessing the inherent capabilities of VLMs to develop VLA models for stable manipulation in dynamic environments is essential [53].

On the one hand, some existing VLA methods [10, 45, 54, 72] quantize continuous actions into discrete bins, replacing part of the vocabulary in large language models (LLMs). These autoregressive approaches mimic the next-token prediction of VLMs, effectively harnessing their large-scale pretrained knowledge while preserving reasoning capabilities. Although such methods enable generalized manipulation skills [45], the quantization process disrupts the continuity of action poses [95]. On the other hand, some VLA approaches [31, 52, 58, 97] incorporate a policy head (e.g., MLP or LSTM [23]) to transform LLM output embeddings into continuous action poses. However, these regressive methods overlook the scalability of the policy head and fail to incorporate probabilistic action representations [13, 50].

Building on the success of diffusion models in content generation [26, 29, 30, 71, 78], diffusion policies have recently been introduced in robotic imitation learning [13, 43, 69, 77, 100, 105]. Unlike regressive deterministic policy heads, π_0[8], CogACT [50], and DiVLA [95] incorporate a diffusion head after VLMs, leveraging probabilistic noise-denoising for action prediction. While diffusion-based VLA methods enable precise manipulation, the diffusion head operates independently of the VLM, relying solely on VLM-extracted features as input conditions. Consequently, it fails to fully leverage the VLM's reasoning capabilities. Given these advantages and limitations, a question arises: "*How can we elegantly construct a unified VLA model that seamlessly integrates the strengths of both autoregressive and diffusion policies, rather than simply concatenating them?*"

To achieve this, we propose HybridVLA, a unified framework that equips VLMs with both diffusion and autoregressive action prediction capabilities, enabling robust execution across diverse and complex manipulation tasks. Unlike previous diffusion-based VLA methods [8, 50], which append independent diffusion head after the LLM, HybridVLA seamlessly integrates diffusion modeling into the autoregressive next-token prediction within a single LLM, as shown in Figure 1. Specifically, we introduce a collaborative training recipe that encodes the diffusion-noised action as a continuous vector and projects it into the LLM's word embedding space. To ensure consistency when combining the two generation methods, a token sequence formulation is designed to systematically organize multimodal input, diffusion action, and autoregressive action tokens while linking them through specialized marker tokens. With this design, we observe that the two action predictions not only reinforce each other but also exhibit varying performance across different tasks. For instance, the diffusion prediction excels in intricate tasks, while the autoregressive prediction performs better in tasks requiring rich semantic understanding. Therefore, a collaborative action ensemble mechanism is proposed, where the two predictions are adaptively fused based on autoregressive action token confidence, improving robustness in manipulation.

To enhance HybridVLA's generalization, we employ a step-by-step training approach [8], first performing large-scale pretraining, followed by fine-tuning on downstream tasks, as shown at the top of Figure 1. In addition to being initialized with parameters from a pretrained VLM [42], our model undergoes further pretraining on large, diverse, cross-embodiment robotic datasets, including Open X-Embodiment [67], DROID [44], and ROBOMIND [98], encompassing 760K trajectories and over 10K A800 GPU training hours. Subsequently, HybridVLA is fine-tuned on high-quality simulation data [34] and self-collected real-world data, achieving state-of-the-art (SOTA) manipulation performance across a variety of tasks with both single-arm and dual-arm robots. Meanwhile, HybridVLA demonstrates sufficient generalization capabilities to unseen manipulated objects, backgrounds, spatial positions, and lighting conditions during real-world testing, highlighting the effectiveness of our collaborative model design and training recipe. To optimize inference speed, we introduce HybridVLA-dif, which integrates diffusion and autoregressive generation during training but relies exclusively on diffusion-based actions for inference at 9.4 Hz. In summary, our contributions are as follows:

- We propose HybridVLA, a unified model that seamlessly integrates diffusion and autoregressive action prediction within a single LLM, effectively combining the continuous nature of diffusion-based actions with the reasoning capabilities of autoregressive generation.
- We introduce a collaborative training recipe that bridges the gap between the two action generation approaches, enabling mutual reinforcement. Additionally, we propose a collaborative action ensemble mechanism that adaptively fuses diffusion- and autoregressive-based actions, enhancing manipulation robustness.
- Our proposed method achieves SOTA performance across diverse tasks while demonstrating strong generalization to unseen configurations.

2. Related Work

Traditional robotic manipulation primarily relies on state-based reinforcement learning [4, 21, 40, 104], whereas re-

cent approaches [9, 13, 19, 90] integrate visual observations for imitation learning. Building on the strong reasoning capabilities of vision-language models (VLMs) [3, 25, 39, 48, 49, 56], recent research has integrated them into robotic manipulation [51, 57, 101, 102].

Vision-language-action (VLA) models. Some studies [1, 17, 32, 33] enable robots to interpret both language and visual observations, automatically generating task plans. Meanwhile, vision-language-action (VLA) models leverage the inherent reasoning abilities of VLMs to predict low-level SE(3) poses. Specifically, RT2[10] quantizes 7-DoF actions into discrete bins for autoregressive pose prediction. Building on this, ManipLLM[54] incorporates affordance priors through chain-of-thought reasoning, while OpenVLA[45] performs large-scale pretraining on the Open X-Embodiment dataset[67], enhancing generalization capabilities. FAST [72] applies the discrete cosine transform to enable fast and scalable training of autoregressive-based VLA models. To support continuous action prediction, some VLA approaches [31, 52, 58, 97] incorporate a policy head, such as an MLP or LSTM [23], and use regression loss for imitation learning. However, quantization in autoregressive methods disrupts action continuity, limiting intricate manipulation, while regressive methods fail to incorporate probabilistic action representations.

Diffusion models in robotics. Building on the success of diffusion models in content generation [29, 30, 71, 78], diffusion policies have been applied in robotics, including reinforcement learning [2, 94], imitation learning [13, 70, 73, 77, 100], grasping [86, 92, 99], and motion planning [36, 80]. Following this, 3D Diffusion Actor [43] and DP3[13] employ diffusion models to interpret point cloud data. Octo[89] and RDT-1B[59] enhance a transformer model with a diffusion head to predict flexible actions.

Diffusion-based VLA models. To integrate diffusion with VLMs, π_0[8] adds a diffusion expert head that generates actions through flow matching, while TinyVLA [96] incorporates a simple diffusion head after the lightweight VLM. CogACT[50] and DiVLA [95] decouple reasoning and action prediction into the VLM and an injected diffusion head, respectively. However, in these methods, the diffusion head functions as a separate module, relying solely on LLM-extracted features for action conditions, limiting its use of the VLM's pretrained knowledge and reasoning capabilities. To fully harness the strengths of both diffusion and autoregressive-based methods, we integrate diffusion into next-token prediction within a single LLM.

3. HybridVLA Method

Overview. Existing diffusion-based VLA methods [8, 50, 95] append a separate diffusion head after the VLM, using VLM-extracted features as conditions for the diffusion process. However, these methods fail to fully exploit the VLM's inherent reasoning capabilities derived from internet-scale pretraining. In contrast, HybridVLA equips a single LLM with both diffusion and autoregressive action generation capabilities. To construct HybridVLA, we first describe the model architecture in Section 3.1. Since simply merging the two generation methods could cause inconsistency, we introduce a collaborative training recipe in Section 3.2. To further enhance robustness, we propose a collaborative action ensemble mechanism in Section 3.3.

Problem Statement. At time t, each demonstration consists of image observations o_t, language description l_t, and the current robot state r_t. Our model π aims to predict action a to control the robot arms, which can be formulated as:

$$\pi : (o_t, l_t, r_t) \rightarrow a_{t+1}$$

Following [45, 50], the action a represents the end-effector pose, which uses 7-DOF and 14-DOF for single-arm and dual-arm control, respectively. Each 7-DOF action includes 3-DOF for relative translation offsets ($[\Delta x, \Delta y, \Delta z] \in \mathbb{R}^3$), 3-DOF for rotation (Euler angles $\in \mathbb{R}^3$), and 1-DOF for the gripper state (open/closed $\in \mathbb{R}^1$). The ground truth (GT) and the model-predicted action are in SE(3), formulated as:

$$a = [\Delta x, \Delta y, \Delta z, Roll, Pitch, Yaw, 0/1]$$

3.1. HybridVLA Architecture

This section presents the architecture and workflow of HybridVLA, available in two model sizes, using 2.7B and 7B large language models (LLMs). Following [45], HybridVLA inherits a base architecture adapted from the Prismatic VLMs [42], leveraging its internet-scale pretrained parameters. We first introduce the two basic components—vision encoders and the LLM—as shown in Figure 2.

Vision encoders. HybridVLA leverages powerful vision encoder combinations, such as DINOv2 [68] and SigLIP [106], to capture rich semantic features $f_d \in \mathbb{R}^{B\times N_v\times 1024}$ and $f_s \in \mathbb{R}^{B\times N_v\times 1152}$. B and N represent batch size and token dimension, respectively. These features are concatenated along the channel dimension to form $f_v \in \mathbb{R}^{B\times N_v\times 2176}$, which is subsequently projected into the LLM's word embedding via a projection layer. HybridVLA(2.7B) uses only the CLIP [75] model as its vision encoder. When processing multi-view images, a shared vision encoder extracts features, which are then concatenated along the token dimension.

LLM. HybridVLA adopts 7B LLAMA-2 [91] as LLM, responsible for multi-modal understanding and reasoning. Language prompts are encoded into embedding space $f_l \in \mathbb{R}^{B\times N_l\times 4096}$ using the pre-trained tokenizer, then concatenated with visual tokens and input into LLM. The other specially designed LLM inputs are presented in the next section, and the output tokens are processed in two ways. First,

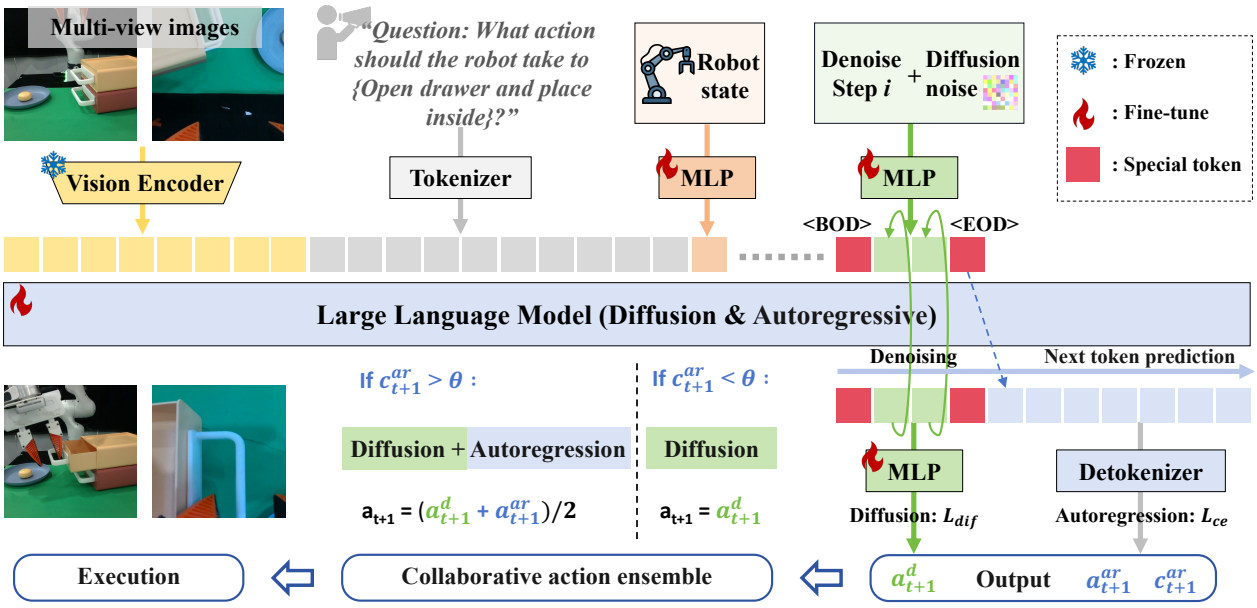


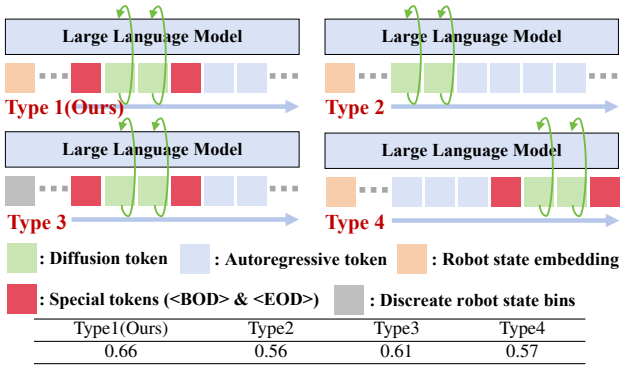
Figure 2. **HybridVLA Framework.** The input data, regardless of modality, is encoded and concatenated into our formatted token sequence. To integrate diffusion into the LLM, HybridVLA simultaneously projects the denoising timestep and noisy actions into the token sequence. The marker tokens, <BOD> (beginning of diffusion) and <EOD> (end of diffusion), are designed to bridge the two generation methods. By employing collaborative training to explicitly incorporate knowledge from both generation methods, these two action types reinforce each other and are adaptively ensembled to control the robot arms. For HybridVLA's output, continuous actions are generated through iterative denoising, while discrete actions are produced autoregressively, all within the next-token prediction process.

diffusion-based action (a^d_{t+1}) generation through a denoising process, where an MLP maps the tokens into the action space. Second, autoregressive-based action (a^{ar}_{t+1}) generation is performed through a detokenizer [45], which also computes the mean confidence (c^{ar}_{t+1}) of the action tokens, serving as a guiding factor for the collaborative action ensemble. For HybridVLA (2.7B), the workflow remains the same as that of HybridVLA (7B) but utilizes the 2.7B Phi-2 [37] as the LLM. In the next section, we introduce how to simultaneously equip a single LLM with diffusion and autoregressive action generation capabilities.

3.2. Collaborative Training Recipe

Directly combining diffusion and autoregressive pose prediction within a single LLM presents challenges such as instability and inconsistencies in next-token prediction. Therefore, we propose a collaborative training recipe that encompasses token sequence formulation design, hybrid objectives, and structured training stages.

Token sequence formulation design. As shown in the upper part of Figure 2, the input token sequence during training comprises not only visual and language tokens but also robot state, diffusion noise, and autoregressive action tokens. For the **robot state**, we integrate it into the LLM to enhance temporal consistency in action generation. Instead of discretizing the robot state and merging it with the language query [54] (Type 3 of Table 1), we employ a learnable MLP to map the robot state directly into the word embedding space, $f_r \in \mathbb{R}^{B\times1\times4096}$. The motivation is that diffusion action tokens are generated during the subsequent next-token prediction process, using all preceding tokens as conditions. Introducing discrete robot states could negatively impact the diffusion prediction of continuous actions. For **diffusion actions**, we predict them through a diffusion denoising process, conditioned on the previous tokens. The denoising step i and noisy actions a^i_t are projected into the LLM's word embeddings through an MLP, represented as continuous vectors. To seamlessly connect previous multi-modal tokens, diffusion tokens, and subsequent discrete tokens within a single sequence, we introduce special beginning-of-diffusion (<BOD>) and end-of-diffusion (<EOD>) tokens to encapsulate the diffusion tokens. This design not only clarifies the boundaries between the diffusion and autoregressive generation but also prevents confusion in the next-token prediction process, such as diffusion tokens directly predicting masked discrete tokens (Type 2 of Table 1). For **autoregressive actions**, we quantize the end-effector pose into discrete bins and replace part of the vocabulary in the LLM [45], which is then tokenized into a sequence of discrete tokens. Owing to the inherent nature of next-token prediction, both the question and the answer (discrete action GT) are fed into the LLM during training, while only the question is provided during inference. If the autoregressive prediction is placed before the diffusion to-



Type1(Ours)	Type2	Type3	Type4
0.66	0.56	0.61	0.57

Table 1. Exploring and validating our proposed token sequence formulation. The model is trained using both diffusion and autoregressive generation but tested only on diffusion-based actions (HybridVLA-dif) across 10 simulation tasks.

kens, it would lead to action GT leakage, serving as a condition for diffusion training (Type 4 of Table 1). Consequently, we place the diffusion token at the front explicitly providing continuous knowledge for the subsequent autoregressive generation.

Hybrid objectives. To simultaneously train diffusion and autoregressive action generation, we require two distinct loss functions. For diffusion part, following previous diffusion policies [13], we minimize the mean squared error between the predicted noise (ϵ_π) from the VLA model and the GT noise (ϵ). The loss function is defined as follows:

$$L_{dif} = E_{a,i,c}||\epsilon - \epsilon_\pi(a_t^i, i, c)||^2$$

Where $\epsilon \sim N(0,1)$ and c denotes the condition. Additionally, classifier-free guidance [28] is not used in order to ensure stable robot arm behavior [59]. For the autoregressive part, the cross-entropy loss (L_{ce}) is adopted to supervise the discrete output. With our designed token sequence formulation, the two losses can be seamlessly combined for collaborative penalization, defined as:

$$L_{hybrid} = L_{dif} + L_{ce}$$

Structured training stage. After loading the pretrained VLM parameters, HybridVLA undergoes two training stages with hybrid objectives: large-scale pretraining on open-source robotic data and fine-tuning on self-collected simulation and real-world data. During pretraining, we train HybridVLA for 5 epochs on 35 datasets from Open X-Embodiment [67], DROID [44], ROBOMIND [98], and others. The pretrain datasets contain 760k robot trajectories, comprising 33m frames. Due to dataset differences, pretraining relies solely on single 2D observations, whereas fine-tuning relies on either single or multi-view observations, depending on the downstream task. The details of the pretraining dataset are provided in Appendix A.1.

3.3. Collaborative Action Ensemble

During inference, given visual, language, and robot state inputs, HybridVLA concurrently generates actions using both diffusion and autoregressive methods, and then ensembles the actions to produce more stable execution.

Autoregressive actions. As shown in Figure 2, the autoregressive generation begins after the special token <EOD>. Similar to [45, 54], the generation of 7-DoF or 14-DoF actions closely mirrors the text generation process in an LLM. Unlike previous autoregressive VLA methods, HybridVLA's autoregressive generation additionally conditions on the inherently continuous nature of actions from diffusion tokens, outperforming independent autoregressive approaches, as demonstrated in the ablation study.

Diffusion actions. When generating diffusion actions, we append the special token <BOD> after the previous condition tokens to indicate that the model should perform the denoising process. We employ DDIM [87] sampling with n sampling steps. In HybridVLA, we find that integrating diffusion into the next-token prediction process not only improves action precision by fully leveraging the reasoning abilities and pretrained knowledge of VLMs but also mitigates performance degradation when reducing the number of inference denoising steps (e.g., $n = 4$), as demonstrated in the ablation study. To accelerate the sampling process, we introduce the KV cache before the diffusion tokens, forwarding conditional information, the denoising timestep, and pure noise only during the initial sampling step. In subsequent steps, the cached keys and values from the first pass are reused, while only the timestep and noise are iteratively forwarded. This strategy eliminates redundant computations and improves inference speed.

Ensembled actions. After obtaining the two types of actions under our collaborative training recipe, we empirically observe two phenomena. 1) Different action types demonstrate varying performance across tasks. Diffusion-based predictions excel in precise manipulation tasks, such as *Phone on base* and *Close laptop lid*, while autoregressive predictions perform better in tasks requiring scene semantic reasoning, such as *Water plants* and *Frame off hanger*. 2) The confidence of autoregressive action tokens serves as a reliable indicator of action quality. In over 80% of successfully completed test samples, the average confidence of autoregressive action tokens exceeds 0.96. Quantitative evaluations are provided in Appendix B.1 and B.2. Therefore, as shown at the end of Figure 2, we use the mean confidence of autoregressive tokens (c_{t+1}^{ar}) to guide the action ensemble. If the confidence exceeds θ ($\theta = 0.96$), we consider the autoregressive action (a_{t+1}^{ar}) sufficiently accurate and perform an average operation with the diffusion action (a_{t+1}^{d}). Otherwise, we rely solely on the diffusion action to control the robot. Additionally, to accelerate inference, HybridVLA-dif relies solely on diffusion-based action generation while

Models	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean S.R.↑	Infer. speed
ManipLLM (7B) [54]	0.50	0.80	0.40	0.20	0.80	0.35	0.10	0.25	0.15	0.20	0.38	2.2 Hz
OpenVLA (7B) [45]	0.65	0.40	0.75	0.60	0.80	0.20	0.35	0.15	0.10	0.10	0.41	6.3 Hz
π_0 (2.6B) [8]	0.90	0.60	**1.00**	0.30	0.90	0.25	0.35	**0.75**	0.05	0.45	0.55	13.8 Hz
CogACT (7B) [50]	0.80	0.85	0.90	0.65	0.90	**0.50**	**0.60**	0.35	0.25	0.25	0.60	9.8 Hz
HybridVLA-dif (7B)	0.85	0.75	**1.00**	0.80	0.95	**0.50**	0.50	0.30	**0.70**	0.25	0.66	9.4 Hz
HybridVLA (7B)	0.85	**0.95**	**1.00**	**0.90**	**1.00**	**0.50**	0.50	0.70	0.50	**0.50**	**0.74**	6.1 Hz
HybridVLA (2.7B)	**1.00**	0.80	0.90	0.80	0.90	0.25	0.20	0.45	0.25	0.25	0.58	12.3 Hz

Table 2. **Comparison of our proposed method and baselines on RLBench.** We train all methods in the Multi-task setting [85] and report the success rates (S.R.) for each task. The success condition follows the definition in RLBench. "HybridVLA-dif" refers to action prediction relying solely on the diffusion process. (7B), (2.7B), and (2.6B) refer to the sizes of the LLM used in the VLA model.

still collaboratively learning both action generation types during training to enhance mutual reinforcement.

4. Experiment

In Section 4.1, we compare the manipulation ability and inference speed of HybridVLA with previous VLA methods in simulation environments. The effectiveness of each component is validated in Section 4.2 and Appendix B. In Section 4.3, we present both quantitative and qualitative manipulation results of HybridVLA in real-world scenarios, including single-arm and dual-arm robot tasks. The generalization capabilities of HybridVLA are examined in Section 4.4, testing on unseen manipulated instances, background, spatial positions, and lighting conditions.

4.1. Simulation Experiment

Simulation benchmark. To systematically evaluate, we select the RLBench [34] benchmark in the CoppeliaSim simulator, which contains 10 different tabletop tasks. These tasks, performed using a Franka Panda robot and a front-view camera, include *Close box*, *Close Laptop*, *Toilet seat down*, *Sweep to dustpan*, *Close fridge*, *Phone on base*, *Take umbrella out*, *Frame off hanger*, *Wine at rack*, and *Water plants*. The data are collected using pre-defined waypoints and the Open Motion Planning Library [88]. Following the frame-sampling method used in previous works [22, 38, 85], we construct the training dataset, with each task consisting of 100 trajectories.

Training and Evaluation Details. We compare our method against four previous SOTA VLA models, including autoregressive-based approaches such as ManipLLM [54] and OpenVLA [45], as well as diffusion-based methods like π_0[8] and CogAct[50] with a DiT-base action head. Meanwhile, we categorize our method into three modes: HybridVLA (7B), HybridVLA (2.7B), and HybridVLA-dif (7B). All modes are jointly trained using our proposed collaborative training recipe; however, HybridVLA-dif relies solely on diffusion-based action generation during inference. To ensure a fair comparison, we load the official pretrained parameters provided by each method, adhering to their respective training settings. For HybridVLA, the single-view RGB input is resized to 224×224, and the robot state is consistent with predicted actions (7-DOF end-effector poses). During training, we use the AdamW optimizer with a constant learning rate of 2e-5. As shown in Figure 2, we only update the LLM and injected MLP parameters. Our models are trained for 300 epochs on 8 NVIDIA A800 GPUs with mixed-precision training. For evaluation, following [45, 50], all methods are tested with 20 rollouts from the latest epoch.

Quantitative Results. As shown in Table 2, HybridVLA(7B) achieves an average success rate of 74% across 10 distinct tasks, outperforming the previous SOTA autoregressive-based VLA (OpenVLA) and diffusion-based VLA (CogACT) by 33% and 14%, respectively. These results demonstrate that our method seamlessly introduces diffusion modeling within the LLM and effectively combines the advantages of two types of pose prediction. Remarkably, compared to CogACT and π_0, HybridVLA-dif also achieves performance improvements of 6% and 11%, respectively. These results highlight that, unlike previous approaches that attach the diffusion head after the VLM, our method effectively leverages the inherent capabilities of the VLM to unlock the full potential of diffusion-based action generation within the VLA model. Finally, HybridVLA(2.7B) delivers satisfactory results, confirming our method's effectiveness in enhancing VLM manipulation capabilities across different model sizes.

Inference Speed. By harnessing VLM's reasoning capabilities and conducting large-scale robotic pretraining, our method reduces denoising steps to 4 without any performance degradation, as shown in Figure 3. In Table 2, when tested on an NVIDIA 4090D GPU, HybridVLA-dif (7B) and HybridVLA (2.7B) achieve satisfactory frequencies comparable to CogACT (7B) and π_0 (2.6B), thanks to the reduced denoising steps and efficient KV cache of our method. All models are loaded with bfloat16 precision for inference. Notably, while KV cache has been used in previous autoregressive VLA methods [45, 54], we first integrate it into the LLM's diffusion-based action prediction.

4.2. Ablation Study

We perform ablation experiments on the 10 RLBench tasks and compute the average manipulation accuracy. **The im-**

	AR	Dif	L_{Hybrid}	LSP	RSE	CTR	CAE	**Mean↑**
Ex0	✓	✓	✓	✓	✓	✓	✓	0.74
Ex1	-	✓	✓	✓	✓	✓	-	0.66
Ex2	-	✓	-	✓	✓	✓	-	0.60
Ex3	✓	-	✓	✓	✓	✓	-	0.62
Ex4	✓	-	-	✓	✓	✓	-	0.57
Ex5	✓	✓	✓	-	✓	✓	✓	0.22
Ex6	✓	✓	✓	-	-	✓	✓	0.10
Ex7	-	✓	✓	✓	✓	-	-	0.57

Table 3. **Impact of each component.** AR and Dif represent autoregressive and diffusion-based action generation, respectively, using LLM as backbone. L_{Hybrid} means jointly training the model by Hybrid objectives. LSP denotes large-scale pretraining on assembled robotic datasets, while RSE refers to our injected robot state embedding. CTR and CAE mean our proposed collaborative training recipe and collaborative action ensemble method.

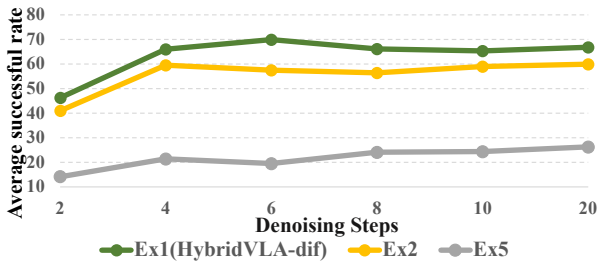


Figure 3. **The impact of denoising steps**, where the x-axis and y-axis represent the denoising steps and manipulation success rate.

pact of each component. As shown in Table 3, a comparison between Ex1 and Ex2, as well as Ex3 and Ex4, demonstrates that under our proposed collaborative training of diffusion and autoregressive action generation, HybridVLA-dif (Ex1) and HybridVLA-ar (Ex2) outperform their individually trained counterparts. These results further confirm that our method effectively integrates diffusion's continuous prediction capability with the reasoning ability of the autoregressive approach, enabling mutual reinforcement. Compared to Ex0, Ex5 and Ex6 highlight the essential role of large-scale pretraining and robot state injection in ensuring stable prediction. A comparison of Ex7 and Ex1 shows that simply combining diffusion and autoregressive approaches in a single LLM without our training recipe fails to improve accuracy. Moreover, Ex7 underperforms compared to training diffusion alone without hybrid loss (Ex2). The various token formulation designs used in our training recipe are explored in Table 1. **The impact of denoising steps.** In Figure 3, we explore the relationship between the manipulation performance and different denoising steps, comparing Ex1 (HybridVLA-dif), Ex2, and Ex5. The results suggest that integrating the diffusion process into the LLM and leveraging large-scale robotic pretraining allows for a reduction in inference denoising steps from 20 to 4 without causing performance degradation. Conversely, Ex5 (without large-scale pretraining) exhibits a significant decline in manipulation accuracy. To balance inference speed and accuracy, we set the diffusion denoising steps to 4. In Appendix B.2, we provide additional ablation studies on (1) confidence thresholds in the collaborative action ensemble and (2) KV cache influence on inference speed.

4.3. Real-World Experiment

Self-collected Data. For single-arm tasks, we use a Franka Research 3 robot with a static front-view and a wrist-view camera. We perform 5 tasks: 1) *Pick and place*, 2) *Unplug charger*, 3) *Open drawer and place inside*, 4) *Pour water*, 5) *Wipe blackboard*. For each task, 100 demonstrations are collected via teleoperation using a SpaceMouse from various positions on the table. For dual-arm tasks, we use an AgileX dual-arm robot equipped with a static exterior view, a right-wrist view, and a left-wrist view camera. We conduct 5 coordinated dual-arm tasks: 1) *Pick and place*, 2) *Lift ball and place*, 3) *place two bottles at rack*, 4) *Wipe blackboard*, 5) *Fold shorts*. Similarly, 100 demonstrations are collected for each task using master-puppet teleoperation. Additional details are provided in Appendix A.2.

Training and Evaluation Details. For a fair comparison, we evaluate only the diffusion-generated actions of our method (HybridVLA-dif) against diffusion-based VLA methods, π_0[8] and CogAct[50]. The implementation details remain consistent with our simulation experiments, except for incorporating two-view inputs for single-arm tasks and three-view inputs for dual-arm tasks. For evaluation, we use the checkpoint from the latest epoch to perform 20 rollouts across diverse tabletop positions.

Quantitative Results. In Table 4, HybridVLA-dif achieves outstanding performance across single-arm real-world tasks. For *Pick and place* and *Unplug charger*, our method achieves success rates of 85% and 95%, respectively, demonstrating accurate object position prediction. For *Pour Water*, HybridVLA-dif outperforms the previous SOTA method by 30%, demonstrating its ability to comprehend object relationships and predict precise rotations. The superior performance in *Wipe Blackboard* and *Open Drawer and Place Inside* further underscores HybridVLA-dif's stability in long-horizon tasks. For dual-arm tasks, we extend the action dimension of diffusion tokens to 14-DOF, representing the 7-DOF end-effector poses for both the right and left arms. Our method consistently surpasses previous VLA method across 5 distinct tasks, highlighting HybridVLA's ability to leverage VLMs' reasoning capabilities for object affordance understanding and dual-arm coordination.

Qualitative Results. In Figure 4, we present visualizations of the manipulation process performed by our method. HybridVLA-dif accurately predicts control actions across various task demands, including precise positioning and ro-

Models	Franka single-arm robot						AgileX dual-arm robot					
	Pick and place	Unplug charger	Pour water	Wipe blackboard	Open drawer and place inside	Mean. S.R. ↑	Pick and place	Lift ball and place	Place bottles at rack	Wipe blackboard	Fold shorts	Mean. S.R. ↑
π_0 (2.6B) [8]	0.50	0.35	0.45	0.35	**0.60**	0.45	0.75	0.65	0.40	0.30	0.65	0.55
CogACT (7B) [50]	0.80	0.70	0.40	0.65	0.50	0.61	-	-	-	-	-	-
HybridVLA-dif(7B)	**0.85**	**0.95**	**0.75**	**0.85**	**0.60**	**0.80**	**0.80**	**0.75**	**0.60**	**0.45**	**0.70**	**0.66**

Table 4. **Comparison of our method and baselines in real-world scenarios.** We train all methods in a single-task setting [105] and report the success rates. Success is determined by human evaluation based on whether the task is completed. Since CogAct lacks support for multi-view images, which are crucial for dual-arm tasks [8, 19], we conduct our dual-arm comparison solely with π_0.

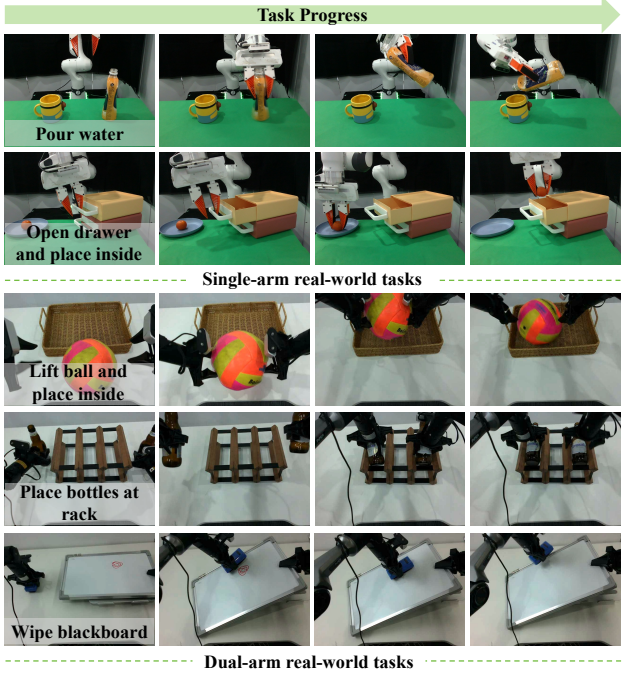
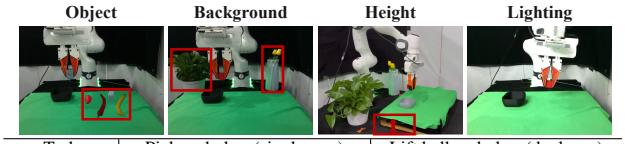


Figure 4. The qualitative results of HybridVLA-dif in real-world tasks. More visualizations are shown in Appendix C.



Task	Pick and place(single arm)		Lift ball and place(dual arm)	
Scenario	HybridVLA-dif	Cogact	HybridVLA-dif	π_0
Original	0.85	0.80	0.75	0.65
Object	0.50(-41%)	0.45(-43%)	0.70(-7%)	0.60(-8%)
Background	0.60(-31%)	0.50(-37%)	0.60(-20%)	0.50(-23%)
Height	0.70(-17%)	0.50(-37%)	0.55(-20%)	0.45(-31%)
Lightning	0.70(-17%)	0.60(-25%)	0.65(-13%)	0.55(-15%)

Table 5. **Generalization.** "Object", "Background", "Height", and "Lighting" denote unseen manipulated objects, backgrounds, spatial positions, and lighting conditions, respectively. The image above illustrates the four unseen test scenarios, with red boxes highlighting the key differences.

tation, dual-arm coordination, and scene understanding. For instance, in the *Wipe blackboard* dual-arm task, our method enables the left arm to precisely grasp the eraser while the right arm holds the edge of the blackboard. It then coordinates the movement and rotation of both objects to establish a relative spatial relationship. Finally, the left arm controls the eraser to precisely remove the red circle. Additional failure case analyses are provided in Appendix D, and execution videos are available in the supplementary materials.

4.4. Generalization Experiment

Beyond achieving stable manipulation across diverse tasks, HybridVLA also exhibits strong generalization capabilities in real-world scenarios. Since CogAct and π_0 excel in single-arm and dual-arm tasks, respectively, we design four common generalization experiments, comparing our method with CogAct on the single-arm *Pick and place* task and with π_0 on the dual-arm *Lift ball and place* task. **1) Unseen manipulated objects.** In this scenario, we replace the training manipulated objects with a series of unseen objects, e.g., replacing the red block with a charger. As shown in the "Object" row of Table 5, our method demonstrates the smallest accuracy drop. These results indicate that HybridVLA effectively incorporates diffusion into next-token prediction, leveraging VLM pretraining knowledge to reason about diverse object semantics based on observation and language context. **2) Unseen background.** In this scenario, cluttered backgrounds are introduced during testing, such as adding unseen flowers around the manipulated object. As shown in the "Background" row of Table 5, HybridVLA exhibits minimal accuracy degradation. These results demonstrate that our collaborative training recipe effectively absorbs the strengths of both action generation methods, enhancing robustness to environmental variations. **3) Unseen Spatial position.** Unlike position shifts within the same plane, we introduce height variations during testing, further challenging the model's spatial comprehension. As shown in the "Height" row of Table 5, HybridVLA consistently achieves precise manipulation even when encountering objects in previously unseen spatial positions. These results highlight that HybridVLA not only excels in observational generalization but also exhibits strong trajectory generalization capabilities. **4) Unseen lighting conditions.** Finally, we introduce variations in lighting conditions, a common challenge in real-world environments. As shown in the last row of Table 5, all methods maintain satisfactory performance. This can be attributed to large-scale pretraining on robotic datasets, which significantly enhances these models' ability to generalize across diverse data distributions.

5. Conclusion and Limitation

In this paper, we introduce HybridVLA, a unified framework that equips a single LLM with both diffusion and autoregressive action prediction capabilities. Our proposed collaborative training recipe bridges the gap between these two action generation approaches, enabling mutual reinforcement and enhancing manipulation robustness. By effectively integrating the inherently continuous nature of actions from diffusion with the reasoning capabilities of autoregressive methods, HybridVLA achieves outstanding performance and strong generalization across both simulation and real-world tasks. One limitation of HybridVLA is that its inference time is constrained by the slower autoregressive generation, reducing control frequency, similar to previous autoregressive-based VLA methods [10, 45, 54]. However, our collaborative training enables mutual reinforcement between the two generation methods, allowing inference with only the diffusion process (HybridVLA-dif) and achieving a 9.4 Hz inference speed.

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. 2, 3

[2] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022. 3

[3] 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*, 35: 23716–23736, 2022. 1, 3

[4] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 2

[5] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 1

[6] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Hydra: Hybrid robot actions for imitation learning. *arxiv*, 2023. 14

[7] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024. 2

[8] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 2, 3, 6, 7, 8

[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. 3, 14

[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, 3, 9

[11] Federico Ceola, Lorenzo Natale, Niko Sünderhauf, and Krishan Rana. Lhmanip: A dataset for long-horizon language-grounded manipulation tasks in cluttered tabletop environments. *arXiv preprint arXiv:2312.12036*, 2023. 14

[12] Lawrence Yunliang Chen, Simeon Adebola, and Ken Goldberg. Berkeley UR5 demonstration dataset. https://sites.google.com/view/berkeley-ur5/home. 14

[13] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 2, 3, 5

[14] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024. 14

[15] Zichen Jeff Cui, Yibin Wang, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. From play to policy: Conditional behavior generation from uncurated robot data. *arXiv preprint arXiv:2210.10047*, 2022. 14

[16] Shivin Dass, Jullian Yapeter, Jesse Zhang, Jiahui Zhang, Karl Pertsch, Stefanos Nikolaidis, and Joseph J. Lim. CLVR jaco play dataset, 2023. 14

[17] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 2, 3

[18] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021. 14

[19] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024. 3, 8

[20] Peng Gao*, Jiaming Han*, Renrui Zhang*, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He,

Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 1

[21] Yiran Geng, Boshi An, Haoran Geng, Yuanpei Chen, Yaodong Yang, and Hao Dong. End-to-end affordance learning for robotic manipulation. In *International Conference on Robotics and Automation (ICRA)*, 2023. 2

[22] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023. 6

[23] Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012. 2, 3

[24] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills, 2023. 14

[25] Ziyu Guo*, Renrui Zhang*#, Xiangyang Zhu, Chengzhuo Tong, Peng Gao, Chunyuan Li, and Pheng-Ann Heng. Sam2point: Segment any 3d as videos in zero-shot and promptable manners. *arXiv preprint arXiv:2408.16768*, 2024. 3

[26] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. Can we generate images with cot? let's verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*, 2025. 2

[27] Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In *Robotics: Science and Systems*, 2023. 14

[28] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3

[30] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. 2, 3

[31] Siyuan Huang, Iaroslav Ponomarenko, Zhengkai Jiang, Xiaoqi Li, Xiaobin Hu, Peng Gao, Hongsheng Li, and Hao Dong. Manipvqa: Injecting robotic affordance and physically grounded information into multi-modal large language models. *arXiv preprint arXiv:2403.11289*, 2024. 2, 3

[32] 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. 2, 3

[33] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. *arXiv preprint arXiv:2409.01652*, 2024. 3

[34] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 2, 6

[35] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022. 14

[36] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 3

[37] Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3, 2023. 4

[38] Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilue Wang, Longzan Luo, Lily Lee, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, et al. Lift3d foundation policy: Lifting 2d large-scale pretrained models for robust 3d robotic manipulation. *arXiv preprint arXiv:2411.18623*, 2024. 6

[39] Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanwei Li, Yu Qi, Xinyan Chen, Liuhui Wang, Jianhan Jin, Claire Guo, Shen Yan, et al. Mme-cot: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency. *arXiv preprint arXiv:2502.09621*, 2025. 3

[40] Shirin Joshi, Sulabh Kumra, and Ferat Sahin. Robotic grasping using deep reinforcement learning. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 1461–1466. IEEE, 2020. 2

[41] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. QT-Opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018. 14

[42] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. *arXiv preprint arXiv:2402.07865*, 2024. 2, 3

[43] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024. 2, 3

[44] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe,

Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024. 2, 5, 14

[45] 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. 2, 3, 4, 5, 6, 9, 14, 16

[46] Vikash Kumar, Rutav Shah, Gaoyue Zhou, Vincent Moens, Vittorio Caggiano, Abhishek Gupta, and Aravind Rajeswaran. Robohive: A unified framework for robot learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 14

[47] Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks, 2019. 14

[48] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *ICLR 2025 Spotlight*, 2024. 3

[49] 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*, pages 19730–19742. PMLR, 2023. 1, 3

[50] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024. 2, 3, 6, 7, 8

[51] Xiaoqi Li, Lingyun Xu, Jiaming Liu, Mingxu Zhang, Jiahui Xu, Siyuan Huang, Iaroslav Ponomarenko, Yan Shen, Shanghang Zhang, and Hao Dong. Crayonrobo: Toward generic robot manipulation via crayon visual prompting. 3

[52] 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, 3

[53] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024. 2

[54] Xiaoqi Li, Mingxu Zhang, Yiran Geng, Haoran Geng, Yuxing Long, Yan Shen, Renrui Zhang, Jiaming Liu, and Hao Dong. Manipllm: Embodied multimodal large language model for object-centric robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18061–18070, 2024. 2, 3, 4, 5, 6, 9, 16

[55] Huihan Liu, Soroush Nasiriany, Lance Zhang, Zhiyao Bao, and Yuke Zhu. Robot learning on the job: Human-in-the-loop autonomy and learning during deployment. In *Robotics: Science and Systems (RSS)*, 2023. 14

[56] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1, 3

[57] Jiaming Liu, Chenxuan Li, Guanqun Wang, Lily Lee, Kaichen Zhou, Sixiang Chen, Chuyan Xiong, Jiaxin Ge, Renrui Zhang, and Shanghang Zhang. Self-corrected multimodal large language model for end-to-end robot manipulation. *arXiv preprint arXiv:2405.17418*, 2024. 3

[58] Jiaming Liu, Mengzhen Liu, Zhenyu Wang, Lily Lee, Kaichen Zhou, Pengju An, Senqiao Yang, Renrui Zhang, Yandong Guo, and Shanghang Zhang. Robomamba: Multimodal state space model for efficient robot reasoning and manipulation. *arXiv preprint arXiv:2406.04339*, 2024. 2, 3

[59] 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. 3, 5

[60] Jianlan Luo, Charles Xu, Xinyang Geng, Gilbert Feng, Kuan Fang, Liam Tan, Stefan Schaal, and Sergey Levine. Multi-stage cable routing through hierarchical imitation learning. *arXiv preprint arXiv:2307.08927*, 2023. 14

[61] Jianlan Luo, Charles Xu, Fangchen Liu, Liam Tan, Zipeng Lin, Jeffrey Wu, Pieter Abbeel, and Sergey Levine. Fmb: a functional manipulation benchmark for generalizable robotic learning. *arXiv preprint arXiv:2401.08553*, 2024. 14

[62] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023. 14

[63] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. RoboTurk: A crowdsourcing platform for robotic skill learning through imitation. *CoRR*, abs/1811.02790, 2018. 14

[64] Oier Mees, Jessica Borja-Diaz, and Wolfram Burgard. Grounding language with visual affordances over unstructured data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023. 14

[65] Russell Mendonca, Shikhar Bahl, and Deepak Pathak. Structured world models from human videos. *CoRL*, 2023. 14

[66] Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. In *Conference on Robot Learning (CoRL)*, 2022. 14

[67] 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. 2, 3, 5

[68] 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. 3

[69] Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. In *The Eleventh International Conference on Learning Representations*. 2

[70] Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann, et al. Imitating human behaviour with diffusion models. *arXiv preprint arXiv:2301.10677*, 2023. 3

[71] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 3

[72] 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. 2, 3

[73] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. *arXiv preprint arXiv:2405.07503*, 2024. 3

[74] Gabriel Quere, Annette Hagengruber, Maged Iskandar, Samuel Bustamante, Daniel Leidner, Freek Stulp, and Joern Vogel. Shared Control Templates for Assistive Robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, page 7, Paris, France, 2020. 14

[75] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3

[76] Ilija Radosavovic, Tete Xiao, Stephen James, Pieter Abbeel, Jitendra Malik, and Trevor Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023. 14

[77] Moritz Reuss, Maximilian Li, Xiaogang Jia, and Rudolf Lioutikov. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*, 2023. 2, 3

[78] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3

[79] Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task agnostic offline reinforcement learning. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022. 14

[80] Kallol Saha, Vishal Mandadi, Jayaram Reddy, Ajit Srikanth, Aditya Agarwal, Bipasha Sen, Arun Singh, and Madhava Krishna. Edmp: Ensemble-of-costs-guided diffusion for motion planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10351–10358. IEEE, 2024. 3

[81] Saumya Saxena, Mohit Sharma, and Oliver Kroemer. Multi-resolution sensing for real-time control with vision-language models. In *7th Annual Conference on Robot Learning*, 2023. 14

[82] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023. 14

[83] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. MUTEX: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023. 14

[84] Haochen Shi, Huazhe Xu, Samuel Clarke, Yunzhu Li, and Jiajun Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools, 2023. 14

[85] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022. 6

[86] Anthony Simeonov, Ankit Goyal, Lucas Manuelli, Lin Yen-Chen, Alina Sarmiento, Alberto Rodriguez, Pulkit Agrawal, and Dieter Fox. Shelving, stacking, hanging: Relational pose diffusion for multi-modal rearrangement. *arXiv preprint arXiv:2307.04751*, 2023. 3

[87] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 5

[88] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, 2012. 6

[89] 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. 3

[90] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. *arXiv preprint arXiv:1805.01954*, 2018. 3

[91] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar,

et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[92] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5923–5930. IEEE, 2023. 3

[93] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale, 2023. 14

[94] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022. 3

[95] 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. 2, 3

[96] 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. 3

[97] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023. 2, 3

[98] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024. 2, 5, 14

[99] Tianhao Wu, Mingdong Wu, Jiyao Zhang, Yunchong Gan, and Hao Dong. Learning score-based grasping primitive for human-assisting dexterous grasping. *Advances in Neural Information Processing Systems*, 36:22132–22150, 2023. 3

[100] Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. Chaineddiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *7th Annual Conference on Robot Learning*, 2023. 2, 3

[101] Chuyan Xiong, Chengyu Shen, Xiaoqi Li, Kaichen Zhou, Jiaming Liu, Ruiping Wang, and Hao Dong. Autonomous interactive correction mllm for robust robotic manipulation. In *8th Annual Conference on Robot Learning*, 2024. 3

[102] Ran Xu, Yan Shen, Xiaoqi Li, Ruihai Wu, and Hao Dong. Naturalvlm: Leveraging fine-grained natural language for affordance-guided visual manipulation. *arXiv preprint arXiv:2403.08355*, 2024. 3

[103] Ge Yan, Kris Wu, and Xiaolong Wang. ucsd kitchens Dataset. 2023. 14

[104] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021. 2

[105] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy. *arXiv preprint arXiv:2403.03954*, 2024. 2, 8

[106] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 3

[107] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023. 1

[108] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? *ECCV 2024*, 2024.

[109] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, et al. Mavis: Mathematical visual instruction tuning with an automatic data engine. *arXiv preprint arXiv:2407.08739*, 2024. 1

[110] Gaoyue Zhou, Victoria Dean, Mohan Kumar Srirama, Aravind Rajeswaran, Jyothish Pari, Kyle Hatch, Aryan Jain, Tianhe Yu, Pieter Abbeel, Lerrel Pinto, Chelsea Finn, and Abhinav Gupta. Train offline, test online: A real robot learning benchmark, 2023. 14

[111] Xinghao Zhu, Ran Tian, Chenfeng Xu, Mingyu Ding, Wei Zhan, and Masayoshi Tomizuka. Fanuc manipulation: A dataset for learning-based manipulation with fanuc mate 200id robot. 2023. 14

[112] Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022. 14

[113] Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors, 2023. 14

Appendix A. We begin by detailing the large-scale pretraining and self-collected real-world datasets.
Appendix B. Additional simulation experiments and ablation studies are presented.
Appendix C. We include further visualizations of both single-arm and dual-arm manipulation processes.
Appendix D. An analysis of failure cases encountered when using HybridVLA to control a robot.

A. Additional Dataset Details

A.1. Large-scale Pretraining Dataset

Our pre-training dataset collection comprises 35 datasets, encompassing a total of 760k trajectories and 33m frames. Table 6 provides a comprehensive list of our pre-training datasets along with their respective sampling weights. The number of trajectories and the sampling weights can be automatically adjusted during dataset assembly. Following the prior data preprocessing approach [45], we reformulate the pre-training datasets to emphasize end-effector sequence control, ensuring alignment with the specific requirements of our model training. Due to inherent differences among datasets, only single 2D observations are used during pre-training. However, during fine-tuning, HybridVLA can accommodate both single- and multi-view observations depending on the task requirements. For instance, AgileX dual-arm robot tasks require three viewpoints—an ego view and two wrist camera views—to capture a comprehensive observation of the object while mitigating occlusions caused by the robot arm. HybridVLA processes multi-view images using a shared vision encode and then concatenates the visual feature along the token dimension. Notably, the difference in the number of images used during pre-training and fine-tuning does not impact manipulation performance in downstream tasks.

Training Dataset Mixture	
Fractal [9]	9.1%
Kuka [41]	27.8%
Bridge[18, 93]	4.1%
Taco Play [64, 79]	2.1%
Jaco Play [16]	0.3%
Berkeley Cable Routing [60]	0.2%
Roboturk [63]	1.7%
Viola [113]	0.7%
Berkeley Autolab UR5 [12]	0.9%
Toto [110]	1.5%
Language Table [62]	3.1%
Stanford Hydra Dataset [6]	3.2%
Austin Buds Dataset [112]	0.2%
NYU Franka Play Dataset [15]	0.6%
Furniture Bench Dataset [27]	1.8%
UCSD Kitchen Dataset [103]	<0.1%
Austin Sailor Dataset [66]	1.6%
Austin Sirius Dataset [55]	1.2%
DLR EDAN Shared Control [74]	<0.1%
IAMLab CMU Pickup Insert [81]	0.7%
UTAustin Mutex [83]	1.6%
Berkeley Fanuc Manipulation [111]	0.6%
CMU Stretch [65]	0.1%
BC-Z [35]	5.4%
FMB Dataset [61]	5.0%
DobbE [82]	1.0%
DROID [44]	7.2%
Stanford Kuka Dataset [47]	0.1%
Stanford Robocook Dataset [84]	0.1%
Maniskill [24]	6.3%
Berkeley RPT [76]	0.1%
QUT Dexterous Manipulation [11]	0.1%
RoboSet [46]	1.8%
BridgeData V2 [93]	4.7%
RoboMind [98]	5.2%

Table 6. The dataset name and sampling weight used in our mixed large-scale pretraining dataset.

A.2. Self-collected Real-world Dataset

The experimental assets and environments for the single-arm and dual-arm setups are shown in Figure 5 (a) and (b), respectively. For the single-arm setup, a 3D-printed UMI gripper [14] is attached to the Franka robot and is used across all baselines. We utilize RealSense 435 and RealSense 515 cameras to capture both wrist and front views. For the dual-arm setup, two Orbbec DABAI cameras are used to capture the left and right wrist views, while a RealSense 515 is mounted overhead to capture a static third-person view. We provide a detailed explanation of the real-world tasks and their success conditions. We begin by describing the single-arm tasks:

1. Pick and place. This task requires the robot to pick up a specifically colored block based on a language description and place it in a specifically colored bowl.

2. Unplug charger. The robot needs to grasp the charger at an optimal position and rotation, and then lift it to a certain height without slipping.

3. Pour water. The robot needs to first pick the bottle, then rotate it to a position slightly above the cup, and tilt it to perform the pouring action. The task is deemed successful only if the bottle opening is correctly aligned with the cup.

4. Wipe blackboard. The robot needs to first grasp an eraser and then use it to remove the red markings from a blackboard placed on the tabletop. The red markings are drawn on an unfixed region, and the task is considered successful only if they are completely erased.

5. Open drawer and place inside. The robot needs to

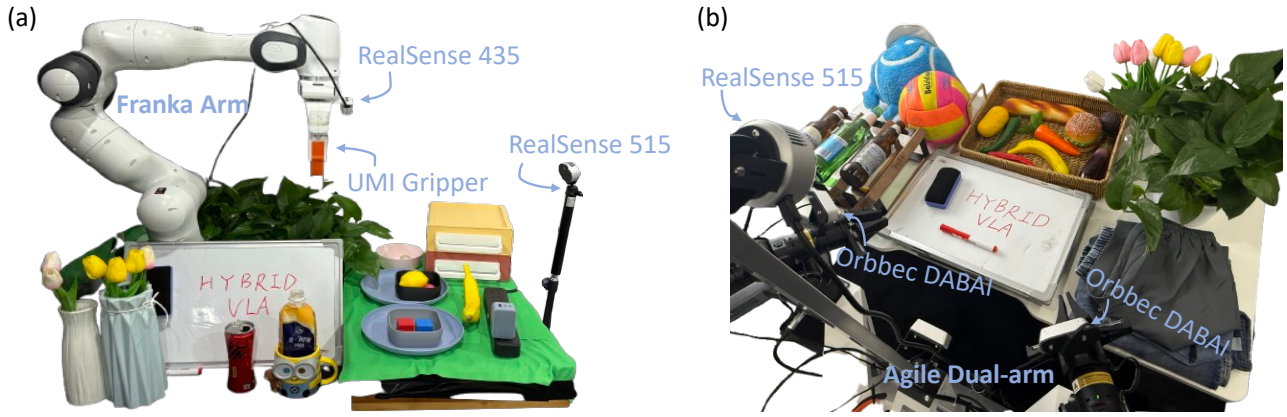


Figure 5. **Real-World Assets and Experimental Settings.** We provide visualizations of the assets used and the experimental settings for single-arm FR3 robot tasks and dual-arm AgileX robot tasks, respectively.

Models	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean. S.R. ↑
HybridVLA-ar(7B)	**0.85**	0.70	0.90	0.85	0.95	0.30	0.25	0.40	0.45	**0.50**	0.62
HybridVLA-dif(7B)	**0.85**	0.75	**1.0**	0.80	0.95	**0.50**	**0.50**	0.30	**0.70**	0.25	0.66
HybridVLA(7B)	**0.85**	**0.95**	**1.0**	**0.90**	**1.0**	**0.50**	**0.50**	**0.70**	0.50	**0.50**	**0.74**

Table 7. **Detailed Simulation Experiments.** We validate that different action types within our proposed framework exhibit varying performance across tasks. All models undergo joint training using our proposed collaborative training recipe; however, HybridVLA-ar and HybridVLA-dif rely exclusively on autoregressive-based and diffusion-based action generation during inference, respectively. Underlining indicates the highest score between HybridVLA-ar and HybridVLA-dif.

Threshold	0.90	0.92	0.94	0.96	0.98
Success rate	0.66	0.64	0.70	0.74	0.69

Table 8. **Ablation Study.** We explore the impact of different confidence thresholds on the performance of ensemble actions.

open the top drawer, pick up the required objects based on the language description, place them in the opened drawer, and then close it. This task consists of four sequential sub-tasks: *open drawer*, *pick object*, *place object*, and *close drawer*. The task is considered complete once all sub-tasks have been successfully executed.

We then describe the details of dual-arm tasks:

1. Pick and place. The robot must use both its left and right arms to pick up two objects based on the language description and place them in the container.

2. Lift ball and place. Both the left and right arms must simultaneously make contact with the ball, which is secured between the two grippers. The arms coordinate their movements to transport the ball to the container while ensuring it does not slip. This task highly tests the model's dual-arm coordination capabilities.

3. Place bottles at rack. The left and right robot arms need to grasp the bottles placed on their respective sides and rotate them to position them parallel to the rack.

4. Wipe blackboard. Unlike the single-arm setting, the dual-arm setting requires one arm to hold the whiteboard while the other picks up the eraser and wipes off the red marker.

5. Fold shorts: This task requires folding a pair of shorts, involving two sequential steps. First, one pant leg is folded over the other to align them. Then, the pants are folded in half from top to bottom. Throughout the process, both arms must coordinate their movements. For example, in the first step, the left arm holds the bottom of the pant leg while the right arm grips the upper part, working together to complete the folding.

B. Additional Quantitative Results

B.1. Additional Simulation Experiments

In Table 7, we validate the first observed phenomenon mentioned in Section 3.3: different action types within our proposed framework exhibit varying performance across tasks. Meanwhile, we categorize our method into three modes: HybridVLA (7B), HybridVLA-ar (7B), and HybridVLA-dif (7B). All modes undergo joint training using our proposed collaborative training recipe; however, HybridVLA-ar and HybridVLA-dif rely exclusively on autoregressive-

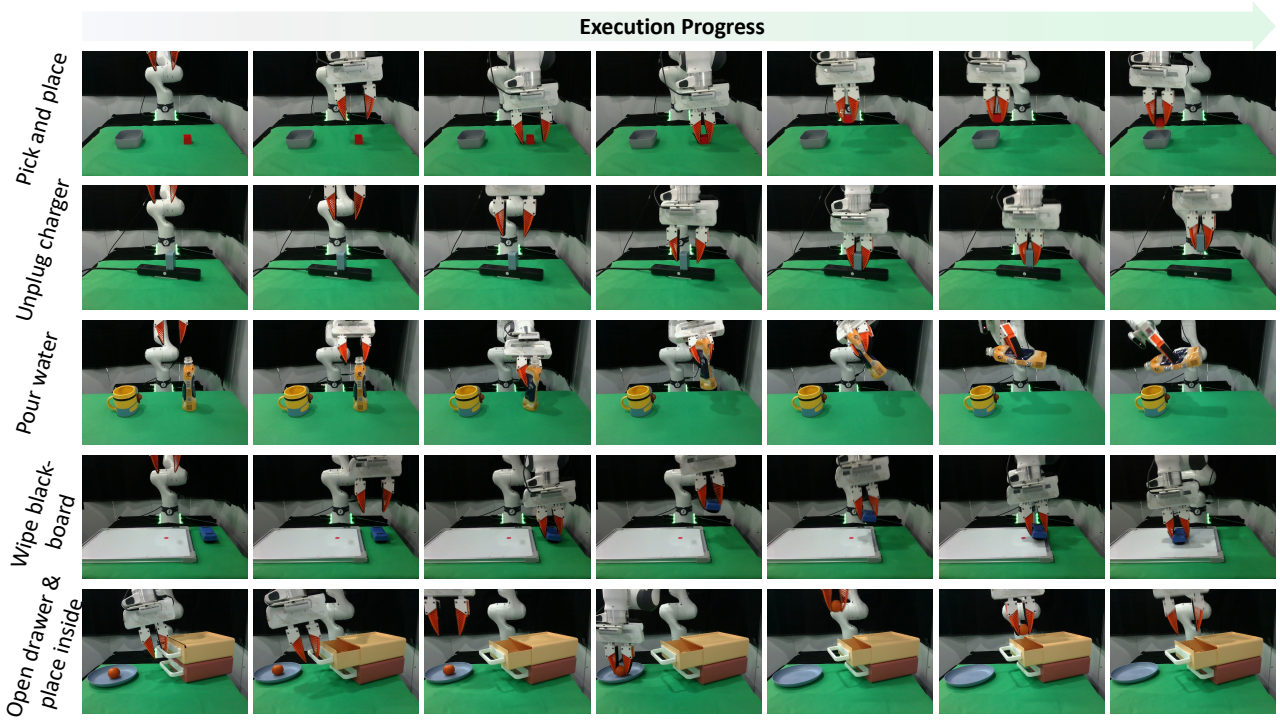


Figure 6. **Single-arm Execution Visualization**. We visualize key frames of the agent's execution process from the front perspective.

based and diffusion-based action generation during inference, respectively. The experiments are conducted in the RLBench simulator across 10 tasks, and evaluated based on success rate. Comparing HybridVLA-ar and HybridVLA-dif, HybridVLA-ar outperforms in 4 out of 10 tasks, while HybridVLA-dif leads in the remaining 6 tasks. These results validate our findings that, within the HybridVLA framework, diffusion-based predictions excel in precise manipulation tasks, such as *Phone on base*, *Toilet seat down*, and *Close laptop lid*, whereas autoregressive predictions perform better in tasks requiring scene-level semantic reasoning, such as *Sweep to dustpan*, *Water plants*, and *Frame off hanger*. Therefore, while collaborative training allows diffusion-based and autoregressive-based action generation to reinforce each other, assembling both methods results in more robust actions.

B.2. Additional Ablation Study

The impact of confidence threshold in collaborative action ensemble. The proposed collaborative ensemble strategy determines whether to use the action predicted by diffusion alone or the averaged output of both diffusion and autoregressive methods, guided by a mean confidence threshold derived from the autoregressive action token. In this experiment, we investigate the optimal confidence threshold required to ensure the accuracy of autoregressive actions and enhance the overall precision of the ensemble-generated action. Specifically, as shown in Tab. 8, we vary the threshold from 0.90 to 0.98. We find that when the confidence threshold drops below 0.94, autoregressive predictions become unreliable, leading to a slight degradation in the performance of the ensemble action. Conversely, when the threshold reaches 0.98, the number of valid autoregressive actions becomes too limited, causing the performance of the ensemble action to closely match that of the diffusion-predicted action. Empirically, we conclude that setting the threshold to 0.96 ensures a stable action ensemble.

The impact of KV cache in inference speed. As described in Section 3.3, we adopt the KV cache to eliminate redundant computations and enhance inference speed. In this experiment, we examine the extent to which this mechanism accelerates inference. With the KV cache enabled (Table 2 of the main paper), HybridVLA-dif achieves an average success rate of 66% across 10 simulation tasks with an inference speed of 9.4 Hz. Removing it results in a similar average success rate but reduces the inference speed to 5.0 Hz. While the KV cache has typically been used in previous autoregressive VLA methods [45, 54], we are the first to integrate it into an LLM's diffusion-based action generation.

C. Additional Visualizations

Figure 6 and Figure 7 illustrate keyframes of single-arm and dual-arm real-world execution processes. Notably, our

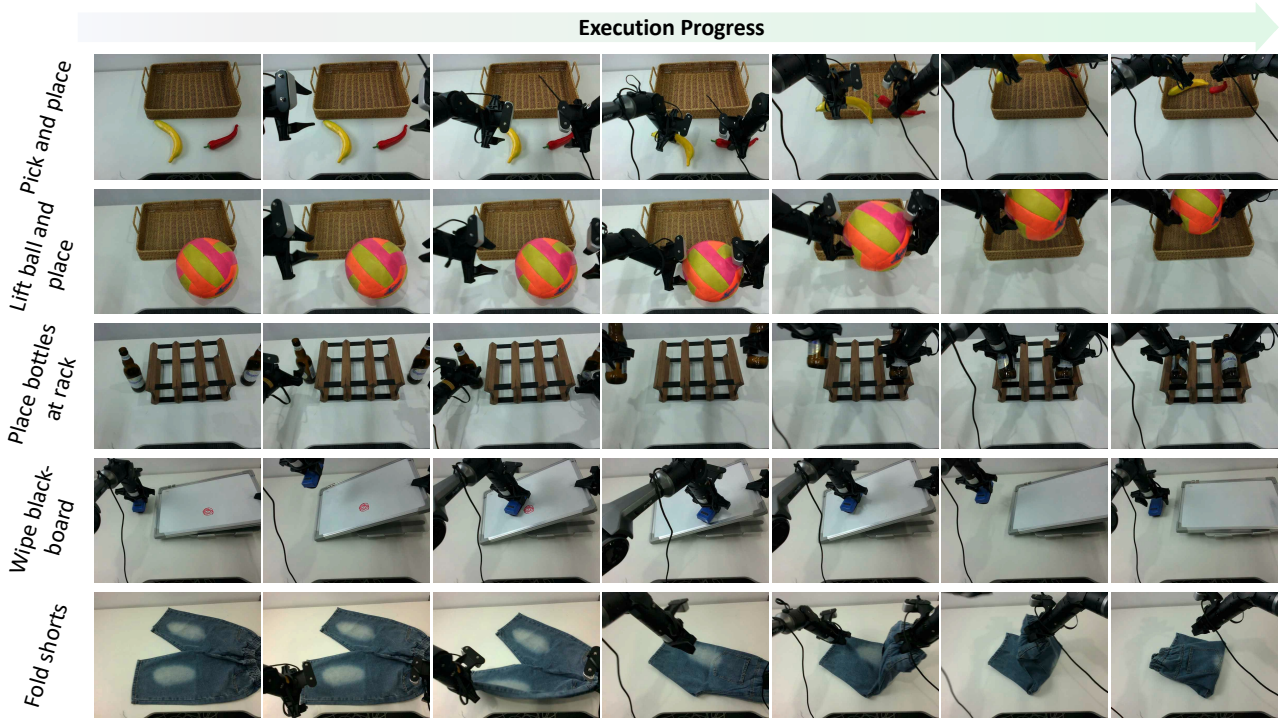


Figure 7. **Dual-arm Execution Visualization**. We visualize key frames of the agent's execution process from a static exterior view.

Franka Research 3 (FR3) operates with controller version 5.6.0, libfranka version 0.13.3, Franka ROS version 0.10.0, and Ubuntu 20.04 with ROS Noetic. Under these software settings, the FR3 remains in *green* light execution mode with the FCI switch set to 'on'.

These tasks demonstrate HybridVLA's capability in accurately predicting position and rotation, as well as determining the precise timing for changing the gripper's open state. Additionally, the dual-arm tasks highlight HybridVLA's ability to coordinate both robotic arms, enabling it to complete tasks beyond the capability of a single arm, such as transporting a ball to a container. Notably, the single-arm task 'open drawer and place' and the dual-arm tasks 'wipe whiteboard' and 'fold shorts' are long-horizon tasks that involve at least three multi-step actions. These results further confirm that HybridVLA can reliably predict sequential actions, demonstrating the capability to complete long-horizon tasks.

D. Failure Case Analysis.

Through extensive real-world experiments, we identify three primary failure categories that impact the performance of HybridVLA. The first category, **rotational prediction deviations**, is particularly evident in tasks requiring precise rotation control, such as *Pour water* and *Place bottle at rack*. These failures include accumulated errors in multi-step rotational movements and incorrect rotation angles when interacting with target objects. The second category pertains to pose predictions that exceed the robot's **degree of freedom limits**. The model sometimes predicts poses beyond the mechanical constraints of the Fr3 arm or AgileX dual-arm robot, generates target positions that fall outside the workspace boundaries, or produces kinematically infeasible configurations during complex transitions. The third category involves failures in **dual-arm coordination**, where both arms must collaborate to complete a task. Since the model predicts each arm's actions based on the current object state, any interaction by one arm can alter the object's state, potentially invalidating the previously predicted action of the other arm.