2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

DAM-VLA: A Dynamic Action Model-Based Vision-Language-Action Framework for Robot Manipulation

Xiongfeng Peng[1], Jiaqian Yu[1], Dingzhe Li[1], Yixiang Jin[1], Lu Xu[1], Yamin Mao[1], Chao Zhang[1], Weiming Li[1], Sujin Jang[2,3], Dongwook Lee[2], and Daehyun Ji[2]

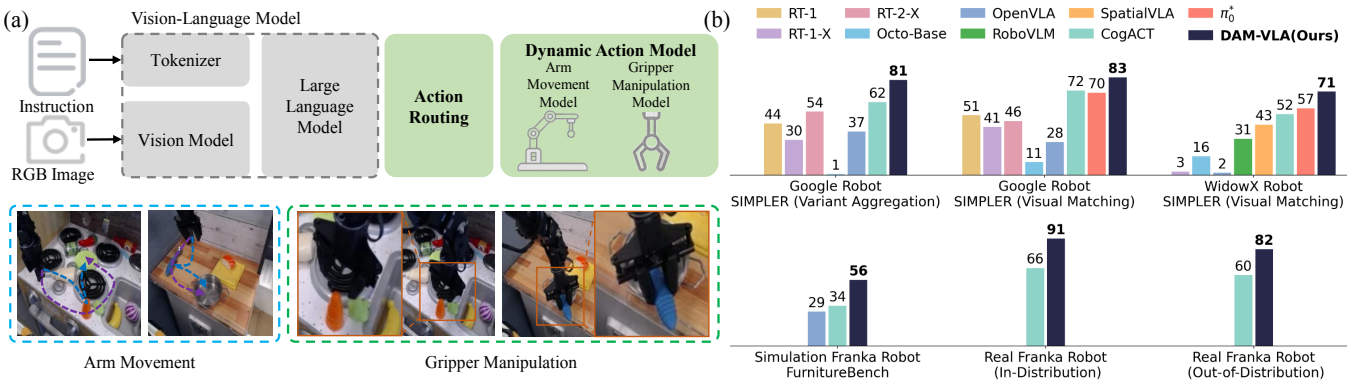

Fig. 1: DAM-VLA framework and experimental results. (a) We propose a DAM-VLA framework that dynamically integrates the inherent reasoning capabilities of VLMs with specialized diffusion-based action models tailored for arm movement and gripper manipulation. In various robotic tasks, arm movement typically covers a larger spatial range than gripper manipulation. consequently, in the observed images, the trajectories of the arm movement often occupy the majority of the region, while gripper manipulation is usually confined to a small, localized area; (b) Across extensive evaluations, our DAM-VLA achieves superior average success rates compared to state-of-the-art VLA methods, demonstrating improvements in both pick-and-place tasks within the SIMPLER simulation and long-horizon tasks on the FurnitureBench simulation, as well as in real-world pick-and-place evaluations.

***Abstract*— In dynamic environments such as warehouses, hospitals, and homes, robots must seamlessly transition between gross motion and precise manipulations to complete complex tasks. However, current Vision-Language-Action (VLA) frameworks, largely adapted from pre-trained Vision-Language Models (VLMs), often struggle to reconcile general task adaptability with the specialized precision required for intricate manipulation. To address this challenge, we propose DAM-VLA, a dynamic action model-based VLA framework. DAM-VLA integrates VLM reasoning with diffusion-based action models specialized for arm and gripper control. Specifically, it introduces (i) an action routing mechanism, using task-specific visual and linguistic cues to select appropriate action models (e.g., arm movement or gripper manipulation), (ii) a dynamic action model that fuses high-level VLM cognition with low-level visual features to predict actions, and (iii) a dual-scale action weighting mechanism that enables dynamic coordination between the arm-movement and gripper-manipulation models. Across extensive evaluations, DAM-VLA achieves superior success rates compared to state-of-the-art VLA methods in simulated (SIMPLER, FurnitureBench) and real-world settings, showing robust generalization from standard pick-and-place to demanding long-horizon and contact-rich tasks.**

[1]Xiongfeng Peng, Jiaqian Yu, Dingzhe Li, Yixiang Jin, Lu Xu, Yamin Mao, Chao Zhang, and Weiming Li are with Advanced Research Lab, Samsung R&D Institute China-Beijing (SRCB), China
[2]Sujin Jang, Dongwook Lee, and Daehyun Ji are with Samsung AI Center, DS Division, South Korea
[3]Sujin Jang is also with Hanyang University ERICA, South Korea

I. Introduction

A central challenge in robotics is enabling robots to perform diverse tasks in dynamic environments. Conventional robot learning methods typically train policies on datasets curated for a specific robot and task. The resulting policies act as specialists, such as the popular ACT [1] and Diffusion Policy [2]. Although these approaches achieve high precision in targeted scenarios, they generalize poorly across varying environments and tasks.

Recently, VLA models have attracted attention for their ability to extend pretrained VLMs to robotics by discretizing continuous actions into bins for action prediction. Representative works such as RT-2 [3] and OpenVLA [4] have demonstrated impressive performance in multi-task learning and generalization. By enabling robots to interpret visual observations and language instructions, VLA models can generate generalizable action sequences. Consequently, leveraging the inherent capabilities of VLMs in developing VLA frameworks is crucial for achieving both task-specific precision and broad generalization in dynamic environments.

On the one hand, several existing VLA methods, such as RT-H [5], RT-Affordance [6], and ECOT [7], incorporate Chain-of-Thought (CoT) reasoning to analyze spatial relationships between the gripper and the object, and predict the corresponding actions. Although CoT enhances the

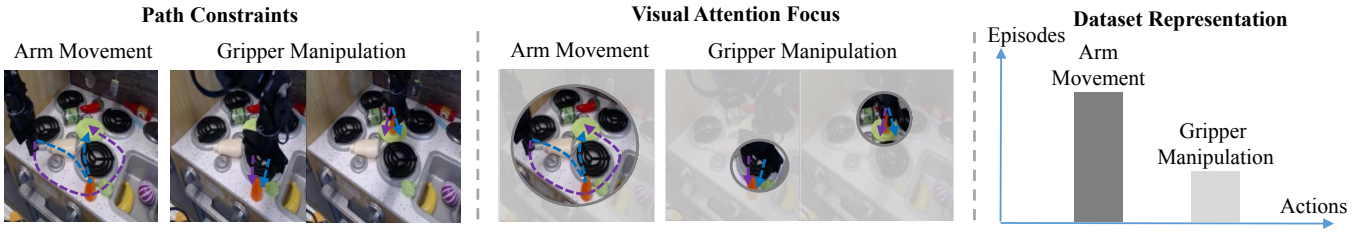

Fig. 2: We identify three distinctions between the arm movement and the gripper manipulation using the task of placing a carrot on a plate as an illustrative example: Path Constrains, Visual Attention, and Dataset Representation.

reasoning capabilities of VLMs and improves generalization, it introduces many extra reasoning tokens and substantially increase inference time. On the other hand, diffusion-based VLA methods, including $\pi 0$ [8], TinyVLA [9], RDT-1B [10], RoboDual [11], CogACT [12], and HybridVLA [13], append a separate diffusion head after the VLM. These methods either condition on VLM-extracted features or jointly embed the denoising timestep and noisy actions into the token sequence during diffusion. While enabling more precise manipulation, relying solely on VLM-extracted features limits the integration of richer multi-modal cues.

To better leverage the inherent capabilities of VLMs for VLA models that combine both task-specific and general manipulation in dynamic environments, we first identify several key distinctions between arm movement and gripper manipulation. In many robotic tasks, arm movement spans a larger spatial range than gripper manipulation. Consequently, arm trajectories often dominate the scene in the observed images, while gripper manipulations are confined to small, localized regions. Figure 2 illustrates this disparity using the task of placing a carrot on a plate as an example, where the non-gray regions of visual attention highlight the difference. Specifically, arm movement requires global attention, whereas gripper manipulation demands localized focus. The fundamental distinctions can be summarized as follows: (1) **Path Constraints**. Arm movement is relatively unconstrained since the robot can take multiple trajectories to reach the carrot. In contrast, gripper manipulation is highly constrained, requiring precise grasping postures for success. (2) **Visual Attention**. Arm movement depends on global scene understanding, whereas gripper manipulation necessitates fine-grained, localized visual attention. (3) **Dataset Representation**. Datasets usually contain far more arm movement episodes than gripper manipulation ones. Nevertheless, despite being fewer, gripper manipulations are critical for task success and often more complex.

Building on these distinctions, we leverage VLM reasoning to differentiate action types (arm movement vs. gripper manipulation), and apply the corresponding action model to perform the required manipulation. Rather than loosely coupling a VLM with separate action models, we introduce the **DAM-VLA** framework (Figure 1), which fully exploits the strengths of VLMs to support both task-specific precision and generalization in dynamic environments. The main contributions of this work are summarized as follows: (1) **Action Routing.** A VLM-guided router interprets task-specific visual and linguistic cues to select the appropriate action models (e.g., arm movement or gripper manipulation). (2) **Dynamic Action Model.** A dual-head diffusion model that integrates high-level cognition from the VLM with low-level visual information to predict actions across different models. (3) **Dual-Scale Action Weighting.** A two-scale weighting mechanism (i.e., trajectory-level and action-chunk-level) enables dynamic coordination between the arm-movement and gripper-manipulation models. (4) **Extensive Evaluation.** DAM-VLA achieves superior average success rates compared to state-of-the-art VLA methods, across both pick-and-place tasks in the SIMPLER simulation [14] and long-horizon tasks in the FurnitureBench simulation [15], as well as in real-world pick-and-place experiments.

II. Related Work

Vision-Language-Action Models. LLMs [16], [17], [18] and VLMs [19], [20], [21], [22], [23] inspire the development of VLA models, which extend VLMs by integrating action generation. RT-2 [3] tokenizes 7D actions into discrete tokens and employs the VLM PaLI-X [24] for prediction, while OpenVLA [4] follows a similar approach with the Prismatic VLM [25]. RT-H [5], RT-Affordance [6], and ECoT [7] incorporate Chain-of-Thought (CoT) reasoning to analyze spatial relationships between the gripper and objects before predicting actions. While these methods more effectively exploit the reasoning capabilities of pretrained VLMs to improve generalization, the large number of additional reasoning tokens (e.g., 7 in OpenVLA vs. 350 in ECoT) substantially slows inference and reduces control frequency. Moreover, they do not adequately address the continuity, accuracy, and specificity required for precise action estimation.

Diffusion Action Models. The Diffusion Policy [2] applies diffusion models [26], [27] to robot learning, demonstrating the ability to model multimodal action distributions. Subsequent research [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37] has further advanced this line of work by extending diffusion policies to 3D environments, scaling their capabilities, improving efficiency, and introducing architectural innovations. Among these, Octo [37] augments a transformer-based backbone with compact diffusion heads, enabling adaptation of action outputs across different robots. Although diffusion-based models improve efficiency and performance across diverse tasks and robotic platforms, they have yet to exploit pretrained LLMs and VLMs, whose

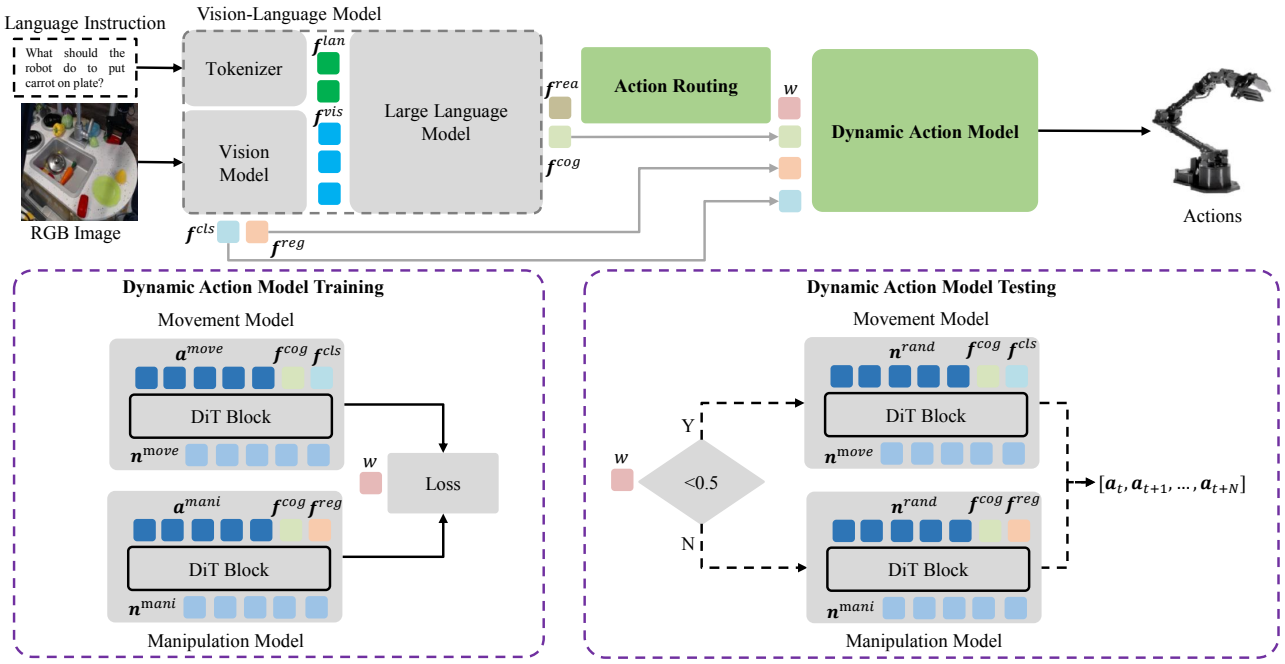

Fig. 3: The architecture of our DAM-VLA. Given an RGB image observation and a task description, the model predicts a sequence of temporal actions. The process consists of three key components: 1) a vision-language model that encodes observation into visual, class and register tokens, and integrates visual tokens with a set of linguistic tokens, and produces the cognition and reasoning latents; 2) an action routing module that generates a weight and feeds it into the dynamic action model; 3) a dynamic action model that dynamically executes different action models by combining the low-level class token or register token from the vision model with the high-level cognition latent from the VLM to predict an action sequence.

strong generalization and reasoning capabilities could further enhance policy robustness.

Diffusion-based Vision-Language-Action Models. To combine the strengths of VLMs and diffusion-based action models, $\pi 0$ [8] introduces a separate diffusion head to generate actions via flow matching, while TinyVLA [9] attaches a lightweight diffusion head after a compact VLM. CogACT [12] and DiVLA [38] decouple reasoning and action prediction, assigning these functions to the VLM and the diffusion head, respectively. HybridVLA [13] further integrates diffusion and autoregressive action prediction within a single LLM. However, these approaches do not fully exploit the inherent capabilities of VLMs to build VLA models that achieve both task-specific precision and generalizable manipulation in dynamic environments.

III. Method

In this section, we first describe the overall DAM-VLA architecture in Section III-A. Then we introduce the VLM in Section III-B. To fully leverage the specific manipulation capabilities of different action models and the VLM's inherent reasoning capabilities, we introduce an action routing mechanism and our dynamic action model in Section III-C. To further enhance robustness, we propose a dual-scale action weighting mechanism in Section III-D.

A. Overall Architecture

Our goal is to develop a dynamic action model-based VLA framework that enables different robots to physically execute diverse tasks in dynamic environments while receiving an RGB image observation and a task description in the form of a language instruction. Formally, given the language instruction $\boldsymbol{l}$ and visual observation $\boldsymbol{o}_t$ at time t, the model π predicts a temporal action sequence $[\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, ..., \boldsymbol{a}_{t+N}] = \pi(\boldsymbol{l}, \boldsymbol{o}_t)$. The action space $\boldsymbol{a}_t = [\delta \boldsymbol{x}, \delta \boldsymbol{\theta}, s^{grip}]$ corresponds to the gripper with 7 degrees of freedom (DoF), where $\delta \boldsymbol{x}$ represents the relative translation offsets of the end effector, $\delta \boldsymbol{\theta}$ denotes the rotational changes, and $s^{grip} \in \{0, 1\}$ indicates the gripper's open or close state.

In Figure 3, the architecture of DAM-VLA is shown to consist of three key components: 1) A vision-language model, that encodes information from observation $\boldsymbol{o}_t$ into visual tokens $\boldsymbol{f}^{vis}$, a class token $\boldsymbol{f}^{cls}$, and a register token $\boldsymbol{f}^{reg}$, and integrates visual tokens $\boldsymbol{f}^{vis}$ with a set of linguistic tokens $\boldsymbol{f}^{lan}$ from a language instruction $\boldsymbol{l}$, and produces the cognition latent $\boldsymbol{f}^{cog}$ and reasoning latent $\boldsymbol{f}^{rea}$; 2) An action routing module that generates a weight w and feeds it into the dynamic action model; 3) A dynamic action model that dynamically executes different action models by combining the low-level class token $\boldsymbol{f}^{cls}$ or register token $\boldsymbol{f}^{reg}$ from the vision model with the high-level cognition latent $\boldsymbol{f}^{cog}$ from the VLM to predict an action sequence $[\boldsymbol{a}_t, \boldsymbol{a}_{t+1}, ..., \boldsymbol{a}_{t+N}]$.

B. Vision-Language Model

The vision model processes the RGB image input into a set of tokens, which include not only visual tokens $\boldsymbol{f}^{vis}$, but also a class token $\boldsymbol{f}^{cls}$ and a register token $\boldsymbol{f}^{reg}$ from DINOv2 [39]. The vision model consists of powerful

vision transformers, DINOv2 and SigLIP [40], pretrained on internet-scale image data to capture both low-level rich visual features and high-level semantic understanding. At each timestep t, the image observation $\boldsymbol{o}_t$ is fed into both the arm movement model and the gripper manipulation model, each producing a downsampled feature map. These feature maps are then concatenated along the channel dimension, passed through a linear projection layer, and serialized into a set of tokens, including visual tokens $\boldsymbol{f}^{vis} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_{NV}]$, a class token $\boldsymbol{f}^{cls}$, and a register token $\boldsymbol{f}^{reg}$.

The large language model is responsible for integrating visual information and language instructions and estimating the reasoning token. We use an LLaMA-2 model [18] as the backbone. The language instruction $\boldsymbol{l}$ is tokenized into a set of linguistic tokens, $\boldsymbol{f}^{lan} = [\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_{NL}]$. These tokens $\boldsymbol{f}^{lan}$ are then concatenated with the visual tokens $\boldsymbol{f}^{vis}$ and an additional learnable reasoning token, and are processed by the large language model using a causal attention mechanism. The resulting output consists of the cognition and reasoning latents, $\boldsymbol{f}^{cog}$ and $\boldsymbol{f}^{rea}$, respectively. $\boldsymbol{f}^{rea}$ and $\boldsymbol{f}^{cog}$ are derived from the hidden features of the second and last transformer layers of the LLM, respectively. $\boldsymbol{f}^{rea}$ serves as the input to the subsequent action routing module to select the appropriate action models, while $\boldsymbol{f}^{cog}$ serves as the input for the action model to predict the actions.

C. Action Routing Mechanism and Dynamic Action Model

To determine whether the action state is in arm movement or gripper manipulation, we design an action routing mechanism. It leverages the reasoning ability of the VLM to learn the weight w and it is supervised by the labeled weight $\hat{w} \in 0, 1$ we designed for dynamically executing either the arm movement model or the gripper manipulation model. The labeled weight $\hat{w}$ is obtained based on the state of the robot gripper. When the state of the robot gripper changes from open to closed or vice versa, we define this as a transition of the robot action from arm movement ($\hat{w} = 0$) to gripper manipulation ($\hat{w} = 1$). The detailed calculation process for $\hat{w}$ is explained in the a dual-scale action weighting section. To leverage the reasoning ability of the VLM, the reasoning latent $\boldsymbol{f}^{rea}$ from the VLM is input to the action routing module, where it is processed through a fully connected layer and a normalization layer. The output is the predicted weight w, which is supervised by the following cross-entropy loss:

$$L_{class} = || - (\hat{w} \log(w) + (1 - \hat{w}) \log(1 - w))||^1. \quad (1)$$

To fully leverage the specific manipulation capabilities of different diffusion action models and the VLM's inherent reasoning capabilities, we propose the dynamic action model. In the training phase, the dynamic action model enables targeted learning of the arm movement model and the gripper manipulation model using the labeled weights $\hat{\boldsymbol{w}}^{move}$ and $\hat{\boldsymbol{w}}^{mani}$. These weights are calculated by the trajectory weights and the action chunk weights. A more detailed explanation of the design of $\hat{\boldsymbol{w}}^{move}$ and $\hat{\boldsymbol{w}}^{mani}$ is provided in the dual-scale action weighting section. In addition to the high-level VLM-extracted cognition latent $\boldsymbol{f}^{cog}$ as a condition, the two action models also take lower-level visual tokens $\boldsymbol{f}^{cls}$ or $\boldsymbol{f}^{reg}$ as conditions, respectively. Since the two models focus on different attention focuses in the image, we input different visual tokens. The arm movement model requires more global attention, so we input the class token $\boldsymbol{f}^{cls}$ as a condition. In contrast, the gripper manipulation model focuses on more local attention, so we input the register token $\boldsymbol{f}^{reg}$. Regarding the Diffusion Transformer (DiT) [27] block of the action models, we refer to [12]. The loss functions for the arm movement and the gripper manipulation models are supervised separately as follows:

$$L_{move} = ||\boldsymbol{n}_i^{move} - \hat{\boldsymbol{n}}^{move}||^2_{\sum \hat{\boldsymbol{w}}^{move}}, \quad (2)$$

$$L_{mani} = ||\boldsymbol{n}_i^{mani} - \hat{\boldsymbol{n}}^{mani}||^2_{\sum \hat{\boldsymbol{w}}^{mani}}, \quad (3)$$

where $|| \cdot ||_{\sum}$ denotes the Mahalanobis distance, which weights the error terms based on the labeled weights $\hat{\boldsymbol{w}}^{move}$ and $\hat{\boldsymbol{w}}^{mani}$. $\boldsymbol{n}_i^{move}$ and $\boldsymbol{n}_i^{mani}$ represent the predicted noise values for the arm movement model and the gripper manipulation model, respectively, at the i-th denoising step for the noised action sequence. $\hat{\boldsymbol{n}}^{move}$ and $\hat{\boldsymbol{n}}^{mani}$ correspond to the ground truth noise values of the diffusion action models for movement and manipulation, respectively, and are randomly generated. The inputs to the two action models include the noised actions $\boldsymbol{a}^{move}$ and $\boldsymbol{a}^{mani}$. $\boldsymbol{a}^{move}$ is calculated using the ground truth noise $\hat{\boldsymbol{n}}^{move}$ and the ground truth action $\hat{\boldsymbol{a}}$, while $\boldsymbol{a}^{mani}$ is computed using the ground truth noise $\hat{\boldsymbol{n}}^{mani}$ and the ground truth action $\hat{\boldsymbol{a}}$. The total loss is computed as a weighted sum of the movement loss, the manipulation loss, and the classification loss. The corresponding hyperparameters w_1, w_2, and w_3 are set to 1.0, 1.0, and 0.0001, respectively.

$$L = w_1 * L_{move} + w_2 * L_{mani} + w_3 * L_{class}. \quad (4)$$

In the testing phase, the dynamic action model selects and executes the appropriate action model based on the predicted weight w. If $w < 0.5$, the model runs the arm movement model, using the high-level VLM-extracted cognition latent $\boldsymbol{f}^c$ and the global class token $\boldsymbol{f}^{cls}$ as conditions to predict a sequence of multi-step actions. Otherwise, if $w \geq 0.5$, the model runs the gripper manipulation model, using the cognition latent $\boldsymbol{f}^c$ and the local register token $\boldsymbol{f}^{reg}$. Additionally, both models receive random noise $\boldsymbol{n}^{rand}$ as input to facilitate the diffusion process.

D. Dual-Scale Action Weighting

To enhance the robustness in distinguishing between arm movement and gripper manipulation, we propose a dual-scale action weighting mechanism for model training, as illustrated in Figure 4. The core objective of the dual-scale action weighting mechanism is to adaptively supervise the learning process by modulating the importance of different action types from both global (trajectory-level) and local (action-chunk-level) perspectives.

Trajectory-level Weights ($\boldsymbol{w}^t$): This global perspective segments the entire task trajectory into distinct phases

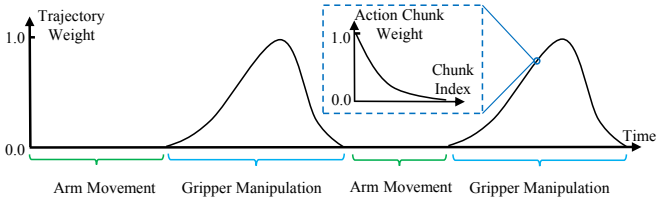

Fig. 4: Illustration of the dual-scale action weighting mechanism. The trajectory weight highlights critical manipulation phases via asymmetrical Gaussian distributions. Within each predicted chunk, the action chunk weight applies exponential decay to prioritize immediate temporal accuracy. The final weight integrates both scales to guide model supervision.

of arm movement and gripper manipulation based on the binary state changes of the robot gripper. For each gripper manipulation process k, we employ an asymmetrical Gaussian distribution $\{\mathcal{N}(u, \sigma_l^2), \mathcal{N}(u, \sigma_r^2)\}$ to define the weights $\boldsymbol{w}_k^t$. Both distributions share the same mean u, representing the temporal midpoint where the gripper state transitions (e.g., from open to closed). To reflect the prior that the action model requires higher precision and supervision focus immediately before the state change, we assign a larger variance to the leading edge ($\sigma_l = 6$) and a smaller variance to the trailing edge ($\sigma_r = 2$). The aggregated trajectory weights are defined as $\boldsymbol{w}^t = \text{Norm}(\sum_k \boldsymbol{w}_k^t)$, representing the normalized sum of all manipulation-related weights.

Action-chunk-level Weights ($\boldsymbol{w}^a$): From a local perspective, we account for the inherent temporal uncertainty in action sequences. Given that prediction confidence typically decays as the temporal distance from the current state increases, we apply an exponentially decreasing function: $\boldsymbol{w}_j^a = \gamma^j$, where j denotes the index within the action chunk and $\gamma = 0.8$ is the decay factor.

Multi-scale Integration: The final weights are formulated as $\boldsymbol{w}^{move} = (1 - \boldsymbol{w}^t) \odot \boldsymbol{w}^a$ and $\boldsymbol{w}^{mani} = \boldsymbol{w}^t \odot \boldsymbol{w}^a$. By applying these weights to L_{move} and L_{mani}, our proposed dual-scale action weighting mechanism dynamically coordinates the arm-movement and gripper-manipulation models through pointwise modulation.

Furthermore, the labeled weight $\hat{w}$ is derived from the trajectory weights: $\hat{w} = 1$ if $\hat{\boldsymbol{w}}^t > 0.5$, and $\hat{w} = 0$ otherwise. This weight $\hat{w}$ acts as the ground-truth label for the predicted weight w, supervised via the cross-entropy loss L_{class}.

IV. Experiments

We conduct extensive experiments to comprehensively evaluate the performance of our proposed method and to clearly demonstrate its effectiveness in both task-specific and general-purpose manipulation scenarios. Specifically, Section IV-A details the training and fine-tuning procedures. We then present the experimental results on SIMPLER [14] and FurnitureBench [15] in Section IV-B. We also conduct real-world evaluations based on a pick-and-place task and present the results in Section IV-C. Section IV-D provides an ablation study to analyze the contribution of each component in our framework.

Method / Google(VA)	Success Rates on Different Tasks				Avg
	PCC	MN	OCD	ODPA	
RT-1 [3]	90%	46%	35%	3%	44%
RT-1-X [44]	49%	33%	29%	10%	30%
RT-2-X [44]	82%	79%	35%	21%	54%
Octo-Base [37]	1%	4%	1%	0%	1%
RoboVLM [45]	76%	60%	11%	-	-
SpatialVLA [46]	88%	73%	42%	-	-
OpenVLA [4]	64%	64%	19%	1%	37%
CogACT [12]	96%	**84%**	29%	40%	62%
DAM-VLA(Ours)	**98%**	74%	**68%**	**84%**	**81%**

TABLE I: Comparison of success rates between our method and existing VLA methods on the Google robot in Variant Aggregation (VA) setting of the SIMPLER simulated evaluation across four tasks. (PCC: Pick Coke Can, MN: Move Near, OCD: Open/Close Drawer, ODPA: Open Drawer and Place Apple, Avg: Average)

Method / Google(VM)	Success Rates on Different Tasks				Avg
	PCC	MN	OCD	ODPA	
RT-1 [3]	87%	39%	72%	8%	51%
RT-1-X [44]	59%	33%	56%	17%	41%
RT-2-X [44]	79%	78%	25%	4%	46%
Octo-Base [37]	18%	4%	24%	0%	11%
OpenVLA [4]	14%	51%	48%	0%	28%
RoboVLM [45]	77%	62%	44%	-	-
SpatialVLA [46]	86%	78%	57%	-	-
CogACT [12]	92%	82%	**75%**	39%	72%
π_0^* [8]$^+$	89%	81%	55%	53%	70%
DAM-VLA(Ours)	**96%**	**84%**	**75%**	**78%**	**83%**

TABLE II: Comparison of success rates between our method and existing VLA methods on the Google robot in Visual Matching (VM) setting of the SIMPLER simulated evaluation across four tasks. (PCC: Pick Coke Can, MN: Move Near, OCD: Open/Close Drawer; ODPA: Open Drawer and Place Apple, Avg: Average; $^+$: open-pi-zero, which is trained on Fractal dataset.)

A. Training and Fine-tuning Details

The large-scale Open X-Embodiment Dataset [41] contains over 1 million robot manipulation trajectories collected from across 22 distinct robotic embodiments. We primarily utilize two of its major subsets, Fractal [42] and BridgeDataV2 [43], as our primary pre-training dataset. Our VLA model is trained using a constant learning rate of 2×10^{-5} and a batch size of 256 on 8 NVIDIA H100 GPUs for approximately two days.

Furthermore, we fine-tune our DAM-VLA model on both simulated and real-world datasets. For the simulation experiments, we specifically use the FurnitureBench benchmark, which involves contact-rich and long-horizon manipulation tasks. Specifically, we fine-tune our DAM-VLA model on 500 expert trajectories from the "One-Leg" assembly task. For real-world evaluation, we construct a pick-and-place scenario in which a Franka robot is teleoperated to pick up a cup and place it into a bowl. A total of 50 demonstrated trajectories were collected for fine-tuning. The fine-tuning process adopts the same hyperparameters as pre-training: a learning rate of 2×10^{-5} and a batch size of 256, utilizing 8 NVIDIA H100 GPUs. For a fair comparison, we also fine-tune the OpenVLA and CogACT baselines using the identical datasets.

Method / WidowX(VM)	Success Rates on Different Tasks				Avg
	SoT	CoP	GoY	EiB	
RT-1-X [44]	4%	8%	0%	0%	3%
Octo-Base [37]	7%	13%	0%	44%	16%
Octo-Small [37]	47%	8%	1%	51%	27%
OpenVLA [4]	4%	0%	0%	4%	2%
ECOT [7]	4%	8%	0%	0%	3%
RoboVLM [45]	29%	25%	13%	58%	31%
SpatialVLA [46]	17%	25%	**29%**	**100%**	43%
CogACT[12]	63%	50%	25%	71%	52%
π_0^*[8]$^+$	62%	59%	24%	81%	57%
DAM-VLA(Ours)	**88%**	**71%**	25%	**100%**	**71%**

TABLE III: Comparison of success rates between our method and existing VLA methods on the WidowX robot in the Visual Matching (VM) setting of the SIMPLER simulated evaluation across four tasks. (SoT: Put Spoon on Towel, CoP: Put Carrot on Plater; GoY: Stack Green Block on Yellow Block; EiB: Put Eggplant in Yellow Basket; Avg: Average; $^+$: open-pi-zero, which is trained on BridgeDataV2.)

B. Simulated Evaluations

We first evaluate our method using the SIMPLER simulation [14], a suite of open-source simulated environments designed to mirror common real-world robot manipulation setups. Compared to real-world evaluations, this real-to-sim approach offers a scalable, reproducible, and informative tool that complements high-quality real-world assessments. Extensive testing of various VLA models has shown a strong correlation between evaluations in SIMPLER's simulated environments and real-world performance. SIMPLER supports two robot embodiments: the Google robot and the WidowX robot. For the Google robot, evaluations are conducted under both Visual Matching (VM) and Variant Aggregation (VA) settings across four tasks, whereas the WidowX robot is evaluated only under the VM setting. The VM setting closely replicates real-world tasks by minimizing discrepancies between simulated and real-world environments. In contrast, the VA setting extends the VM setting by introducing variations in factors such as background, lighting, distractors, and table texture. The success rate of task completion is used as the evaluation metric for all VLA models. Notably, we follow CogACT [12] in determining the number of trials conducted in SIMPLER.

Tables I and II compare our method against existing VLA approaches on the Google robot across four tasks. Our model leads with average success rates of 83% (VM) and 81% (VA). Notably, we see substantial gains in the "Open Drawer and Place Apple" task, which requires task-specific manipulation. Moreover, in the VA setting, our success rate markedly exceeds competitors, demonstrating DAM-VLA mitigates performance degradation in dynamic environments.

Table III reports the success rates of our method compared with existing VLA approaches on the WidowX robot in SIMPLER across four tasks under the VM setting. Our model achieves the highest average success rate of 71%, outperforming competing methods by a substantial margin. In particular, DAM-VLA shows notable improvements in the "Put Spoon on Towel" and "Put Carrot on Plate" tasks, which involve more diverse object pose variations.

In addition to the SIMPLER, we further evaluate our method on the "One-Leg" assembly task from FurnitureBench [15], which involves contact-rich and long-horizon manipulation. The full task sequence is illustrated in Figure 5. We conduct 50 evaluation trials with randomized initial furniture placements. As shown in Table IV are the success rates of each step of the "One-Leg" assembly task compared with OpenVLA and CogACT models. Our method consistently outperforms prior VLA approaches on this challenging furniture assembly task, demonstrating superior generalization and precision under contact-rich manipulation settings. More specifically, we achieve a 100% success rate in the first three steps of "grasp tabletop", "place tabletop to corner", and "pick up leg", and we achieve higher success rates in the contact-rich manipulation "screw leg".

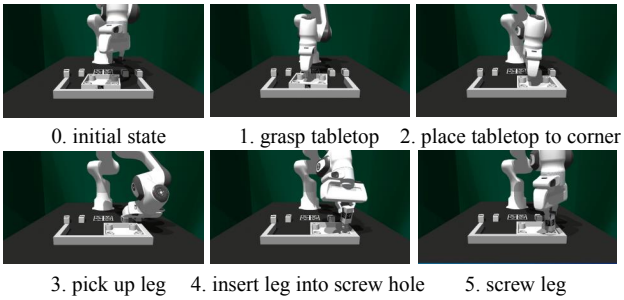

Fig. 5: The entire process of the "One-Leg" assembly task in the FurnitureBench environment.

Method / FurnitureBench	Success Rates at Each Step				
	1	2	3	4	5
OpenVLA [4]	96%	94%	78%	53%	29%
CogACT [12]	98%	96%	96%	56%	42%
DAM-VLA(Ours)	**100%**	**100%**	**100%**	**62%**	**56%**

TABLE IV: The success rate of each step of the "One-Leg" assembly task in FurnitureBench is compared with the existing OpenVLA and CogACT methods. The fifth step represents the final success rate of the task.

C. Real-world Evaluations

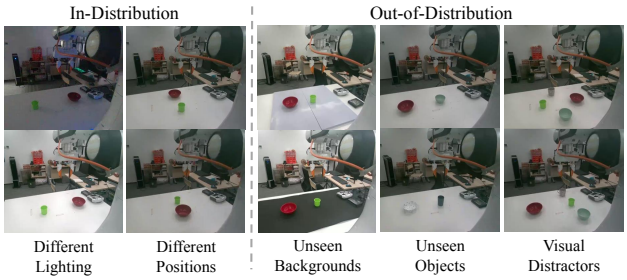

Fig. 6: The evaluation encompasses both in-distribution and out-of-distribution scenarios. The in-distribution setting includes variations in object positions and lighting conditions consistent with the training data, while the out-of-distribution setting introduces novel backgrounds, unseen objects, and visual distractors absent during training.

Our real-world dataset is collected under diverse object placements and lighting conditions. To assess robustness, we divide the evaluation into in-distribution and out-of-distribution scenarios, as illustrated in Figure 6. The in-

Method / Real-world	Success Rates on Different Scenarios		
	ID	OOD	Avg
CogACT	65.7%	60.0%	62.9%
DAM-VLA (Ours)	**91.4%**	**82.2%**	**86.8%**

TABLE V: The success rates of our method compared with CogACT on the pick-and-place task in the real-world evaluation. ID: In-Distribution, OOD: Out-of-Distribution.

Components					Success Rates			
VT	ACW	TW	DAM	DL	Google (VM)	Google (VA)	WidowX (VM)	Avg
-	-	-	-	-	64%	61%	50%	58%
✓	✓	-	-	-	78%	68%	53%	66%
✓	✓	✓	-	-	76%	63%	51%	63%
✓	✓	✓	✓	-	82%	72%	43%	66%
✓	✓	✓	✓	✓	83%	**81%**	**71%**	**78%**
-	✓	✓	✓	✓	**84%**	75%	58%	73%
-	-	-	✓	✓	66%	64%	49%	60%

TABLE VI: An ablation study is conducted on the WidowX robot in Visual Matching (VM) setting and the Google robot in both VM and Variant Aggregation (VA) settings of the SIMPLER environment.

distribution scenario includes variations in object positions and lighting that are consistent with the training distribution. In contrast, the out-of-distribution scenario introduces previously unseen backgrounds, novel objects, and visual distractors that are absent during training.

Table V reports the success rates of DAM-VLA compared with the baseline CogACT on the real-world pick-and-place task. To ensure a fair comparison, each method is evaluated with 5 trials per condition across 16 distinct conditions, covering both in-distribution and out-of-distribution scenarios, for a total of 80 trials. The results demonstrate that DAM-VLA consistently outperforms CogACT, achieving higher success rates in both evaluation settings.

D. Ablation Study

For the ablation studies, we employ the SIMPLER evaluation environment using the WidowX robot in the VM setting and the Google robot in both the VM and VA settings. To assess the contribution of different components of our method, we conduct a detailed analysis of their corresponding success rates. As shown in Table VI, the main components under investigation include: (1) **Visual Tokens (VT)**: the visual class and register tokens output by the vision model, used by the action models to predict actions. (2) **Action Chunk Weights (ACW)**: the action-chunk weights used to supervise DAM-VLA. (3) **Trajectory Weights (TW)**: the trajectory weights used to supervise DAM-VLA. (4) **Dynamic Action Model (DAM)**: consisting of the arm movement model and the gripper manipulation model. (5) **Dual Latents (DL)**: the reasoning latent and cognition latent extracted from different transformer layers of the LLM. Starting from a baseline VLA model with a single action model, adding VT and ACW improves performance on both the Google and WidowX robots. Since current VLA consists of only a single diffusion action model, VT here only contains the class token. Incorporating TW, which is specifically designed for dual action models, leads to a slight decrease in performance on both robots, which is expected. Adding DAM and DL subsequently yields the highest average success rate of 78%, which demonstrates the effectiveness of our DAM-VLA framework. When the VLA is configured with dual action models, VT contains both the class and register tokens. Finally, removing VT and the dual-scale action weighting mechanism (ACW and TW) from DAM-VLA results in a substantial performance drop, which further emphasizes the critical role of both the dynamic action model and the dual-scale action weighting mechanism. Notably, removing VT means removing both the class and register tokens.

V. Conclusion

Our proposed DAM-VLA method dynamically integrates the inherent reasoning capabilities of VLMs with specialized diffusion-based action models designed for arm movement and gripper manipulation. Extensive experiments demonstrate that our approach not only significantly outperforms existing VLA methods but also delivers stable results in both task-specific and general manipulation scenarios within dynamic environments. By dynamically routing between specialized action models based on VLM-guided cues, DAM-VLA paves a new path for incorporating semantic understanding into embodied decision-making. Its demonstrated generalization ability and stability across diverse tasks and dynamic settings highlight its potential as a foundational framework for next-generation adaptable robotic systems in real-world applications.

While DAM-VLA achieves strong performance, several aspects remain open for further improvement. First, although our experimental evaluation already involves contact-rich and long-horizon tasks, it is currently centered on pick-and-place and furniture assembly; expanding to more diverse task families will provide broader validation. Second, while DAM-VLA substantially improves the success rate of most tasks, the relatively modest gains in the "Stack Green Block on Yellow Block" scenario indicate opportunities to enhance fine-grained coordination and stability. Finally, DAM-VLA currently routes between only two action models; generalizing this mechanism to handle multiple action types with richer routing signals will further broaden its applicability to complex real-world settings.

References

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

[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*, p. 02783649241273668, 2023.

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

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

[5] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh, "Rt-h: Action hierarchies using language," *arXiv preprint arXiv:2403.01823*, 2024.

[6] S. Nasiriany, S. Kirmani, T. Ding, L. Smith, Y. Zhu, D. Driess, D. Sadigh, and T. Xiao, "Rt-affordance: Affordances are versatile intermediate representations for robot manipulation," *arXiv preprint arXiv:2411.02704*, 2024.

[7] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine, "Robotic control via embodied chain-of-thought reasoning," *arXiv preprint arXiv:2407.08693*, 2024.

[8] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π0: A vision-language-action flow model for general robot control, 2024," *URL https://arxiv.org/abs/2410.24164*.

[9] J. Wen, Y. Zhu, J. Li, M. Zhu, Z. Tang, K. Wu, Z. Xu, N. Liu, R. Cheng, C. Shen *et al.*, "Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation," *IEEE Robotics and Automation Letters*, 2025.

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

[11] Q. Bu, H. Li, L. Chen, J. Cai, J. Zeng, H. Cui, M. Yao, and Y. Qiao, "Towards synergistic, generalized, and efficient dual-system for robotic manipulation," *arXiv preprint arXiv:2410.08001*, 2024.

[12] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang *et al.*, "Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation," *arXiv preprint arXiv:2411.19650*, 2024.

[13] J. Liu, H. Chen, P. An, Z. Liu, R. Zhang, C. Gu, X. Li, Z. Guo, S. Chen, M. Liu *et al.*, "Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model," *arXiv preprint arXiv:2503.10631*, 2025.

[14] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani *et al.*, "Evaluating real-world robot manipulation policies in simulation," *arXiv preprint arXiv:2405.05941*, 2024.

[15] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," *The International Journal of Robotics Research*, p. 02783649241304789, 2023.

[16] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[17] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[18] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[19] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022.

[20] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[21] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A frontier large vision-language model with versatile abilities," *arXiv preprint arXiv:2308.12966*, vol. 1, no. 2, p. 3, 2023.

[22] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[23] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.

[24] X. Chen, J. Djolonga, P. Padlewski, B. Mustafa, S. Changpinyo, J. Wu, C. R. Ruiz, S. Goodman, X. Wang, Y. Tay *et al.*, "Pali-x: On scaling up a multilingual vision and language model," *arXiv preprint arXiv:2305.18565*, 2023.

[25] S. Karamcheti, S. Nair, A. Balakrishna, P. Liang, T. Kollar, and D. Sadigh, "Prismatic vlms: Investigating the design space of visually-conditioned language models," in *Forty-first International Conference on Machine Learning*, 2024.

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

[27] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4195–4205.

[28] K. Black, M. Nakamoto, P. Atreya, H. Walke, C. Finn, A. Kumar, and S. Levine, "Zero-shot robotic manipulation with pretrained image-editing diffusion models," *arXiv preprint arXiv:2310.10639*, 2023.

[29] S. Dasari, O. Mees, S. Zhao, M. K. Srirama, and S. Levine, "The ingredients for robotic diffusion transformers," *arXiv preprint arXiv:2410.10088*, 2024.

[30] A. Z. Ren, J. Lidard, L. L. Ankile, A. Simeonov, P. Agrawal, A. Majumdar, B. Burchfiel, H. Dai, and M. Simchowitz, "Diffusion policy policy optimization," *arXiv preprint arXiv:2409.00588*, 2024.

[31] M. Reuss, Ö. E. Yağmurlu, F. Wenzel, and R. Lioutikov, "Multimodal diffusion transformer: Learning versatile behavior from multimodal goals," *arXiv preprint arXiv:2407.05996*, 2024.

[32] M. Uehara, Y. Zhao, K. Black, E. Hajiramezanali, G. Scalia, N. L. Diamant, A. M. Tseng, T. Biancalani, and S. Levine, "Fine-tuning of continuous-time diffusion models as entropy-regularized control," *arXiv preprint arXiv:2402.15194*, 2024.

[33] M. Uehara, Y. Zhao, K. Black, E. Hajiramezanali, G. Scalia, N. L. Diamant, A. M. Tseng, S. Levine, and T. Biancalani, "Feedback efficient online fine-tuning of diffusion models," *arXiv preprint arXiv:2402.16359*, 2024.

[34] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid, "Aloha unleashed: A simple recipe for robot dexterity," *arXiv preprint arXiv:2410.13126*, 2024.

[35] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *arXiv preprint arXiv:2402.10885*, 2024.

[36] X. Jia, Q. Wang, A. Donat, B. Xing, G. Li, H. Zhou, O. Celik, D. Blessing, R. Lioutikov, and G. Neumann, "Mail: Improving imitation learning with selective state space models," in *8th Annual Conference on Robot Learning*, 2024.

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

[38] J. Wen, M. Zhu, Y. Zhu, Z. Tang, J. Li, Z. Zhou, C. Li, X. Liu, Y. Peng, C. Shen *et al.*, "Diffusion-vla: Scaling robot foundation models via unified diffusion and autoregression," *arXiv preprint arXiv:2412.03293*, 2024.

[39] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[40] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 11 975–11 986.

[41] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.

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

[43] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du *et al.*, "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning*. PMLR, 2023, pp. 1723–1736.

[44] Q. Vuong, S. Levine, H. R. Walke, K. Pertsch, A. Singh, R. Doshi, C. Xu, J. Luo, L. Tan, D. Shah *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," in *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition CoRL2023*, 2023.

[45] X. Li, P. Li, M. Liu, D. Wang, J. Liu, B. Kang, X. Ma, T. Kong, H. Zhang, and H. Liu, "Towards generalist robot policies: What matters in building vision-language-action models," *arXiv preprint arXiv:2412.14058*, 2024.

[46] D. Qu, H. Song, Q. Chen, Y. Yao, X. Ye, Y. Ding, Z. Wang, J. Gu, B. Zhao, D. Wang *et al.*, "Spatialvla: Exploring spatial representations for visual-language-action model," *arXiv preprint arXiv:2501.15830*, 2025.