This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

SwiftVLA: Unlocking Spatiotemporal Dynamics for Lightweight VLA Models at Minimal Overhead

Chaojun Ni[1,2*] Cheng Chen[3,4*] Xiaofeng Wang[2,5*] Zheng Zhu[2*†] Wenzhao Zheng[5]
Boyuan Wang[2] Tianrun Chen[4†] Guosheng Zhao[2] Haoyun Li[2] Zhehao Dong[1,2]
Qiang Zhang[2] Yun Ye[2] Yang Wang[2] Guan Huang[2] Wenjun Mei[1†]
[1]Peking University [2]GigaAI [3]Nanyang Technological University
[4]Moxin (Hangzhou) Technology Co., Ltd. [5]Tsinghua University

Project Page: https://Swiftvla.github.io

Abstract

Vision–Language–Action (VLA) models built on pretrained Vision–Language Models (VLMs) show strong potential but are limited in practicality due to their large parameter counts. To mitigate this issue, using a lightweight VLM has been explored, but it compromises spatiotemporal reasoning. Although some methods suggest that incorporating additional 3D inputs can help, they usually rely on large VLMs to fuse 3D and 2D inputs and still lack temporal understanding. Therefore, we propose SwiftVLA, an architecture that enhances a compact model with 4D understanding while preserving design efficiency. Specifically, our approach features a pretrained 4D visual geometry transformer with a temporal cache that extracts 4D features from 2D images. Then, to enhance the VLM's ability to exploit both 2D images and 4D features, we introduce Fusion Tokens, a set of learnable tokens trained with a future prediction objective to generate unified representations for action generation. Finally, we introduce a mask-and-reconstruct strategy that masks 4D inputs to the VLM and trains the VLA to reconstruct them, enabling the VLM to learn effective 4D representations and allowing the 4D branch to be dropped at inference with minimal performance loss. Experiments in real and simulated environments show that SwiftVLA outperforms lightweight baselines and rivals VLAs up to 7× larger, achieving comparable performance on edge devices while being 18× faster and reducing memory footprint by 12×.

1. Introduction

Visual–Language–Action (VLA) models [6, 28–32, 35, 39, 44, 57–60, 62, 64, 67, 73, 76, 77, 79, 82–84, 90–93] represent a new paradigm in robotics, leveraging the representational and reasoning strengths of large, pretrained Vision–Language Models (VLMs) [1, 3, 8, 12, 23, 52, 54, 79] to map natural-language instructions and visual observations directly to actions. Despite their promise, real-world deployment is hindered by a significant obstacle: the massive parameter counts of foundation VLMs induce high inference latency and memory usage, which is prohibitive for real-time control on resource-constrained robotic platforms.

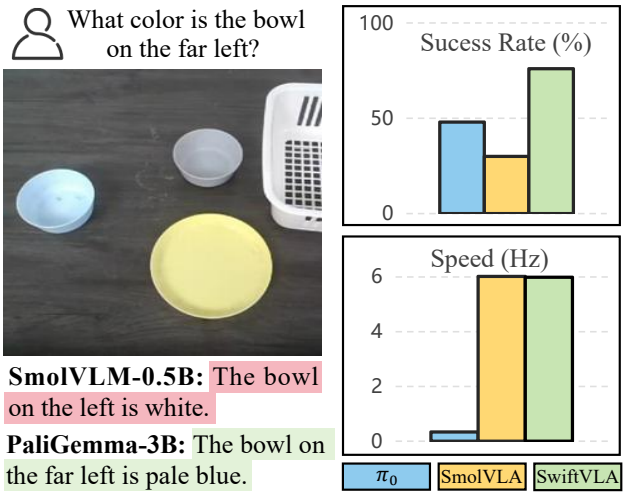


Figure 1. Large VLMs like PaliGemma-3B [3] excel in spatial reasoning over small VLMs [45], with correct answers in green and incorrect ones in red. This performance advantage allows π_0 [6] based on it to achieve a higher success rate, despite slower inference speed compared to the SmolVLA [55] based on a small VLM. However, SwiftVLA enhances spatiotemporal dynamics for small VLA models while preserving the speed advantages. The success rate and speed are tested on the NVIDIA Jetson Orin [49].

Therefore, recent studies [2, 55, 71] have reduced model capacity by shrinking the size of VLMs or decreasing the number of network layers, enabling deployment on edge devices. However, merely compressing model capacity of-

*These authors contributed equally to this work.
†The corresponding authors for this research are: zhengzhu@ieee.org, tianrun.chen@kokoni3d.com, mei@pku.edu.cn.

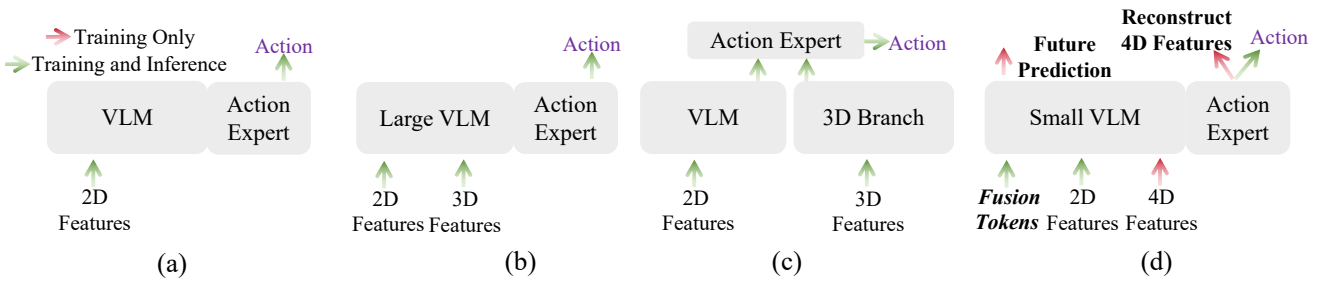


Figure 2. (a) Using only 2D features as input to the VLM [6, 25], which results in limited spatiotemporal awareness. (b) Direct fusion approaches combine spatial and 2D features within large VLMs [4, 37, 89]. (c) Decoupled designs that introduce a dedicated spatial branch [27, 56], causing large parameter overhead. (d) SwiftVLA leverages a pretrained model [95] to extract 4D features and applies a feature reconstruction objective to align 4D and 2D representations. In addition, *Fusion Tokens* and a future prediction objective are introduced to strengthen cross-modal integration. The 4D inputs and auxiliary heads are removed at inference to maintain efficiency.

ten weakens reasoning ability, making it difficult to capture the 3D spatial information that is crucial for VLAs to plan precise actions, leading to poor localisation and imprecise trajectories and lowering task success rates. As shown in Fig. 1, smaller VLM models such as SmolVLM-0.5B [45] significantly underperform in spatial reasoning tasks, such as answering the question "What color is the bowl on the far left?" compared to larger VLM models [3]. Therefore, while SmolVLA [45], based on SmolVLM-0.5B [45], exhibits significantly faster inference speeds than π_0 [6] based on PaliGemma-3B [3], its task success rate is notably lower, as complex manipulation tasks often require stronger spatiotemporal reasoning and scene understanding capabilities.

Therefore, recent works [4, 27, 37, 56, 89] have explored integrating 3D and 4D information [13, 16, 17, 21, 42, 43, 46, 48, 69, 70, 75, 87, 88] to enhance VLAs' perception of complex environments. However, existing fusion approaches are still suboptimal for lightweight architectures. As shown in Fig. 2 (b), some methods [4, 34, 37, 51, 81, 89] directly fuse 3D features with 2D representations within a large VLM. While this improves spatial awareness compared to Fig. 2 (a), which uses only 2D input, it has to rely on heavyweight VLMs to handle cross-modal fusion. To mitigate this dependency, other approaches [27, 56] (Fig. 2 (c)) decouple 3D processing from the VLM by introducing an additional branch. However, this design substantially increases parameter overhead, making it unsuitable for compact models. In summary, as shown in Fig. 2 (a–c), existing approaches still fall short of effectively balancing the lightweight design of VLAs with the practical need for robust and reliable spatiotemporal perception.

In this paper, we present SwiftVLA, a lightweight VLA model built upon a compact VLM [45], which incorporates 4D spatiotemporal information with minimal computational cost. As shown in Fig. 2 (d), SwiftVLA takes 4D representations as auxiliary inputs and employs a reconstruction objective to learn spatiotemporal dynamics from 4D features, enabling the model to discard them during inference while maintaining performance comparable to full 4D inputs. Meanwhile, *Fusion Tokens* are introduced and supervised by a future prediction objective to promote effective cross-modal fusion. Specifically, SwiftVLA integrates a pretrained 4D visual geometry transformer [95] with a temporal cache to convert streaming frames into 4D features incrementally. The cache enables feature reuse across frames and provides temporal context. Meanwhile, because 4D cues are derived directly from standard visual inputs, no additional sensors such as depth cameras or LiDAR are required. For efficient fusion of 2D and 4D features in a compact VLM, we introduce learnable *Fusion Tokens* to unify representations across modalities. Their outputs are supervised by the robot end-effector's future trajectory to encourage task-relevant learning. Finally, we propose a mask-and-reconstruct strategy, where during training, SwiftVLA randomly masks either the 2D or 4D modality with a certain probability and requires the action expert to reconstruct the masked features, which encourages the learning of geometry- and dynamics-aware representations. This enables the model to achieve performance comparable to that with 4D inputs during inference, even without them, minimizing the overhead of 4D inputs while preserving spatiotemporal modeling capability.

We validate SwiftVLA with experiments on simulated and real-world environments. SwiftVLA outperforms lightweight baselines and matches the performance of VLA models up to 7× larger. On edge devices, it maintains comparable performance while running 18× faster than the state-of-the-art baseline π_0 and reducing memory by 12×.

The main contributions of this paper are summarized as follows:

- We propose SwiftVLA, a method that integrates 4D spatiotemporal information into a lightweight VLA model at minimal cost. SwiftVLA extracts 4D features and adopts a mask-and-reconstruct training strategy that distills 4D knowledge into the VLA. This enables the model to maintain comparable performance to that with 4D inputs dur-

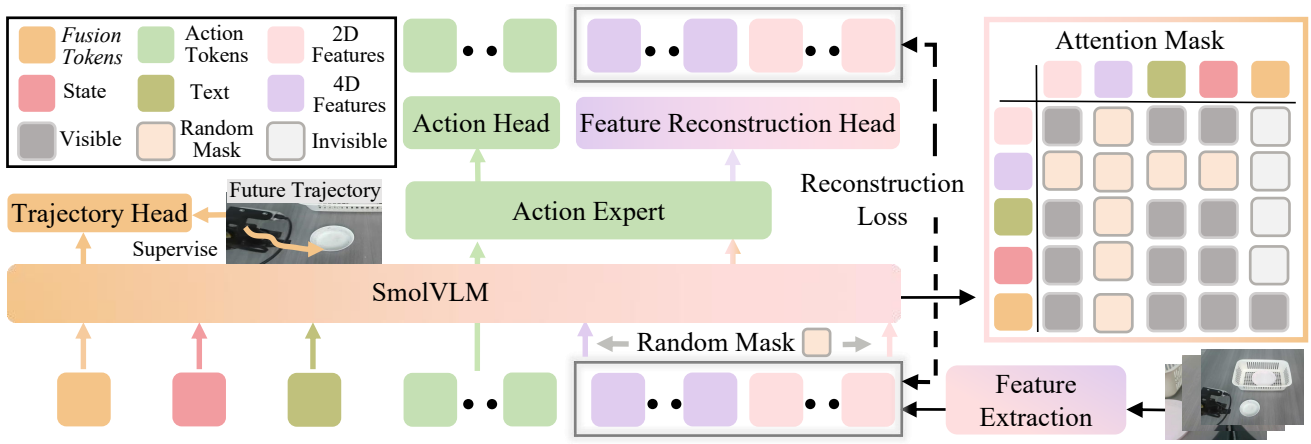


Figure 3. The pipeline of the SwiftVLA. We first extract 2D and 4D features from input images. A lightweight VLM [45] processes 2D and 4D features with *Fusion Tokens* to achieve cross-modal integration. The outputs of the *Fusion Tokens* are supervised by the robot end-effector's future trajectory. During training, we randomly mask either the 2D or the 4D features, and we require the action expert to reconstruct the masked features while learning to generate actions. We show the attention mask under random masking of the 4D features. In this case, 4D features are excluded from the VLM attention, and the model is required to reconstruct the 4D features from the others.

ing inference while requiring only 2D inputs.

- We fuse 2D and 4D features in a lightweight VLM via learnable *Fusion Tokens*, trained with supervision from the robot arm's future end-effector trajectory to produce a unified, action-aware representation.
- Extensive experiments in simulation and on real robots demonstrate that SwiftVLA achieves performance comparable to a baseline that is 7× larger. On edge devices, it runs 18× faster and uses 12× less memory than π_0.

2. Related Work

2.1. Lightweight VLA Models

Recent advances in VLA models integrate VLM backbones with action modules through end-to-end training. OpenVLA [25] introduced a 7B parameter model trained on public datasets [24, 61] to generate discrete action tokens. To overcome the limitations of tokenizing actions in continuous control, π_0 [6] uses a diffusion-based decoder to directly generate continuous actions. However, these models [6, 25] have a large number of parameters, leading to high training costs and significant inference latency.

To address this, several approaches have shifted toward lighter VLA designs. Based on OpenVLA [25], MiniVLA [2] replaces the backbone with a smaller model [1], thereby reducing the total size to 1B parameters. TinyVLA [71] introduces a diffusion policy decoder that directly generates continuous multi-step action sequences to avoid the high latency of autoregressive generation, and employs LoRA [22] for parameter-efficient fine-tuning. To further lighten the model, SmolVLA [55] uses pixel shuffle to limit each frame's tokens and skips a subset of VLM layers, ultimately compressing the parameter count to around 0.5B.

However, to achieve model lightweighting, these methods typically rely on shrinking the backbone parameters, which leads to a degradation in the VLA model's spatial reasoning and fine-grained control capabilities.

2.2. 3D Perception in VLA Models

3D perception [11, 15, 19, 21, 40, 41, 52, 53, 62, 68, 74, 78, 85, 86] is crucial for enhancing robotic manipulation capabilities. Recent studies have attempted to directly incorporate 3D features into the VLM to enhance their geometric awareness, as shown in Fig. 2 (b). 3D-VLA [89] extracts spatial embeddings and encodes them into VLM embeddings to improve spatio-temporal reasoning. SpatialVLA [51] introduces 3D positional encoding and an adaptive action network into VLMs to improve spatial understanding. Evo-0 [37] obtains 3D features by leveraging VGGT [63], injecting the 3D geometric features into the VLA. However, the domain gap between 2D pixels and 3D geometry is substantial, and directly injecting both into VLMs often requires larger VLMs for better alignment and fusion. Some approaches [9, 14, 94] also attempt to fine-tune VLMs for spatio-temporal reasoning, but this often relies on massive amounts of temporally annotated data, which are expensive to collect.

Therefore, as shown in Fig. 2 (c), some methods adopt decoupled designs that introduce a spatial branch. PointVLA [27] treats point clouds as auxiliary conditioning signals and decouples 3D processing from the 2D vision encoder, enabling the model to leverage geometric cues while preserving the integrity of pretrained 2D representations. GeoVLA [56] adopts parallel branches for multimodal inputs and leverages modality-specific experts to achieve fusion. However, these approaches focus only on 3D informa-

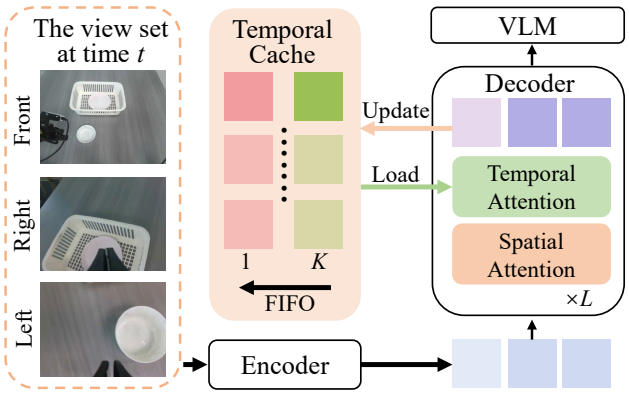


Figure 4. The process of 4D feature extraction. At each step, we sequentially process multi-view observations and load contextual information from the cache for temporal attention. The generated 4D features are updated to the cache and delivered to the VLM.

tion, neglecting temporal dynamics, while increasing memory footprint and inference latency. More recently, 4D-VLA [81] incorporates the temporal dimension into VLA modeling by leveraging a history-similarity-based keyframe sampling strategy and generating 3D-aware spatial-visual tokens. While this method enhances spatiotemporal perception, sampling multiple frames introduces additional inference overhead. In contrast, SwiftVLA maintains a lightweight design while injecting 4D cues at lower cost.

3. Method

3.1. Model architecture

As shown in Fig. 3, SwiftVLA consists of two connected components: a pretrained lightweight VLM [45] and an action expert. The VLM processes the input to extract both 2D and 4D features with a pretrained 4D visual geometry transformer [95]. Meanwhile, *Fusion Tokens* are introduced into the VLM to better leverage both 2D and 4D features, supervised by end-effector trajectory prediction. Then, the VLM fuses the 2D and 4D features, *Fusion Tokens*, and other inputs to generate the intermediate hidden states, which serve as the condition for the action expert model to perform action prediction. Additionally, during training, we employ a mask-and-reconstruction strategy in which the 2D or 4D features are masked so that they do not contribute to action generation, and the action expert is required to reconstruct the corresponding features. This strategy encourages the model to exploit cross-modal cues and enables the removal of 4D feature inputs during inference with minimal performance loss, thereby maintaining a lightweight design.

Specifically, at each time step t, given the ordered view set $S = [left, right, front]$, the robot receives a natural language instruction l, multi-view observations $o_t = \{o_t^v\}_{v \in S}$, and the proprioceptive state s_t. From the raw images, we first extract 2D visual features from each view using an image encoder [80], obtaining $F_{2D}^t = \{F_{2D}^{t,v}\}_{v \in S}$. Then, a pretrained 4D visual geometry transformer [95] is applied to derive temporally-enhanced 4D features F_{4D}^t. A set of learnable *Fusion Tokens* Q_f are then introduced to interact with F_{2D}^t, F_{4D}^t, with proprioceptive embeddings E_s^t and language embeddings E_l^t through a lightweight VLM $\mathcal{V}$, resulting in a unified representation:

$$Z_f^t = \mathcal{V}(Q_f, E_s^t, E_l^t, F_{4D}^t, F_{2D}^t). \tag{1}$$

In the fused representation Z_f^t, the portion corresponding to the *Fusion Tokens* is decoded to predict the end-effector trajectory, with explicit supervision from ground-truth trajectories, thereby enabling the intermediate hidden states of $\mathcal{V}$ to learn trajectory-aware cross-modal alignment. In parallel, the intermediate hidden states of $\mathcal{V}$, denoted as $\{h_{\mathcal{V}}^{(i)}\}$, are employed as hierarchical conditional features for the action expert $\mathcal{A}$. The action expert is formulated as a conditional diffusion model. Given a noise sample ϵ and conditioned on VLM features $\{h_{\mathcal{V}}^{(i)}\}$, it produces an action latent:

$$Z_{\mathcal{A}}^t = \mathcal{A}\left(\epsilon \,\middle|\, \{h_{\mathcal{V}}^{(i)}\}\right). \tag{2}$$

We decode $Z_{\mathcal{A}}^t$ with two complementary heads: the first predicts the diffusion noise for the action, and the second reconstructs the masked feature representations. The reconstruction head is trained with an auxiliary objective to improve cross-modal alignment and is discarded at inference time, ensuring a lightweight design and fast inference.

3.2. Incremental 4D Feature Extraction

Recent studies [4, 18, 27, 33, 37, 56, 89] show that integrating 3D information [47, 63, 65, 95], such as depth maps and point clouds, can improve the spatial perception of VLA models. Such signals provide geometric cues that help models better reason about distance and object layout. Yet these methods often rely on additional sensors and underutilize VLAs' temporal modeling capacity, resulting in limited spatio-temporal reasoning.

Therefore, as shown in Fig. 4, we adopt a pretrained 4D visual geometry transformer [95] consisting of an encoder, a decoder, and a temporal cache module, whose weights are kept frozen, that incrementally extracts 4D features from input images, avoiding reliance on extra sensors and leveraging spatial and temporal cues. At each time step t, for each view $v \in S$, the observation o_t^v is encoded into a feature embedding using the encoder:

$$F_e^{t,v} = Encoder(o_t^v). \tag{3}$$

We sequentially feed the encoded features into a spatiotemporal decoder, which employs spatial and temporal attention to extract 4D representations following the view order S. During the temporal attention stage, the current feature

embedding interacts with the temporal cache through cross-attention, enabling the integration of temporal context. The temporal cache is then updated iteratively to generate 4D features in a consistent view order. We initialize the temporal cache as $C^{t,0} = C^{t-1}$. For each $k \in \{1, 2, 3\}$ with the corresponding view $v = S_k$, the decoding process is formulated as:

$$(F_{4D}^{t,v},\, C^{t,k}) = \mathrm{Decoder}\big(\mathrm{CrossAttn}(F_e^{t,v},\, C^{t,k-1})\big). \quad (4)$$

After all views have been processed, the temporal cache is updated as $C^t = C^{t,3}$. To maintain temporal efficiency, we adopt a First-In-First-Out (FIFO) strategy for the cache, retaining only the most recent K representations of the 4D features. Finally, the visual inputs to the VLM are defined as follows:

$$F_{2D}^t = \{F_{2D}^{t,v}\}_{v \in S}, \qquad F_{4D}^t = F_{4D}^{t,front}. \quad (5)$$

The $F_{4D}^{t,left}$ and $F_{4D}^{t,right}$ are only used to update the temporal cache to provide a more comprehensive spatiotemporal context and are not provided to the VLM as 4D features, considering the training cost.

3.3. Fusion Tokens

Many prior works [27, 37, 51, 89] attempt to feed 3D information into VLMs so that they can jointly understand 2D and 3D inputs. However, these approaches typically rely on heavyweight VLMs. Lightweight VLMs struggle to develop robust spatial reasoning and cannot effectively fuse multimodal inputs into a coherent, 3D-aware latent space.

To address this problem, we introduce *Fusion Tokens*, a set of learnable tokens that interact with 2D features and 4D spatiotemporal features, and are directly supervised by the end-effector's future trajectory. The keys and values produced by the *Fusion Tokens*, together with those from other tokens, form the conditioning signal $h_{\mathcal{V}}^{(i)}$ used by the action expert to generate action chunks. Specifically, Fusion Tokens interact with the aggregated multimodal token sequence composed of the 2D features F_{2D}^t, the 4D features F_{4D}^t, the language embeddings E_l^t, and the state embeddings E_s^t through cross-attention within the VLM $\mathcal{V}$, yielding a fused representation Z_f^t, as shown in Eq. 1. The fused representation Z_f^t serves as the perception output and is optimized via end-effector trajectory prediction:

$$\hat{\tau}_t = h_{traj}\big(Z_f^t\big), \qquad \mathcal{L}_{traj} = \|\hat{\tau}_t - \tau_t\|_2^2, \quad (6)$$

where h_{traj} is a predictor that decodes the trajectory, $\hat{\tau}_t$ is the predicted trajectory, and τ_t is the ground-truth. Using the end-effector's future trajectory as supervision, *Fusion Tokens* align multimodal features with spatiotemporal semantics, making $h_{\mathcal{V}}^{(i)}$ more effective for action generation.

3.4. Mask and Reconstruct Strategy

Although incorporating 4D features with *Fusion Tokens* into VLAs can significantly enhance their spatial reasoning capabilities, the resulting increase in parameters and computational overhead runs counter to the lightweight design principle of VLAs. Therefore, we propose a mask-and-reconstruction strategy that leverages 4D supervision signals during training to build geometry-aware representations, while discarding 4D feature inputs during inference to maintain model efficiency with minimal performance degradation. Our approach encourages the model to build geometry-aware representations through structured masking and reconstruction, thereby distilling rich spatial and temporal knowledge into the learned features. Next, we present the training and inference procedures in detail.

Training. During training, we employ a random masking strategy that, with a certain probability, applies masks to either the 2D or 4D features. Under this setting, the VLA is required to predict actions based on the remaining modalities, while simultaneously reconstructing the masked features. As illustrated in Fig. 3, we visualize the masking operation applied to the 4D features during training. The gray and white blocks indicate the fixed visible and invisible tokens, respectively, while the pink blocks represent tokens that undergo random masking, turning originally visible tokens into invisible ones. Meanwhile, the reconstruction losses are defined as follows:

$$\begin{aligned} \mathcal{L}_{2D} &= \big\|h_{2D}\big(Z_{\mathcal{A}}^t\big) - F_{2D}^t\big\|_2, \\ \mathcal{L}_{4D} &= \big\|h_{4D}\big(Z_{\mathcal{A}}^t\big) - F_{4D}^t\big\|_2. \end{aligned} \quad (7)$$

where F_t^{2D} and F_t^{4D} denote the inputs to the VLM, h_{2D} and h_{4D} are the feature reconstruction heads, and $Z_{\mathcal{A}}^t$ is the action latent produced by the action module $\mathcal{A}$ (see Eq. 2). At the same time, $Z_{\mathcal{A}}^t$ is fed into the action prediction head to predict the diffusion noise:

$$\mathcal{L}_{action} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)}\Big[\big\|h_{action}\big(Z_{\mathcal{A}}^t\big) - \epsilon\big\|_2^2\Big], \quad (8)$$

where $\epsilon \sim \mathcal{N}(0, I)$ denotes the forward-process noise, and h_{action} is the action prediction head. Finally, the total loss for SwiftVLA is a weighted sum of the reconstruction, action prediction, and trajectory objectives:

$$\begin{aligned} \mathcal{L}_{total} = \lambda_{2D}\,\mathcal{L}_{2D} + \lambda_{4D}\,\mathcal{L}_{4D} \\ + \lambda_{action}\,\mathcal{L}_{action} + \lambda_{traj}\,\mathcal{L}_{traj}, \end{aligned} \quad (9)$$

Where each λ serves as a balancing coefficient among the objectives. This design encourages the model to learn more comprehensive and geometry-aware 4D representations, rather than relying on a single modality for action prediction. Meanwhile, this mechanism enables the model to implicitly reconstruct and reason over 4D spatial structures even when explicit 4D feature inputs are unavailable.

Method	Short-Horizon		Medium-Horizon		Long-Horizon		Average	
	SR ↑	Length ↓	SR ↑	Length ↓	SR ↑	Length ↓	SR ↑	Length ↓
π_0 [6]	<u>0.42</u>	120	0.46	<u>150</u>	0.52	187	0.47	152
GO-1 [7]	0.40	124	0.44	160	0.54	190	0.46	158
TinyVLA [71]	0.08	183	0.08	240	0.06	236	0.07	220
SmolVLA [55]	0.28	152	0.32	178	0.28	234	0.29	188
SmolVLA† [55]	0.38	130	0.36	165	0.34	195	0.36	163
SwiftVLA	**0.56**	<u>115</u>	<u>0.48</u>	156	<u>0.56</u>	**180**	<u>0.53</u>	<u>150</u>
SwiftVLA with 4D input	**0.56**	**100**	**0.50**	**145**	**0.58**	<u>185</u>	**0.55**	**143**

Table 1. Comparison of task success rate and average trajectory length in simulation. The best results are marked in **bold**, and the second-best results are <u>underlined</u>. † denotes the model that is pre-trained and fine-tuned using the same configuration as SwiftVLA.

Inference. To further reduce the overall parameter count and facilitate deployment on edge platforms, we retain only the 2D feature branch during inference. In this stage, the 4D feature extractor, reconstruction heads, and trajectory head are removed, as they are only used for auxiliary supervision during training. Consequently, the deployed model consists solely of the VLM and the action expert, forming a compact yet effective architecture. The total parameter count of the deployed model equals the sum of these two components. Despite its lightweight nature, this design preserves the strong spatiotemporal perception capability learned through masked training, enabling efficient and reliable deployment on real-world robotic platforms.

4. Experiments

4.1. Experimental Setup

Evaluation Metrics. We employ success rate (SR) primarily as our evaluation metric, along with the average trajectory length. In simulations, a task receives an SR of 1 for successful completion and 0 otherwise. For real-world evaluations, we use a detailed scoring system, where in the pick-and-place task, a score of 0.5 is given for grasping the object and another 0.5 for placing it at the target location.

Baselines. We primarily selected VLA models of different parameter sizes as baselines for comparison with SwiftVLA. For large models, we chose the current state-of-the-art model π_0 [6] and GO-1 [7]. For smaller models, we selected TinyVLA [71] and SmolVLA [55]. For SwiftVLA, we adopted two inference configurations: one that uses 4D inputs during inference, referred to as SwiftVLA with 4D input, and another that does not use 4D inputs during inference, referred to simply as SwiftVLA. Both configurations share the same set of trained weights. Additionally, in the LIBERO benchmark [38], we compared several other algorithms, grouped into three categories: spatio-temporal enhanced VLA models [34, 51, 81], which utilize 3D or 4D inputs; small VLA models [20, 55, 90], which employ smaller VLMs; and large VLA models [5, 6, 25, 26, 36, 50, 66], which refer to VLA models with more than 3B parameters.

Implementation Details. We adopt SmolVLM [45] as the backbone. The complete model comprises approximately 450 million parameters, of which around 100 million are allocated to the action-expert module. Meanwhile, we pre-train our model on public datasets [7, 72] using a two-stage training procedure (detailed in the appendix).

4.2. Simulation Benchmark Experiments

Simulation Setup. We conduct a systematic evaluation of SwiftVLA on both the RoboTwin 2.0 [10] and LIBERO benchmarks [38]. For RoboTwin 2.0, our experimental setup considers three categories of tasks: Short-Horizon, Medium-Horizon, and Long-Horizon, with two subtasks selected for each category. For each subtask, we generate 50 demonstration trajectories, which are then used for post-training. For LIBERO benchmark, we perform experiments across four task suites: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long.

Results. As shown in Tab. 1, the mask-and-reconstruct strategy enables SwiftVLA, which does not use 4D input during inference, to maintain competitiveness comparable to that of SwiftVLA with 4D input. Moreover, SwiftVLA demonstrates strong performance across all three task categories, exhibiting competitive capabilities compared to π_0 [6] while using only about 15% of its parameters. Meanwhile, TinyVLA [71] and SmolVLA [55] still lag notably behind π_0 [6], mainly due to their smaller VLM backbones, which lack sufficient capacity to model long-term spatiotemporal dependencies. In contrast, SwiftVLA introduces 4D representations to enhance its spatiotemporal understanding, improving SR by 82.76% over SmolVLA [55].

As shown in Tab. 3, we compare SwiftVLA with three categories of methods on LIBERO. For spatio-temporal enhanced VLA [34, 51, 81], methods like SpatialVLA [51] and 4D-VLA [81] directly integrate 3D and 4D information into the VLM. However, their performance gains are often limited by the model's ability to handle multiple modalities. Approaches such as Qdepth-VLA [34] add additional 3D processing branches, improving performance but increasing the model size to over 3B parameters. Smaller models [20, 55], despite having fewer than 2B parameters, typically show lower success rates. Large VLA models offer

Methods	Clean the desk	Throw the bottle	Stack Bowls	Average
π_0 [6]	0.60	0.66	0.56	0.61
SmolVLA [55]	0.32	0.40	0.30	0.34
SmolVLA† [55]	0.52	0.54	0.52	0.53
SwiftVLA	**0.86**	0.80	0.74	0.80
SwiftVLA with 4D input	**0.86**	**0.82**	**0.78**	**0.82**

Table 2. Comparison of task success rates in real-world experiments. The best results are marked in **bold**, and the second-best results are underlined. † denotes the model that is pre-trained and fine-tuned using the same configuration as SwiftVLA.

Methods	Size	LIBERO				
		Spatial	Object	Goal	Long	Avg
Spatio-Temporal Enhanced VLA						
SpatialVLA [51]	4B	88.2	89.9	78.6	55.5	78.1
4D-VLA [81]	4B	88.9	95.2	90.9	79.1	88.6
QDepth-VLA [34]	4B	**97.6**	96.6	95.2	90.0	94.9
Small VLA						
SmolVLA [55]	**0.45B**	90.0	96.0	92.0	71.0	87.3
SmolVLA† [55]	**0.45B**	93.5	96.5	95.4	83.4	92.2
UniAct [90]	0.5B	77.0	87.0	77.0	70.0	77.8
VLA-OS [20]	0.5B	87.0	96.5	92.7	66.0	85.6
SmolVLA [55]	2B	93.0	94.0	91.0	77.0	88.8
Large VLA						
GR00T-N1 [5]	3B	94.4	97.6	93.0	90.6	93.9
π_0 [6]	3B	96.8	**98.8**	95.8	85.2	94.1
π_0+FAST [50]	3B	96.4	96.8	88.6	60.2	85.5
OpenVLA [25]	7B	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT [26]	7B	**97.6**	98.4	**97.9**	**94.5**	**97.1**
DD-VLA [36]	7B	97.2	98.6	97.4	92.0	96.3
UniVLA [66]	9B	95.4	**98.8**	93.6	94.0	95.4
Spatio-Temporal Enhanced Small VLA						
SwiftVLA	**0.45B**	97.0	96.4	96.8	88.4	94.7
SwiftVLA with 4D input	1.65B	97.2	96.8	97.4	89.0	95.1

Table 3. Comparison of methods on the LIBERO. The best results are marked in **bold**, and the second-best results are underlined. † denotes the model that is pre-trained and fine-tuned using the same configuration as SwiftVLA.

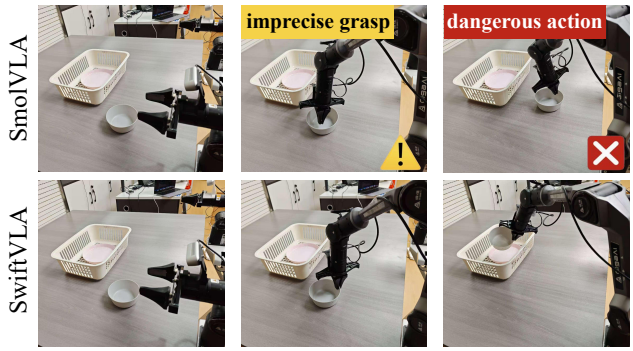


Figure 5. Comparison of SmolVLA and SwiftVLA under identical initial poses. During execution, SmolVLA fails to grasp accurately, as the end-effector misses the target and collides with the object, causing it to shift and posing safety risks. In contrast, SwiftVLA successfully completed the grasp with accurate positioning and stable control, demonstrating superior performance.

Methods	Inference Time (s)	Memory (MB)	Average SR
π_0 [6]	2.966	16236.2	0.48
SmolVLA [55]	0.166	1397.5	0.30
SwiftVLA	0.167	1398.4	0.76

Table 4. Comparison of inference time, memory usage, and SR across models on the NVIDIA Jetson Orin [49].

strong performance but also require over 3B parameters, making them costly for real-world deployment. In contrast, SwiftVLA effectively leverages 4D information while maintaining a compact design, achieving performance comparable to large VLA models.

4.3. Real-World Experiment

To evaluate the effectiveness of the method in the real world, we conducted a gripper grasping experiment using the AgileX PiPER six-degree-of-freedom robotic arm, with computational support provided by an NVIDIA RTX 4090 GPU. In addition, we designed a series of real-world tasks, including Clean the Desk, Throw the Bottle, and Stack Bowls, with detailed descriptions provided in the supplementary material. As summarized in Tab. 2, our method demonstrates strong performance compared to π_0 [6] while using fewer parameters, and significantly outperforms similarly sized baselines such as SmolVLA. As shown in Fig. 5, we compared SmolVLA and SwiftVLA under identical initial object placements. During execution, SmolVLA failed to achieve precise grasping due to its limited understanding of geometric information. Meanwhile, the end-effector collided with the target object and displaced it, which could lead to task failure or safety hazards. In contrast, SwiftVLA successfully performed stable and accurate grasping thanks to its superior spatial perception and control.

4.4. Deployment on the Edge Device

To evaluate the deployment efficiency of VLAs on edge devices, we adopt the NVIDIA Jetson Orin [49] as our target platform. As summarized in Tab. 4, we report the inference time and parameter count of each model. The results show that SwiftVLA delivers an $18\times$ speedup in inference compared to the larger VLA model π_0, while reducing

Input Feature Type	*Fusion tokens*	Average SR
2D		0.36
2D & 4D		<u>0.40</u>
2D & 4D	✓	**0.50**

Table 5. Ablation study of SwiftVLA with 4D input evaluated by SR. The input feature type indicates the modalities used during training and inference. The best results are marked in **bold**, and the second-best results are <u>underlined</u>.

memory consumption by 12×. Moreover, its inference latency is nearly identical to that of SmolVLA. Meanwhile, SwiftVLA demonstrates a high success rate compared to π_0 on the NVIDIA Jetson Orin [49], indicating its suitability.

4.5. Ablation Study

In this section, we perform experiments on the RoboTwin 2.0 [10] platform to address the following questions.

Q1. How do 4D features affect task success? We compare two settings that use 2D inputs and that combine 2D inputs with the 4D features, as shown in the first and second rows of Tab. 5. The results indicate that relying solely on 2D inputs leads to a lower SR. Incorporating the 4D features yields a substantial improvement, suggesting that 4D features provide stronger representations for action planning.

Q2: What role do *Fusion Tokens* play? *Fusion Tokens* are designed to integrate 4D and 2D features, using the 2D end-effector trajectory as supervision for trajectory prediction. In the second and third rows of Tab. 5, we compare models with and without *Fusion Tokens* and observe significant improvements when the token is enabled. This is because small models struggle to fully leverage the input 4D information. The introduction of *Fusion Tokens*, along with the design of a target task, helps guide the model to effectively use both 2D and 4D cues, leading to improved cross-modal alignment and more effective temporal cue utilization.

Q3. What is the effect of the mask-and-reconstruction strategy? We employ a mask-and-reconstruction strategy during training, where 2D or 4D features are randomly dropped with a fixed probability, and the VLA is tasked with reconstructing the masked features. The aim is to enable the model to maintain performance comparable to full 4D input, even when 4D information is missing during inference. As shown in Tab. 6, we compare different training strategies and evaluate performance under both inference with and without 4D input. The results show that directly removing the 4D input during inference, without applying any strategy, leads to a significant performance drop, as the model becomes overly dependent on 4D cues for prediction. Introducing 4D feature masking alleviates this dependency and preserves part of the performance when 4D input is unavailable. Moreover, incorporating the feature reconstruction helps distill 4D information into the VLA during training, allowing the model to achieve performance comparable to that with full 4D input, even in the absence of 4D features during inference. In addition, we find that moderately masking 2D features encourages the model to better exploit underlying 4D geometric cues and enhances cross-modal consistency, as reflected in the last column of Tab. 6.

4D Feature Mask	2D Feature Mask	Feature Reconstruction	SwiftVLA	SwiftVLA with 4D Input
			0.02	0.50
✓			0.40	0.48
✓		✓	<u>0.50</u>	<u>0.52</u>
✓	✓	✓	**0.53**	**0.55**

Table 6. Comparison of SR across different training strategies. All models are trained with both 2D and 4D feature inputs, while only 2D features are used during inference. The best results are marked in **bold**, and the second-best results are <u>underlined</u>.

Size	SwiftVLA	SwiftVLA with 4D input
K=3	0.47	0.49
K=4	0.48	<u>0.52</u>
K=5	0.50	0.51
K=6	<u>0.52</u>	**0.55**
Random	**0.53**	**0.55**

Table 7. Ablation study on the temporal cache size evaluated by SR. All models are trained with both 2D and 4D feature inputs, while only 2D features are used during inference. The best results are marked in **bold**, and the second-best results are <u>underlined</u>.

Q4. How does the cached memory size K affect performance? We analyze how the cached memory size K chosen during training affects model performance. We evaluate four fixed settings with $K \in \{3, 4, 5, 6\}$ and a randomized strategy that samples $K \in \{3, 4, 5, 6\}$ at each training step. As shown in Tab. 7, the randomized strategy outperforms all fixed-length baselines, indicating that exposure to variable temporal horizons process notably enhances adaptability.

5. Conclusion

In this paper, we present SwiftVLA, a lightweight framework that achieves strong spatiotemporal reasoning while maintaining design efficiency. Specifically, we employ a 4D visual geometry transformer with a temporal cache that extracts 4D features and integrates them into the VLM to enhance both spatial and temporal modeling. To bridge the gap between 2D and 4D features, we introduce *Fusion Tokens*, whose representations are supervised by the future trajectory of the end-effector, effectively capturing integrated multimodal information. Additionally, we employ a mask-and-reconstruct strategy to distill 4D knowledge into the VLA, while minimizing performance degradation when 4D inputs are omitted during inference. Experiments show that SwiftVLA matches the performance of models with up to 7× more parameters, while offering up to 18× faster inference and 12× smaller memory footprint on edge devices.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 3

[2] Suneel Belkhale and Dorsa Sadigh. Minivla: A better vla with a smaller footprint, 2024. 1, 3

[3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 1, 2

[4] Vineet Bhat, Yu-Hsiang Lan, Prashanth Krishnamurthy, Ramesh Karri, and Farshad Khorrami. 3d cavla: Leveraging depth and 3d context to generalize vision language action models for unseen tasks. *arXiv preprint arXiv:2505.05800*, 2025. 2, 4

[5] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 6, 7

[6] 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. 1, 2, 3, 6, 7

[7] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. 6

[8] Yifan Chang, Jie Qin, Limeng Qiao, Xiaofeng Wang, Zheng Zhu, Lin Ma, and Xingang Wang. Scalable training for vector-quantized networks with 100% codebook utilization. *arXiv preprint arXiv:2509.10140*, 2025. 1

[9] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024. 3

[10] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. 6, 8

[11] Zhiwei Chen, Yupeng Hu, Zixu Li, Zhiheng Fu, Haokun Wen, and Weili Guan. Hud: Hierarchical uncertainty-aware disambiguation network for composed video retrieval. In *Proceedings of the ACM International Conference on Multimedia*, page 6143–6152, 2025. 3

[12] Zhangquan Chen, Xufang Luo, and Dongsheng Li. Visrl: Intention-driven visual perception via reinforced reasoning. *arXiv preprint arXiv:2503.07523*, 2025. 1

[13] Zhangquan Chen, Manyuan Zhang, Xinlei Yu, Xufang Luo, Mingze Sun, Zihao Pan, Yan Feng, Peng Pei, Xunliang Cai, and Ruqi Huang. Think with 3d: Geometric imagination grounded spatial reasoning from limited views. *arXiv preprint arXiv:2510.18632*, 2025. 2

[14] An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093, 2024. 3

[15] Yubo Cui, Zheng Fang, Jiayao Shan, Zuoxu Gu, and Sifan Zhou. 3d object tracking with transformer. *arXiv preprint arXiv:2110.14921*, 2021. 3

[16] Tianqi Ding, Dawei Xiang, Pablo Rivas, and Liang Dong. Neural pruning for 3d scene reconstruction: Efficient nerf acceleration. *arXiv preprint arXiv:2504.00950*, 2025. 2

[17] Tianqi Kirk Ding, Dawei Xiang, Yijiashun Qi, Ze Yang, Zunduo Zhao, Tianyao Sun, Pengbin Feng, and Haoyu Wang. Nerf-based defect detection. In *International Conference on Remote Sensing, Mapping, and Image Processing (RSMIP 2025)*, pages 368–373. SPIE, 2025. 2

[18] Yicheng Feng, Wanpeng Zhang, Ye Wang, Hao Luo, Haoqi Yuan, Sipeng Zheng, and Zongqing Lu. Spatial-aware vla pretraining through visual-physical alignment from human videos. *arXiv preprint arXiv:2512.13080*, 2025. 4

[19] Zhiheng Fu, Zixu Li, Zhiwei Chen, Chunxiao Wang, Xuemeng Song, Yupeng Hu, and Liqiang Nie. Pair: Complementarity-guided disentanglement for composed image retrieval. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE, 2025. 3

[20] Chongkai Gao, Zixuan Liu, Zhenghao Chi, Junshan Huang, Xin Fei, Yiwen Hou, Yuxuan Zhang, Yudi Lin, Zhirui Fang, Zeyu Jiang, et al. Vla-os: Structuring and dissecting planning representations and paradigms in vision-language-action models. *arXiv preprint arXiv:2506.17561*, 2025. 6, 7

[21] Tianhao Guo, Bingjie Lu, Feng Wang, and Zhengyang Lu. Depth-aware super-resolution via distance-adaptive variational formulation. *arXiv preprint arXiv:2509.05746*, 2025. 2, 3

[22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[23] 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. 1

[24] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 3

[25] 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, 6, 7

[26] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025. 6, 7

[27] Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *arXiv preprint arXiv:2503.07511*, 2025. 2, 3, 4, 5

[28] Haoyun Li, Ivan Zhang, Runqi Ouyang, Xiaofeng Wang, Zheng Zhu, Zhiqin Yang, Zhentao Zhang, Boyuan Wang, Chaojun Ni, Wenkang Qin, et al. Mimicdreamer: Aligning human and robot demonstrations for scalable vla training. *arXiv preprint arXiv:2509.22199*, 2025. 1

[29] Jianxiong Li, Zhihao Wang, Jinliang Zheng, Xiaoai Zhou, Guanming Wang, Guanglu Song, Yu Liu, Jingjing Liu, Ya-Qin Zhang, Junzhi Yu, et al. Robo-mutual: Robotic multimodal task specification via unimodal learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4182–4189. IEEE, 2025.

[30] Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. Bridgevla: Input-output alignment for efficient 3d manipulation learning with vision-language models. *arXiv preprint arXiv:2506.07961*, 2025.

[31] Wei Li, Bing Hu, Rui Shao, Leyang Shen, and Liqiang Nie. Lion-fs: Fast & slow video-language thinker as online video assistant. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3240–3251, 2025.

[32] Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. Cogvla: Cognition-aligned vision-language-action model via instruction-driven routing & sparsification. *arXiv preprint arXiv:2508.21046*, 2025. 1

[33] Xiaoqi Li, Liang Heng, Jiaming Liu, Yan Shen, Chenyang Gu, Zhuoyang Liu, Hao Chen, Nuowei Han, Renrui Zhang, Hao Tang, et al. 3ds-vla: A 3d spatial-aware vision language action model for robust multi-task manipulation. In *9th Annual Conference on Robot Learning*, 2025. 4

[34] Yixuan Li, Yuhui Chen, Mingcai Zhou, and Haoran Li. Qdepth-vla: Quantized depth prediction as auxiliary supervision for vision-language-action models. *arXiv preprint arXiv:2510.14836*, 2025. 2, 6, 7

[35] Zhenglin Li, Yangchen Huang, Mengran Zhu, Jingyu Zhang, JingHao Chang, and Houze Liu. Automated adaptive navigation leveraging natural language models in aerospace robotics system. In *2024 6th International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI)*, pages 250–254. IEEE, 2024. 1

[36] Zhixuan Liang, Yizhuo Li, Tianshuo Yang, Chengyue Wu, Sitong Mao, Liuao Pei, Xiaokang Yang, Jiangmiao Pang, Yao Mu, and Ping Luo. Discrete diffusion vla: Bringing discrete diffusion to action decoding in vision-language-action policies. *arXiv preprint arXiv:2508.20072*, 2025. 6, 7

[37] Tao Lin, Gen Li, Yilei Zhong, Yanwen Zou, and Bo Zhao. Evo-0: Vision-language-action model with implicit spatial understanding. *arXiv preprint arXiv:2507.00416*, 2025. 2, 3, 4, 5

[38] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023. 6

[39] Chenghao Liu, Jiachen Zhang, Chengxuan Li, Zhimu Zhou, Shixin Wu, Songfang Huang, and Huiling Duan. Ttf-vla: Temporal token fusion via pixel-attention integration for vision-language-action models. *arXiv preprint arXiv:2508.19257*, 2025. 1

[40] Jiuming Liu, Guangming Wang, Zhe Liu, Chaokang Jiang, Marc Pollefeys, and Hesheng Wang. Regformer: an efficient projection-aware transformer network for large-scale point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8451–8460, 2023. 3

[41] Jiuming Liu, Jinru Han, Lihao Liu, Angelica I Aviles-Rivero, Chaokang Jiang, Zhe Liu, and Hesheng Wang. Mamba4d: Efficient 4d point cloud video understanding with disentangled spatial-temporal state space models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 17626–17636, 2025. 3

[42] Hao Lu, Tianshuo Xu, Wenzhao Zheng, Yunpeng Zhang, Wei Zhan, Dalong Du, Masayoshi Tomizuka, Kurt Keutzer, and Yingcong Chen. Drivingrecon: Large 4d gaussian reconstruction model for autonomous driving. *arXiv preprint arXiv:2412.09043*, 2024. 2

[43] Hao Lu, Yunpeng Zhang, Guoqing Wang, Qing Lian, Dalong Du, and Ying-Cong Chen. Towards generalizable multi-camera 3d object detection via perspective rendering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5811–5819, 2025. 2

[44] Zhichao Ma, Yutong Luo, Zheyu Zhang, Aijia Sun, Yinuo Yang, and Hao Liu. Reinforcement learning approach for highway lane-changing: Ppo-based strategy design. In *2025 10th International Conference on Electronic Technology and Information Science (ICETIS)*, pages 298–301, 2025. 1

[45] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, et al. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025. 1, 2, 3, 4, 6

[46] Chaojun Ni, Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Wenkang Qin, Guan Huang, Chen Liu, Yuyin Chen, Yida Wang, Xueyang Zhang, et al. Recondreamer: Crafting world models for driving scene reconstruction via online restoration. *arXiv preprint arXiv:2411.19548*, 2024. 2

[47] Chaojun Ni, Xiaofeng Wang, Zheng Zhu, Weijie Wang, Haoyun Li, Guosheng Zhao, Jie Li, Wenkang Qin, Guan Huang, and Wenjun Mei. Wonderturbo: Generating interactive 3d world in 0.72 seconds. *arXiv preprint arXiv:2504.02261*, 2025. 4

[48] Chaojun Ni, Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Wenkang Qin, Xinze Chen, Guanghong Jia, Guan Huang, and Wenjun Mei. Recondreamer-rl: Enhancing reinforcement learning via diffusion-based scene reconstruction. *arXiv preprint arXiv:2508.08170*, 2025. 2

[49] *NVIDIA Jetson Orin Series Technical Brief*. NVIDIA Cor-

poration, 2022. Technical Brief v1.2, TB_10749-001_v1.2. 1, 7, 8

[50] 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. 6, 7

[51] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025. 2, 3, 5, 6, 7

[52] Ayushman Sarkar, Mohd Yamani Idna Idris, and Zhenyu Yu. Reasoning in computer vision: Taxonomy, models, tasks, and methodologies. *arXiv preprint arXiv:2508.10523*, 2025. 1, 3

[53] Jiayao Shan, Sifan Zhou, Zheng Fang, and Yubo Cui. Ptt: Point-track-transformer module for 3d single object tracking in point clouds. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1310–1316. IEEE, 2021. 3

[54] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025. 1

[55] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025. 1, 3, 6, 7

[56] Lin Sun, Bin Xie, Yingfei Liu, Hao Shi, Tiancai Wang, and Jiale Cao. Geovla: Empowering 3d representations in vision-language-action models. *arXiv preprint arXiv:2508.09071*, 2025. 2, 3, 4

[57] Zuojin Tang, Bin Hu, Chenyang Zhao, De Ma, Gang Pan, and Bin Liu. Vlascd: A visual language action model for simultaneous chatting and decision making. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. 1

[58] GigaBrain Team, Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Haoyun Li, Jie Li, Jiagang Zhu, Lv Feng, et al. Gigabrain-0: A world model-powered vision-language-action model. *arXiv preprint arXiv:2510.19430*, 2025.

[59] GigaBrain Team, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Jie Li, Jindi Lv, Jingyu Liu, Lv Feng, et al. Gigabrain-0.5 m*: a vla that learns from world model-based reinforcement learning. *arXiv preprint arXiv:2602.12099*, 2026.

[60] Jingyi Tian, Le Wang, Sanping Zhou, Sen Wang, Jiayi Li, Haowen Sun, and Wei Tang. Pdfactor: Learning tri-perspective view policy diffusion field for multi-task robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15757–15767, 2025. 1

[61] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023. 3

[62] Boyuan Wang, Xinpan Meng, Xiaofeng Wang, Zheng Zhu, Angen Ye, Yang Wang, Zhiqin Yang, Chaojun Ni, Guan Huang, and Xingang Wang. Embodiedreamer: Advancing real2sim2real transfer for policy training via embodied world modeling. *arXiv preprint arXiv:2507.05198*, 2025. 1, 3

[63] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 3, 4

[64] Sen Wang, Le Wang, Sanping Zhou, Jingyi Tian, Jiayi Li, Haowen Sun, and Wei Tang. Flowram: Grounding flow matching policy with region-aware mamba framework for robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12176–12186, 2025. 1

[65] Weijie Wang, Jiagang Zhu, Zeyu Zhang, Xiaofeng Wang, Zheng Zhu, Guosheng Zhao, Chaojun Ni, Haoxiao Wang, Guan Huang, Xinze Chen, et al. Drivegen3d: Boosting feed-forward driving scene generation with efficient video diffusion. *arXiv preprint arXiv:2510.15264*, 2025. 4

[66] Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *arXiv preprint arXiv:2506.19850*, 2025. 6, 7

[67] Zhihao Wang, Jianxiong Li, Jinliang Zheng, Wencong Zhang, Dongxiu Liu, Yinan Zheng, Haoyi Niu, Junzhi Yu, and Xianyuan Zhan. Physiagent: An embodied agent framework in physical world. *arXiv preprint arXiv:2509.24524*, 2025. 1

[68] Zihan Wang, Jeff Tan, Tarasha Khurana, Neehar Peri, and Deva Ramanan. Monofusion: Sparse-view 4d reconstruction via monocular fusion, 2025. 3

[69] Quanmin Wei, Penglin Dai, Wei Li, Bingyi Liu, and Xiao Wu. Copeft: Fast adaptation framework for multi-agent collaborative perception with parameter-efficient fine-tuning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 23351–23359, 2025. 2

[70] Quanmin Wei, Penglin Dai, Wei Li, Bingyi Liu, and Xiao Wu. Infocom: Kilobyte-scale communication-efficient collaborative perception with information bottleneck. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 29731–29739, 2026. 2

[71] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025. 1, 3, 6

[72] 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. 6

[73] Shilin Xiao, Tianxiang Xu, Canran Xiao, Weihao Luo, Liwei Hou, and Chuangxin Zhao. Meta-ucf: Unified task-conditioned lora generation for continual learning in large language models. In *International Conference on Learning Representations (ICLR)*, 2026. 1

[74] Shaocheng Yan, Yiming Wang, Kaiyan Zhao, Pengcheng Shi, Zhenjun Zhao, Yongjun Zhang, and Jiayuan Li. Hemora: Unsupervised heuristic consensus sampling for robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1363–1373, 2025. 3

[75] Shanliang Yao, Runwei Guan, Zhaodong Wu, Yi Ni, Zile Huang, Ryan Wen Liu, Yong Yue, Weiping Ding, Eng Gee Lim, Hyungjoon Seo, et al. Waterscenes: A multi-task 4d radar-camera fusion dataset and benchmarks for autonomous driving on water surfaces. *IEEE Transactions on Intelligent Transportation Systems*, 25(11):16584–16598, 2024. 2

[76] Angen Ye, Zeyu Zhang, Boyuan Wang, Xiaofeng Wang, Dapeng Zhang, and Zheng Zhu. Vla-r1: Enhancing reasoning in vision-language-action models. *arXiv preprint arXiv:2510.01623*, 2025. 1

[77] Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, et al. Gigaworld-policy: An efficient action-centered world–action model. *arXiv preprint arXiv:2603.17240*, 2026. 1

[78] Zhenyu Yu, Mohd Yamani Idna Idris, Pei Wang, and Rizwan Qureshi. Cotextor: Training-free modular multilingual text editing via layered disentanglement and depth-aware fusion. In *Advances in Neural Information Processing Systems*, 2025. 3

[79] Shuang Zeng, Xinyuan Chang, Mengwei Xie, Xinran Liu, Yifan Bai, Zheng Pan, Mu Xu, and Xing Wei. Futuresightdrive: Thinking visually with spatio-temporal cot for autonomous driving. *arXiv preprint arXiv:2505.17685*, 2025. 1

[80] 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. 4

[81] Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, et al. 4d-vla: Spatiotemporal vision-language-action pretraining with cross-scene calibration. *arXiv preprint arXiv:2506.22242*, 2025. 2, 4, 6, 7

[82] Le Zhang, Rabiul Awal, and Aishwarya Agrawal. Contrasting intra-modal and ranking cross-modal hard negatives to enhance visio-linguistic compositional understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13774–13784, 2024. 1

[83] Le Zhang, Yihong Wu, Qian Yang, and Jian-Yun Nie. Exploring the best practices of query expansion with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1872–1883, 2024.

[84] Le Zhang, Qian Yang, and Aishwarya Agrawal. Assessing and learning alignment of unimodal vision and language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14604–14614, 2025. 1

[85] Wei Zhang, Feng Jiang, Chi-Fu Yang, Zhi-Peng Wang, and Tie-Jun Zhao. Research on unmanned surface vehicles environment perception based on the fusion of vision and lidar. *IEEE Access*, 9:63107–63121, 2021. 3

[86] Wei Zhang, Xian-zhong Gao, Chi-fu Yang, Feng Jiang, and Zhi-yuan Chen. A object detection and tracking method for security in intelligence of unmanned surface vehicles. *Journal of Ambient Intelligence and Humanized Computing*, 13 (3):1279–1291, 2022. 3

[87] Guosheng Zhao, Chaojun Ni, Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Boyuan Wang, Youyi Zhang, Wenjun Mei, and Xingang Wang. Drivedreamer4d: World models are effective data machines for 4d driving scene representation. *arXiv preprint arXiv:2410.13571*, 2024. 2

[88] Guosheng Zhao, Xiaofeng Wang, Chaojun Ni, Zheng Zhu, Wenkang Qin, Guan Huang, and Xingang Wang. Recondreamer++: Harmonizing generative and reconstructive models for driving scene representation. *arXiv preprint arXiv:2503.18438*, 2025. 2

[89] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 2, 3, 4, 5

[90] Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. Universal actions for enhanced embodied foundation models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22508–22519, 2025. 1, 6, 7

[91] Yaozong Zheng, Bineng Zhong, Qihua Liang, Guorong Li, Rongrong Ji, and Xianxian Li. Toward unified token learning for vision-language tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(4):2125–2135, 2023.

[92] Yaozong Zheng, Bineng Zhong, Qihua Liang, Zhiyi Mo, Shengping Zhang, and Xianxian Li. Odtrack: Online dense temporal token learning for visual tracking. In *Proceedings of the AAAI conference on artificial intelligence*, pages 7588–7596, 2024.

[93] Heng Zhou, Jing Tang, Jusheng Zhang, Yanshu Li, Canran Xiao, Liwei Hou, Zong Ke, and Jiawei Yao. Comem: Compositional concept-graph memory for vision–language adaptation. In *International Conference on Learning Representations (ICLR)*, 2026. 1

[94] Shijie Zhou, Alexander Vilesov, Xuehai He, Ziyu Wan, Shuwang Zhang, Aditya Nagachandra, Di Chang, Dongdong Chen, Xin Eric Wang, and Achuta Kadambi. Vlm4d: Towards spatiotemporal awareness in vision language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8600–8612, 2025. 3

[95] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025. 2, 4