S^2-VLA: State-Space Guided Vision-Language-Action Models for Long-Horizon Manipulation

Zhipeng Xie[1], **Zongyi Han**[1], **Xiangyi Wei**[1], **Shiliang Sun**[2], **Yang Li**[1] and **Jing Zhao**[1*]
[1]School of Computer Science and Technology, East China Normal University, Shanghai 200062, China
[2]State Key Laboratory of Submarine Geoscience, School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China
{51285901044, 51285901051, 52285901014}@stu.ecnu.edu.cn, shiliangsun@gmail.com, {yli, jzhao}@cs.ecnu.edu.cn

Abstract

Vision-Language-Action (VLA) models have demonstrated strong capabilities in robotic manipulation, but their performance degrades significantly in long-horizon tasks due to cumulative error propagation. This limitation largely arises from static feature fusion mechanisms that rely on fixed weights to combine visual, language, and action representations, preventing the model from adapting to different phases of task execution. To address this limitation, we propose S^2-VLA, a framework that introduces a State-Space Guided Adaptive Attention (SSGAA) mechanism. SSGAA maintains a belief state that tracks task progression and generates dynamic gating weights to adaptively fuse information from three complementary sources visual features for spatial perception, task intents for high-level task planning, and temporal action sequences for execution consistency. This adaptive fusion allows the model to shift its focus throughout task execution, aligning with the evolving requirements of different task stages. Despite its compact 2B parameter size, S^2-VLA consistently outperforms larger 7B-scale models and achieves state-of-the-art performance on long-horizon manipulation benchmarks, including LIBERO and SimplerEnv. highlighting the importance of adaptive feature fusion for long-horizon robotic manipulation.

1 Introduction

In recent years, significant advances in vision-language models (VLMs) [Bai *et al.*, 2025a; Chen *et al.*, 2024; Karamcheti *et al.*, 2024] have propelled the development of generalist robotic systems capable of perception, reasoning, and action under open-ended instructions. Within this paradigm, Vision-Language-Action (VLA) [Zhang *et al.*, 2025a] models have emerged as a promising framework that bridges high-level understanding to low-level control, enabling instruction driven robotic manipulation. The core of VLA research lies in effectively aligning multimodal representations comprising visual observations and language instructions with the action space to generate precise and executable policies.

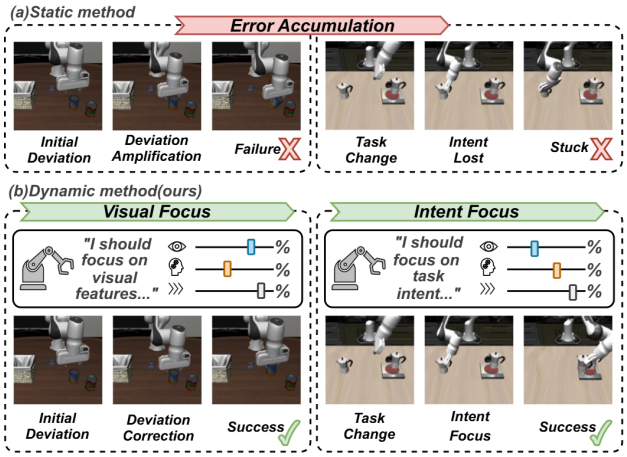

Figure 1: Illustrative examples comparing our S^2-VLA, which incorporates State-Space Guided Adaptive Attention, with static fusion approaches, highlighting its ability to mitigate early-stage bias.

The mainstream method aligns multimodal representations vision and language with the action space to generate executable policies by performing end-to-end fine-tuning of Vision-Language Models on large-scale robotic datasets [Collaboration *et al.*, 2025; Khazatsky *et al.*, 2025]. However, when applied to long-horizon manipulation tasks that require multi-step reasoning, these models are fundamentally limited by their static representation fusion mechanism. Considering a task such as **put both the cream cheese box and the butter in the basket.** The model employs fixed weights to fuse information from different modalities like vision and language throughout its execution. Consequently, during the grasping phase which demands precise localization, the model cannot sufficiently focus on spatial details of the object. Meanwhile, in the planning phase which requires comprehension of the overall goal, it fails to allocate adequate attention to semantic intent. This lack of phase-adaptive capability in its rigid design leads to decision biases at critical steps. More critically, in long-horizon tasks, the lack of a context-aware correction mechanism allows these early biases to propagate and amplify along the action chain, ulti-

mately leading to task failure.

To overcome this limitation, we propose the S^2-VLA framework. Its core is a novel State-Space Guided Adaptive Attention (SSGAA) mechanism. As illustrated in Figure 1, unlike static fusion, S^2-VLA maintains a compact belief state that tracks task progression and dynamically gates three complementary information streams spatial visual features, semantic task intent, and temporal action history enabling phase-aware adaptation. Extensive experiments on benchmarks such as LIBERO [Liu *et al.*, 2023], SimplerEnv [Li *et al.*, 2024b] and ALOHA [Zhao *et al.*, 2023] platform demonstrate that S^2-VLA, despite with compact 2B parameters, consistently outperforms larger 7B-scale models and sets a new state-of-the-art, effectively mitigating error propagation and significantly improving long-horizon task success.

The main contributions of our work are as follows:

- We propose S^2-VLA, which leverages pretrained VLMs to directly generate robot action policies through a belief-state-space interface, eliminating the need for large-scale action-sequence pretraining.
- We propose a novel State-Space Guided Adaptive Attention mechanism, which models task progress as a dynamic belief state. This belief state is used to guide a gating network that dynamically modulates the fusion of visual, linguistic, and action representations for long-horizon manipulation tasks.
- We show that S^2-VLA achieves robust long-horizon robotic manipulation with a lightweight model architecture, outperforming prior VLA methods on multiple standard benchmarks.

2 Related Work

2.1 Vision-Language-Action Models

In the ongoing development of generalist robotic policies, the field is observing a distinct trajectory that moves from imitation-learning-based expert policies toward general-purpose foundation models. Early pioneers in this domain, such as ACT [Zhao *et al.*, 2023], proved that it is possible to achieve strong dexterity by directly fitting the action distribution from demonstration data. Following this breakthrough, researchers have further explored generalist policies pre-trained on large-scale robot trajectories, resulting in representative systems like Octo [Team *et al.*, 2024], RDT [Liu *et al.*, 2025b], Seer [Tian *et al.*, 2024], RT-1 [Brohan *et al.*, 2023], RT-1-X [Collaboration *et al.*, 2025], and UnifiedVLA [Li *et al.*, 2025b]. However, it is worth noting that these approaches often struggle to handle open-world semantic instructions effectively. They tend to face challenges with long-tail concepts and compositional generalization, especially in scenarios where semantic understanding is not grounded in strong pre-trained VLMs. Consequently, to address this bottleneck, contemporary research has started to leverage pre-trained VLMs as semantic backbones, a strategy that has successfully given rise to VLA models [Zhang *et al.*, 2025a; Gao *et al.*, 2025].

Current VLA architectures can be broadly summarized into two paradigms for action decoding. *Autoregressive Token Generation Models* discretize actions into tokens and generate them sequentially, such as OpenVLA [Kim *et al.*, 2024], RT-2 [Zitkovich *et al.*, 2023], π_0-FAST [Pertsch *et al.*, 2025], CoT-VLA [Zhao *et al.*, 2025], FlowVLA [Zhong *et al.*, 2025], UniVLA [Bu *et al.*, 2025], TraceVLA [Zheng *et al.*, 2025], WorldVLA [Cen *et al.*, 2025], SpatialVLA [Qu *et al.*, 2025], Magma [Yang *et al.*, 2025], MolmoAct [Lee *et al.*, 2025], etc. While semantically more unified, this serial generation typically incurs higher inference latency. In contrast, *Parallel Decoding with Decoupled Policy Heads* decouples representation from execution to simultaneously improve speed and precision, such as OpenVLA-OFT [Kim *et al.*, 2025], π_0 [Black *et al.*, 2024], GR00T N1 [NVIDIA *et al.*, 2025], SmolVLA [Shukor *et al.*, 2025], CogACT [Li *et al.*, 2024a], 4D-VLA [Zhang *et al.*, 2025b], ThinkAct [Huang *et al.*, 2025], CronusVLA [Li *et al.*, 2025a], MemoryVLA [Shi *et al.*, 2025], RoboVLMs [Liu *et al.*, 2025a], and others. Under this paradigm, the VLM mainly provides semantic representations or high-level planning signals, while concrete action generation is handled by a decoupled policy head (e.g., diffusion or flow-matching models or ACT-style chunking). We adopt the latter paradigm to achieve efficient continuous control, with a focus on optimizing the feature fusion mechanism, aiming to fully leverage the perceptual capabilities of VLMs and appropriately bridge them with action generation for effective long-horizon task execution.

2.2 Policy Generation and Feature Integration

The performance of decoupled VLAs depends on *what* information is utilized from the backbone and *how* it is integrated. Existing methods typically exhibit statically configured feature integration, which limits their adaptability.

Static Focus on Compressed Representations. Most mainstream methods [Liu *et al.*, 2025b; Kim *et al.*, 2025] restrict their attention to the final output representations. Specifically, OpenVLA-OFT [Kim *et al.*, 2025] utilizes Learnable Action Queries to perform parallel decoding of future action chunks. However, the policy head attends exclusively to embeddings at these specific query token positions. While these embeddings encapsulate rich semantic intent, they constitute an information bottleneck, abstracting away the fine-grained spatial details present in the VLM's image token embeddings. This rigid focus makes it difficult for the model to recover the precise geometric information required for high-dexterity tasks.

Static Multi-level Fusion. To retrieve lost details, SmolVLA [Shukor *et al.*, 2025] and VLA-Adapter [Wang *et al.*, 2025] introduce mechanisms to aggregate multi-level backbone features (e.g., intermediate hidden states). Although this expands the information scope, the fusion strategy remains *static*. The models learn a fixed set of weights to balance these information sources. Consequently, they fail to adaptively shift their focus as applying the same fusion ratio regardless of whether the robot is in a visually critical phase (e.g., aligning for a grasp) or an intent-critical phase (e.g., executing motion primitives).

We argue that effective manipulation necessitates phase aware trade-offs visual precision for localization versus se-

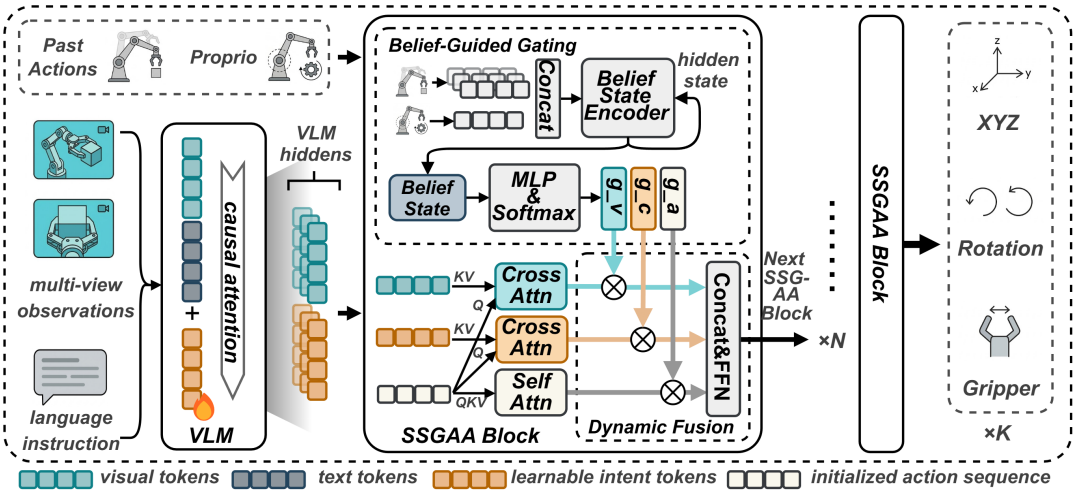

Figure 2: Architecture and Workflow of the S^2-VLA Framework. The Workflow processes multimodal inputs through a vision-language backbone, with the core State-Space Guided Adaptive Attention(SSGAA) module adaptively fusing spatial, semantic, and temporal information under the dynamic guidance of the belief state.

mantic intent for planning a critical temporal dynamic ignored by current methods.

3 Methodology

3.1 Problem Formulation

We formalize the language-conditioned long-horizon manipulation tasks in VLA model. At each time step t, an agent receives a multi-view visual observation $\mathbf{V}_t$, a natural language instruction $\mathbf{L}_t$, and proprioceptive feedback $\mathbf{P}_t$. A standard VLA policy is typically formulated as a direct mapping from these inputs to a future action sequence

$$\mathbf{A}_{t:t+K-1} = \pi_{\text{VLA}}(\mathbf{V}_t, \mathbf{L}_t, \mathbf{P}_t), \tag{1}$$

where $\mathbf{A}_{t:t+K-1} = [\mathbf{a}_t, \mathbf{a}_{t+1}, \ldots, \mathbf{a}_{t+K-1}]$ denotes K the predicted actions.

Although effective for short-horizon tasks, this formulation fundamentally lacks an explicit mechanism to maintain temporal coherence. Each action is generated from a static sensory snapshot, disregarding the execution history and how past actions have led to the current state. This makes the model susceptible to error accumulation in long-horizon scenarios.

3.2 S^2-VLA

Belief-Driven VLA Paradigm

To overcome the lack of temporal coherence in static VLA models for long-horizon tasks, we formulate an internal belief state $\mathbf{b}_t \in \mathbb{R}^{d_b}$. This state serves as a compact temporal context, summarizing **where we are** in task execution, and is recursively updated to encode task progression and historical information. The belief state update integrates the recent action sequence and its sensory feedback, formulated as

$$\begin{cases} (\mathbf{o}_t^{(l)}, \mathbf{h}_t^{(l)}) = f_\phi(\mathbf{h}_t^{(l-1)}, \mathbf{A}_{t-K:t-1}, \mathbf{P}_t) \\ \mathbf{b}_t^{(l)} = \mathbf{W}_b \cdot \mathbf{o}_t^{(l)} + \boldsymbol{\beta}_b \end{cases} \tag{2}$$

where f_ϕ is a learned update function, K is the lookback horizon, and t and l denote the time step and the layer index of the action head, respectively. $\mathbf{h}_t^{(l)}$ represents the hidden state updated via the l layer at time step t, initialized by the low-level features $\mathbf{h}_t^{(0)}$. This formulation explicitly models the causal relationship between the agent's actions $\mathbf{A}_{t-K:t-1}$ and environmental feedback (proprioception), enabling the model to track task progress and detect execution deviations. For initial steps where the number of past actions is less than K, zero-padding is applied to the sequence. We implement f_ϕ using a lightweight Gated Recurrent Unit (GRU) [Cho *et al.*, 2014]. Its recurrent nature maintains temporally coherent representations, compressing the history of action-perception pairs into a dynamic representation of task phase and execution quality. The complete policy is then conditioned on this belief state to generate actions as

$$\mathbf{A}_{t:t+K-1} = \pi'_{\text{VLA}}(\mathbf{V}_t, \mathbf{L}_t, \mathbf{P}_t \mid \mathbf{b}_t). \tag{3}$$

The belief state $\mathbf{b}_t$ is learned end-to-end from the action prediction loss, without any explicit supervision on task phases or execution errors. By minimizing prediction error, the model is forced to encode predictive information about future successful actions from the history of action-perception sequences. Consequently, distinct patterns arising from different task phases (e.g., **approaching**, **grasping**, **placing**) as well as discrepancies between expected and actual outcomes due to errors or perturbations are naturally encoded into $\mathbf{b}_t$. This makes $\mathbf{b}_t$ an latent representation that emergently encodes both task-phase information and execution fidelity, distilled directly from action-perception dynamics through end-to-end optimization.

Framework Overview

The belief state $\mathbf{b}_t$ provides crucial guidance for adaptive fusion of multimodal information. Building upon this, we propose the S^2-VLA framework. Its core is a novel State-Space

Guided Adaptive Attention mechanism. SSGAA consist of three functionally complementary and parallel attention pathways. Instead of static fusion, these pathways are adaptively integrated via a dynamic gating network regulated by the belief state $\mathbf{b}_t$, enabling stage-aware and adaptive feature aggregation. This design enables the policy to dynamically balance spatial details an d semantic intent based on action history and proprioception, thereby enhancing execution consistency in long-horizon tasks. The overall architecture of S^2-VLA are illustrated in Figure 2.

Unified Input Construction

Visual features $\mathbf{F}_t = \text{ViT}(\mathcal{V}_t) \in \mathbb{R}^{N_v \times d}$ are extracted from multi-view images $\mathcal{V}_t$. Language instructions $\mathbf{L}$ are tokenized into $\{\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_{N_l}\} \in \mathbb{R}^{N_l \times d}$, and learnable intent tokens $\mathbf{A}_{\text{tokens}}$ are embedded as $\{\tilde{\mathbf{a}}_1, \tilde{\mathbf{a}}_2, \ldots, \tilde{\mathbf{a}}_{N_a}\} \in \mathbb{R}^{N_a \times d}$ for task-specific context. The combined input sequence is:

$$\mathbf{X} = \left[\mathbf{F}_t;\ \mathbf{l}_1, \ldots, \mathbf{l}_{N_l};\ \tilde{\mathbf{a}}_1, \ldots, \tilde{\mathbf{a}}_{N_a}\right] \in \mathbb{R}^{(N_v+N_l+N_a)\times d}. \tag{4}$$

The sequence is processed through H Transformer layers to produce hierarchical representations.

State-Space Guided Adaptive Attention

The SSGAA mechanism consists of belief-guided dynamic gating and three complementary attention pathways, all operating on contextualized representations derived from the unified input sequence $\mathbf{X}$.

Action Sequence Self-Attention. Ensures temporal coherence in the predicted motion trajectories. This branch operates on a dedicated action-sequence embedding $\mathbf{A}^{(0)} \in \mathbb{R}^{T\times K\times d}$ where T denotes the action dimension and K the number of predicted steps, which is initialized as zero vectors and refined layer-wise through self-attention [Vaswani *et al.*, 2017] within the SSGAA module. This enables modeling of the temporal dependencies across the predicted K steps.

Low-Level Visual Cross-Attention. Provides precise spatial grounding by attending to fine-grained visual features. We construct a visual vector $\mathbf{C}_{\text{vis}} \in \mathbb{R}^{N_v \times d}$ by stacking the hidden states of visual tokens from one transformer layer. The operation is defined as

$$\mathbf{O}_{\text{vis}} = \text{Softmax}\left(\frac{\mathbf{Q}(\mathbf{C}_{\text{vis}}\mathbf{W}_k^{\text{vis}})^\top}{\sqrt{d}}\right)(\mathbf{C}_{\text{vis}}\mathbf{W}_v^{\text{vis}}), \tag{5}$$

where $\mathbf{Q} \in \mathbb{R}^{K\times d}$is the query projection derived from the learnable action sequence representation and $\mathbf{W}_k^{\text{vis}}, \mathbf{W}_v^{\text{vis}} \in \mathbb{R}^{d\times d}$ are learnable key and value projections that preserve fine-grained spatial details crucial for object manipulation.

High-Level Intent Cross-Attention. Facilitates semantic intent understanding by processing aggregated multimodal context. We form a semantic context vector $\mathbf{C}_{\text{ite}} \in \mathbb{R}^{N_a\times d}$ from the hidden states of the N_a learnable intent tokens across layers. The attention is computed as

$$\mathbf{O}_{\text{ite}} = \text{Softmax}\left(\frac{\mathbf{Q}(\mathbf{C}_{\text{ite}}\mathbf{W}_k^{\text{ite}})^\top}{\sqrt{d}}\right)(\mathbf{C}_{\text{ite}}\mathbf{W}_v^{\text{ite}}), \tag{6}$$

where $\mathbf{W}_k^{\text{ite}}, \mathbf{W}_v^{\text{ite}} \in \mathbb{R}^{d\times d}$ project the action-token representations, which encode the distilled task semantics, into keys and values respectively.

Belief-Guided Gating. The belief state $\mathbf{b}_t$ dynamically modulates these three branches through a gating mechanism. At layer l, soft gating weights are computed as

$$\left[g_{\text{vis}}^{(l)},\ g_{\text{ite}}^{(l)},\ g_{\text{act}}^{(l)}\right]^\top = \text{Softmax}\Big(\text{MLP}_g^{(l)}(\mathbf{b}_t^{(l)}\Big). \tag{7}$$

Branch outputs are fused as

$$\mathbf{H}^{(l)} = g_{\text{vis}}^{(l)} \cdot \mathbf{O}_{\text{vis}}^{(l)} + g_{\text{ite}}^{(l)} \cdot \mathbf{O}_{\text{ite}}^{(l)} + g_{\text{act}}^{(l)} \cdot \mathbf{O}_{\text{act}}^{(l)}, \tag{8}$$

where $\mathbf{O}_{\text{vis}}^{(l)}$, $\mathbf{O}_{\text{ite}}^{(l)}$, and $\mathbf{O}_{\text{act}}^{(l)}$ denote the outputs of the visual, semantic, and action self-attention branches at layer l, respectively. The fused representation $\mathbf{H}^{(l)}$ is processed by a feed-forward network. The processed output then serves a dual function. First, it provides the query input $\mathbf{Q}^{(l+1)}$ for the next layer's visual and context cross-attention branches. Second, it is fed into the same layer's action sequence self-attention branch. This design enables iterative refinement through multi-layer processing.

Parallel Action Decoding

The final action prediction is generated by applying layer normalization followed by a linear projection to the output of the last SSGAA layer are computed as

$$\hat{\mathbf{A}}_{t:t+K-1} = \text{LN}\big(\mathbf{H}^{(L_{\text{out}})}\big)\mathbf{W}_{\text{out}}^\top + \boldsymbol{\beta}_{\text{out}}, \tag{9}$$

where $\text{LN}(\cdot)$ denotes layer normalization. The learnable parameters are the weight matrix $\mathbf{W}_{\text{out}} \in \mathbb{R}^{T\times d}$ and the bias vector $\boldsymbol{\beta}_{\text{out}} \in \mathbb{R}^T$. The bias $\boldsymbol{\beta}_{\text{out}}$ is broadcasted across the temporal dimension K during the addition.

3.3 Training and Inference Strategies

Unified Loss Formulation

The training objective is defined by the action prediction loss, which ensures precise trajectory generation by minimizing the discrepancy between predicted and ground-truth actions

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{action}} = \frac{1}{K}\sum_{k=0}^{K-1} \|\mathbf{a}_{t+k} - \hat{\mathbf{a}}_{t+k}\|_2^2, \tag{10}$$

where $\mathbf{a}_{t+k}$ denotes the ground-truth action at future step $t+k$ and $\hat{\mathbf{a}}_{t+k}$ is the corresponding prediction. This formulation directly optimizes the model for action prediction accuracy without additional regularization terms.

End-to-End Training and Inference

The S^2-VLA model is trained end-to-end, jointly optimizing all components including the pre-trained Qwen3-VL [Bai *et al.*, 2025a] backbone, belief update module, and SSGAA attention mechanism to allow gradient flow and coordinated adaptation for robotic manipulation. During inference, the model operates in a temporally recurrent loop. At each step, it first updates its state through a GRU [Cho *et al.*, 2014]. The model then computes adaptive gating weights to fuse visual, textual, and action features into a unified representation. Based on this representation, it predicts and executes the next action before proceeding to the following time step. This design ensures temporal consistency and robustness via continuous state estimation. Simultaneously, the dynamic gating mechanism enables automatic attention shifting across different modalities according to the current task phase, thereby facilitating adaptive and reliable task execution. Details are provided in the supplementary material.

Method	Scale (B)	Spatial (%)	Object (%)	Goal (%)	Long (%)	Avg. (%)
FlowVLA (2025, ArXiv)	8.5	93.2	95.0	91.6	72.6	88.1
UnifiedVLA (2025, ArXiv)	8.5	95.4	98.8	93.6	94.0	95.5
OpenVLA (2024, CoRL)	7	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT (2025, RSS)	7	97.6	98.4	97.9	94.5	97.1
UniVLA (2025, RSS)	7	96.5	96.8	95.6	92.0	95.2
MemoryVLA (2025, ArXiv)	7	**98.4**	98.4	96.4	93.4	96.7
CoT-VLA (2025, CVPR)	7	87.5	91.6	87.6	69.0	81.1
WorldVLA (2025, ArXiv)	7	87.6	99.2	83.4	60.0	81.8
CronusVLA (2025, AAAI)	7	97.3	**99.6**	96.9	94.0	97.0
TraceVLA (2025, ArXiv)	7	84.6	85.2	75.1	54.1	74.8
MolmoAct (2025, ArXiv)	7	87.0	95.4	87.6	77.2	86.6
ThinkAct (2025, NeurIPS)	7	88.3	91.4	87.1	70.9	84.4
4D-VLA (2025, ArXiv)	4	88.9	95.2	90.9	79.1	88.6
SpatialVLA (2025, RSS)	4	88.2	89.9	78.6	55.5	78.1
π_0 (2025, RSS)	3	96.8	98.8	95.8	85.2	94.2
π_0-FAST (2025, RSS)	3	96.4	96.8	88.6	60.2	85.5
SmolVLA (2025, ArXiv)	2.2	93.0	94.0	91.0	77.0	88.8
GR00T N1 (2025, ArXiv)	2	94.4	97.6	93.0	90.6	93.9
Seer (2025, ArXiv)	0.57	-	-	-	78.7	78.7
VLA-OS (2025, ArXiv)	0.5	87.0	96.5	92.7	66.0	85.6
S^2-VLA (ours)	2	**98.4**	**99.6**	**98.4**	**96.4**	**98.2**

Table 1: Performance comparison on LIBERO benchmark (Spatial/Object/Goal/Long subsets).

4 Experiments

4.1 Experimental Setup

To evaluate the robustness of policies in long-horizon manipulation tasks, we evaluate S^2-VLA on three main benchmarks. During training, the model is optimized using four H100 GPUs. For inference, the system requires only 7GB of VRAM to perform deployment. In simulation, we use the LIBERO [Liu *et al.*, 2023] benchmark (Spatial, Object, Goal, and Long subsets) with fixed trajectories per task, language instructions, and third-person views, following official protocols with scripted initial scene sampling. Additionally, we evaluate within the SIMPLER [Li *et al.*, 2024b] beachmark which is a high-fidelity simulation platform designed to bridge the real-to-sim control and visual gap by faithfully replicating real-world conditions for representative robotic arms such as WidowX. Extensive testing has demonstrated a strong correlation between SIMPLER performance and real-world results. In the real world, we use the ALOHA bimanual system [Zhao *et al.*, 2023] (30 Hz control) with multi-view overhead and wrist cameras and proprioceptive inputs. Tasks include pick-and-place, stacking, desktop organization, and utensil distribution. Experimental setup details are provided in the supplementary materials.

4.2 Main Results Analysis

Performance on Simulation Benchmark

As shown in Table 1, we conduct comparative experiments with multiple methods on the LIBERO benchmark. **Bold** is the best performance and Bold is the suboptimal performance. The results indicate that S^2-VLA demonstrates SOTA while maintaining efficiency advantages. It achieves success rates of 98.4%, 99.6%, 98.4%, and 96.4% on the Spatial, Object, Goal, and Long-Horizon subsets, respectively, with an average success rate of 98.2%, significantly outperforming the compared methods. Particularly in long-horizon tasks, the model effectively mitigates error accumulation issues through the State-Space Guided Adaptive Attention mechanism.

Method	Spoon on Towel	Carrot on Plate	Stack Cube	Eggplant in Basket	Avg. Success
RT-1-X (2024, ICRA)	0.0	4.2	0.0	0.0	1.1
OpenVLA (2024, RSS)	4.2	0.0	0.0	12.5	4.2
Octo-Base (2024, ArXiv)	15.8	12.5	0.0	41.7	17.5
RoboVLMs (2024, ArXiv)	45.8	20.8	4.2	79.2	37.5
CogACT-Base (2024, ArXiv)	71.7	50.8	15.0	67.5	51.3
CogACT-Large (2024, ArXiv)	58.3	45.8	29.2	95.8	57.3
π_0-Uniform* (2024, ArXiv)	63.3	58.8	21.3	79.2	55.7
π_0-Beta* (2024, ArXiv)	**84.6**	55.8	**47.9**	85.4	68.4
Magma (2025, PP)	37.5	29.2	20.8	91.7	44.8
TraceVLA (2025, ArXiv)	12.5	16.6	16.6	65.0	27.7
SpatialVLA (2025, ArXiv)	16.7	25.0	29.2	**100.0**	42.7
MemoryVLA (2025, ArXiv)	75.0	75.0	37.5	**100.0**	**71.9**
S^2-VLA (ours)	83.3	**87.5**	**41.7**	**100.0**	**78.1**

Table 2: Performance comparison on SimplerEnv-Bridge with WidowX robot.

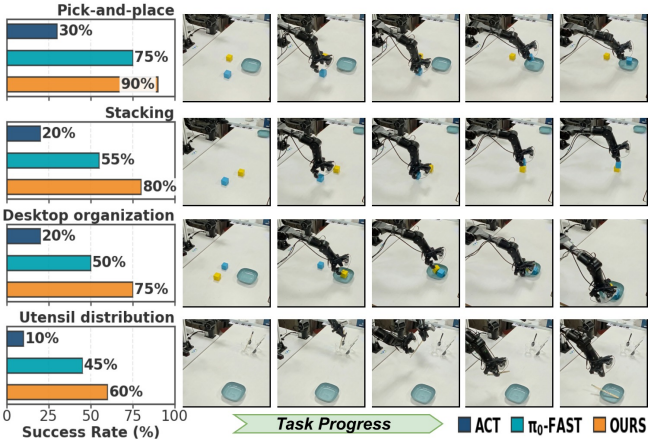

Figure 3: Real World Performance of our S^2-VLA compared with ACT and π_0-FAST.

Table 2 presents the performance on four manipulation tasks using the WidowX robot platform. Our method still achieves state-of-the-art results even in the highly realistic simulation environment, SimplerEnv-Bridge.

Real-World Experimental Evaluation

We evaluated S^2-VLA on a real-world ALOHA dual-arm mobile manipulation platform through a series of complex manipulation tasks. The evaluation comprised four distinct tasks. In the pick-and-place task, the positions of a blue and a yellow block were randomly initialized, requiring the system to pick up the blue block and place it into a bowl. In the stacking task, after random initialization of the block positions, the system was required to stack the yellow block on top of the blue block. The desktop organization task involved randomly initializing block positions and then first placing the yellow block into a bowl, followed by placing the blue block into the same bowl. Finally, the utensil distribution task required the system, after random initialization of utensil positions, to use the left arm to pick up a pair of chopsticks, pass them to the right arm, and then place the chopsticks into a bowl with the right arm. Details are provided in supplementary materials.

As illustrated in Figure 3, the key steps in the execution process of our S^2-VLA are presented. It achieved superior success rates compared to all baseline methods, demonstrat-

Setting	Gated layers	Perf.(%)	Δ vs. w/o Gate
w/o Gate	–	95.0	+0.0
Gate@All	all layers	94.4	-0.6
Single-layer ablations			
Gate@1	1	94.2	-0.8
Gate@6	6	96.0	+1.0
Gate@12	12	96.4	+1.4
Gate@18	18	95.8	+0.8
Gate@24	24	94.8	-0.2
Combination strategy			
Combination	6,12,18	94.4	-0.6
State guidance ablation			
w/o State	12	95.8	+0.8
S^2-VLA(ours)	**12**	**96.4**	**+1.4**

Table 3: Ablation on gating configuration and belief-state guidance. Δ shows performance change relative to "w/o Gate".

ing that its SSGAA mechanism enables effective real-world transfer capability.

4.3 Ablation Studies

This section ablates the core design of S^2-VLA, the action head constructed from a 24-layer SSGAA mechanism. We systematically address four key questions *(i) Is dynamic gating necessary? (ii) Is gating performance depth-sensitive? (iii) Does belief-state guidance provide additional benefit? (iv) What is the optimal SSGAA configuration?* All experiments follow the same training and evaluation protocols as the main experiments, with only the gating strategy modified.

The study begins by evaluating whether state-driven gating improves performance. As shown in Table 3, disabling gating (**w/o Gate**) results in 95.0% performance, while enabling gating in all layers (**Gate@All**) slightly degrades performance to 94.4%. This suggests that indiscriminate gating across all layers is not beneficial. To investigate depth sensitivity, we conduct single-layer gating experiments, enabling gating only at specific layers while keeping others static. Results reveal a clear pattern that gating provides significant gains at middle layers (6, 12, 18), with layer 12 achieving the best performance (96.4%), while shallow and deep layers show minimal or negative impact. This reflects the hierarchical structure of feature extraction. Intermediate levels capture semantically rich features ideal for adaptive resource allocation, while early and final stages are better by static fusion.

Given the success of single-layer gating at layer 12, we investigate whether combining multiple high-performing layers (6, 12, 18) yields further improvements. Surprisingly, this multi-layer configuration performs worse (94.4%) than both the best single-layer setting and the no-gating baseline. This indicates that multi-point gating introduces instability. To isolate the contribution of belief-state guidance, we replace the dynamic state input $\mathbf{b}_t$ with a learnable static vector in the Gate@12 configuration. This **w/o State** variant achieves only 95.8%, significantly lower than the full SSGAA (96.4%). This confirms that performance gains arise not merely from introducing gating, but specifically from the dynamic modulation of gating weights by the evolving belief state, enabling phase-aware adaptation.

Method / Model	Size	SR(%)	Δ vs. S^2-VLA
Qwen2.5-VL + FAST (2025, RSS)	3B	90.2	-6.2
Qwen2.5-VL + OFT (2025, RSS)	3B	92.0	-4.4
Qwen2.5-VL + PI (2025, RSS)	3B	88.4	-8.0
Qwen2.5-VL + GR00T (2025, ArXiv)	3B	90.8	-5.6
Qwen3-VL + FAST (2025, RSS)	4B	90.6	-5.8
Qwen3-VL + OFT (2025, RSS)	4B	93.8	-2.6
Qwen3-VL + PI (2025, RSS)	4B	88.4	-8.0
Qwen3-VL + GR00T (2025, ArXiv)	4B	92.0	-4.4
Qwen3-VL + Adapter-Pro (2025, AAAI)	2B	94.4	-2.0
S^2-VLA (ours)	**2B**	**96.4**	**–**

Table 4: Performance comparison on LIBERO Long benchmark.

Ablation on State Guided Gating

To further validate the core role of belief-state guidance, we conduct a controlled experiment. On the best-performing gating layer (Gate@12), we remove the input from the belief state $\mathbf{b}_t$ and replace it with a learnable constant vector to generate static gating weights (denoted as **w/o Belief State**). The results in table 3 show that the static gating setup performs worse than the full belief-state-guided gating. This demonstrates that the performance improvement is not simply due to the introduction of a gating mechanism, but rather depends critically on having the gating weights be dynamically modulated by the current belief state $\mathbf{b}_t$. This dynamic mechanism enables the model to adaptively adjust the fusion of multi-source information (visual, semantic, and action-sequence) according to the task phase. Our ablation study demonstrates that Gating is more effective at semantically rich middle layers and belief-state guidance is essential for optimal performance. Consequently, S^2-VLA adopts single-layer gating at a critical middle layer (layer 12, The experimentally determined optimal SSGAA configuration layer) with belief-state guidance as its core configuration.

4.4 Evaluation of QwenVL-Based Models

To isolate the performance gains attributable to architectural design, we systematically transplant several mainstream SOTA VLA Policy Head methods onto the QwenVL family of backbones including Qwen2.5-VL [Bai *et al.*, 2025b] and Qwen3-VL [Bai *et al.*, 2025a] for a fair comparison. As shown in Table 4, although all methods are built upon the same pre-trained QwenVL weights, S^2-VLA achieves the highest success rate of 96.4% with its compact 2B parameter architecture. This performance not only surpasses the similarly sized Qwen3-VL+Adapter-Pro [Wang *et al.*, 2025] with the success rate of 94.4% but also exceeds all methods using larger 3B and 4B backbones, which Qwen3-VL+OFT [Kim *et al.*, 2025] only gets success rate of 93.8%. The results demonstrate that the SSGAA mechanism is more effective than merely scaling up parameters for long-horizon tasks. S^2-VLA successfully breaks through the performance plateau observed in static fusion methods, which typically saturates in the 88%-94% range as task complexity increases. By integrating the semantic representations from the foundation model with real-time state predictions, it achieves dynamic

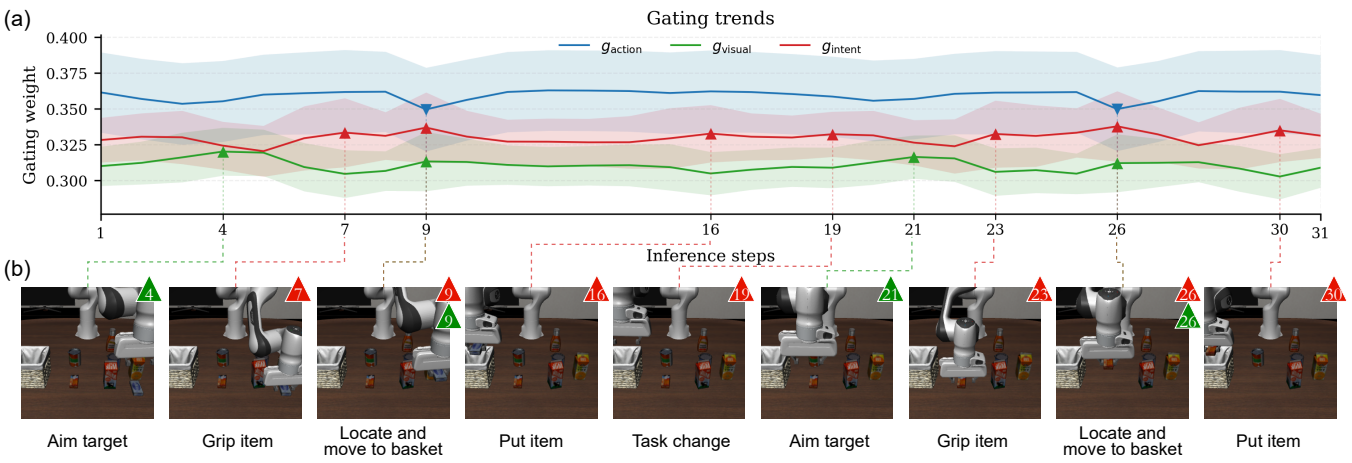

Figure 4: Gating weight trajectories and stage-aligned keyframes for a representative LIBERO-Long rollout.

shifting of perceptual focus across task phases, thereby enhancing robustness against long-horizon error accumulation.

4.5 Gating Interpretability Analysis

The primary cause of failure in long-horizon manipulation tasks lies in the propagation and amplification of perception or decision-making errors from early key steps during subsequent execution, ultimately leading to overall task failure. To elucidate how S^2-VLA addresses this challenge at a mechanistic level, we performed step-by-step visualization of the gating weights in its SSGAA module. A representative long-horizon task **put both the cream cheese box and the butter in the basket** is selected for analysis. During a successful execution trajectory, the model's dynamic allocation of gating weights across its three attention branches is recorded at each reasoning step, which include low-level visual cross-attention (preserving fine-grained spatial details), high-level semantic cross-attention (goal-oriented planning), and action sequence self-attention (temporal coherence in motion generation). The resulting dynamic trajectories of the weights, aligned with stage-wise keyframes, are shown in Figure 4. The main observations are summarized as follows.

During phases that require precise object grounding and fine spatial alignment (e.g., when the end-effector approaches the target object, corresponding to the "Aim target" and "Locate and move to basket" keyframes marked by a green triangle in Figure 4), the weight of the low-level visual cross-attention (green curve, g_{visual}) rises significantly. This indicates that the model relies more heavily on early-layer visual tokens to support high-precision positioning.

At key decision points involving subtask switching or phase transitions (e.g., triggering the grasp action, transporting the object to the basket, and executing the release action, corresponding to stages such as "Grip item" and "Place item" in keyframes marked by a red triangle Figure 4), the weight of the high-level Intent cross-attention (red curve, g_{intent}) exhibits clear peaks. This demonstrates that the model leverages aggregated multimodal task representations to perform explicit stage transitions and behavioral planning.

During long-range, steady-state motion segments, the action sequence self-attention (blue curve, g_{action}) remains dominant. This allows the model to maintain smooth and temporally consistent trajectories through coherent step-by-step motion modeling, thereby reducing unnecessary perceptual and planning overhead.

This SSGAA mechanism enables the model to adaptively enhance the most pertinent information-processing pathway according to different task phases, thereby improving decision robustness in error-prone stages such as fine manipulation and phase switching. Experimental results confirm that this mechanism effectively reduces the likelihood of errors in critical steps, suppresses error propagation and accumulation from the source, and consequently enhances both the overall success rate and execution stability of long-horizon manipulation tasks.

Efficiency	TraceVLA	CronusVLA	OpenVLA-OFT	**S^2-VLA(ours)**
Throughput (Hz) ↑	4.3	8.7	<u>71.4</u>	**80.8**

Table 5: Inference efficiency comparison.

4.6 Performance Efficiency Metrics

To verify inference efficiency, we compare the 7B model with our method, with results shown in Table 5.

5 Conclusion

S^2-VLA effectively mitigates error accumulation in long-horizon tasks by employing a state-space-guided adaptive attention mechanism, which dynamically fuses visual, linguistic, and action representations. Despite its compact size of 2B parameters, the model outperforms conventional 7B-scale models and achieves state-of-the-art success rates on long-horizon manipulation benchmarks, including LIBERO [Liu *et al.*, 2023] and SimplerEnv [Li *et al.*, 2024b]. Future work will explore the architectural compatibility of the SSGAA mechanism and extend its core idea of belief state modeling to different generative frameworks, such as Diffusion models or Flow Matching, to enhance the modeling capability for complex multimodal action distributions.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Projects 62476089, 62576206 and 62572194, and by the Shanghai Science and Technology Committee under Grant No. 24511103202.

Contribution Statement

Zhipeng Xie and Zongyi Han contributed equally to this work. Jing Zhao is the corresponding author.

References

[Bai *et al.*, 2025a] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL Technical Report, November 2025. arXiv:2511.21631 [cs].

[Bai *et al.*, 2025b] 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, February 2025. arXiv:2502.13923 [cs].

[Black *et al.*, 2024] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi_0: A Vision-Language-Action Flow Model for General Robot Control, November 2024. arXiv:2410.24164 [cs].

[Brohan *et al.*, 2023] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. RT-1: Robotics Transformer for Real-World Control at Scale, August 2023. arXiv:2212.06817 [cs].

[Bu *et al.*, 2025] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to Act Anywhere with Task-centric Latent Actions, November 2025. arXiv:2505.06111 [cs].

[Cen *et al.*, 2025] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, et al. WorldVLA: Towards Autoregressive Action World Model, June 2025. arXiv:2506.21539 [cs].

[Chen *et al.*, 2024] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. On scaling up a multilingual vision and language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14432–14444, June 2024.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, September 2014. arXiv:1406.1078 [cs].

[Collaboration *et al.*, 2025] Open X.-Embodiment Collaboration, Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, et al. Open X-Embodiment: Robotic Learning Datasets and RT-X Models, May 2025. arXiv:2310.08864 [cs].

[Gao *et al.*, 2025] Chongkai Gao, Zixuan Liu, Zhenghao Chi, Junshan Huang, Xin Fei, Yiwen Hou, Yuxuan Zhang, Yudi Lin, Zhirui Fang, Zeyu Jiang, and Lin Shao. VLA-OS: Structuring and Dissecting Planning Representations and Paradigms in Vision-Language-Action Models, June 2025. arXiv:2506.17561 [cs].

[Huang *et al.*, 2025] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. ThinkAct: Vision-Language-Action Reasoning via Reinforced Visual Latent Planning, September 2025. arXiv:2507.16815 [cs].

[Karamcheti *et al.*, 2024] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: investigating the design space of visually-conditioned language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 23123–23144, Vienna, Austria, July 2024. JMLR.org.

[Khazatsky *et al.*, 2025] 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, April 2025. arXiv:2403.12945 [cs].

[Kim *et al.*, 2024] 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, September 2024. arXiv:2406.09246 [cs].

[Kim *et al.*, 2025] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success, April 2025. arXiv:2502.19645 [cs].

[Lee *et al.*, 2025] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. MolmoAct: Action Reasoning Models that can Reason in Space, September 2025. arXiv:2508.07917 [cs].

[Li *et al.*, 2024a] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, Yizhong Zhang, et al. CogACT: A Foundational Vision-Language-Action Model for Synergizing Cognition and Action in Robotic Manipulation, November 2024. arXiv:2411.19650 [cs].

[Li *et al.*, 2024b] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, et al. Evaluating Real-World Robot Manipulation Policies in Simulation, May 2024. arXiv:2405.05941 [cs].

[Li *et al.*, 2025a] Hao Li, Shuai Yang, Yilun Chen, Xinyi Chen, Xiaoda Yang, Yang Tian, Hanqing Wang, Tai Wang, Dahua Lin, Feng Zhao, et al. CronusVLA:

Towards Efficient and Robust Manipulation via Multi-Frame Vision-Language-Action Modeling, October 2025. arXiv:2506.19816 [cs].

[Li *et al.*, 2025b] Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified Video Action Model, April 2025. arXiv:2503.00200 [cs].

[Liu *et al.*, 2023] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 44776–44791. Curran Associates, Inc., 2023.

[Liu *et al.*, 2025a] Huaping Liu, Xinghang Li, Peiyan Li, et al. Towards generalist robot policies: What matters in building vision-language-action models. *Research Square*, feb 2025. Preprint available at Research Square.

[Liu *et al.*, 2025b] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation, March 2025. arXiv:2410.07864 [cs].

[NVIDIA *et al.*, 2025] NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. GR00T N1: An Open Foundation Model for Generalist Humanoid Robots, March 2025. arXiv:2503.14734 [cs].

[Pertsch *et al.*, 2025] 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, January 2025. arXiv:2501.09747 [cs].

[Qu *et al.*, 2025] 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, May 2025. arXiv:2501.15830 [cs].

[Shi *et al.*, 2025] Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, et al. MemoryVLA: Perceptual-Cognitive Memory in Vision-Language-Action Models for Robotic Manipulation, August 2025. arXiv:2508.19236 [cs].

[Shukor *et al.*, 2025] 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, June 2025. arXiv:2506.01844 [cs].

[Team *et al.*, 2024] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An Open-Source Generalist Robot Policy, May 2024. arXiv:2405.12213 [cs].

[Tian *et al.*, 2024] Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive Inverse Dynamics Models are Scalable Learners for Robotic Manipulation, December 2024. arXiv:2412.15109 [cs].

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[Wang *et al.*, 2025] Yihao Wang, Pengxiang Ding, Lingxiao Li, Can Cui, Zirui Ge, Xinyang Tong, Wenxuan Song, Han Zhao, Wei Zhao, Pengxu Hou, et al. VLA-Adapter: An Effective Paradigm for Tiny-Scale Vision-Language-Action Model, September 2025. arXiv:2509.09372 [cs].

[Yang *et al.*, 2025] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, and Jianfeng Gao. Magma: A foundation model for multimodal ai agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14203–14214, June 2025.

[Zhang *et al.*, 2025a] Dapeng Zhang, Jing Sun, Chenghui Hu, Xiaoyan Wu, Zhenlong Yuan, Rui Zhou, Fei Shen, and Qingguo Zhou. Pure Vision Language Action (VLA) Models: A Comprehensive Survey, November 2025. arXiv:2509.19012 [cs].

[Zhang *et al.*, 2025b] Jiahui Zhang, Yurui Chen, Yueming Xu, Ze Huang, Yanpeng Zhou, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, and Li Zhang. 4D-VLA: Spatiotemporal Vision-Language-Action Pretraining with Cross-Scene Calibration, November 2025. arXiv:2506.22242 [cs].

[Zhao *et al.*, 2023] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware, April 2023. arXiv:2304.13705 [cs].

[Zhao *et al.*, 2025] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1702–1713, June 2025.

[Zheng *et al.*, 2025] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé, Andrey Kolobov, Furong Huang, and Jianwei Yang. TraceVLA: Visual Trace Prompting Enhances Spatial-Temporal Awareness for Generalist Robotic Policies, June 2025. arXiv:2412.10345 [cs].

[Zhong *et al.*, 2025] Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, and Haoang Li. FlowVLA: Visual Chain of Thought-based Motion Reasoning for Vision-Language-Action Models, October 2025. arXiv:2508.18269 [cs].

[Zitkovich *et al.*, 2023] Brianna Zitkovich, Tianhe Yu, Sichun Xu, et al. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. In *Proceedings of The 7th Conference on Robot Learning*, pages 2165–2183. PMLR, December 2023.