Look Before Acting: Enhancing Vision Foundation Representations for Vision-Language-Action Models

Yulin Luo[1*], Hao Chen[3*,†], Zhuangzhe Wu[1*], Bowen Sui[1*], Jiaming Liu[1*,†], Chenyang Gu[1], Zhuoyang Liu[1], Qiuxuan Feng[1], Jiale Yu[1], Shuo Gu[2], Peng Jia[2], Pheng-Ann Heng[3], Shanghang Zhang[1,✉]

[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University; [2]Simplexity Robotics; [3]The Chinese University of Hong Kong

[*] Equal contribution, [†] Project lead, ✉ Corresponding author

Project page: https://deepvision-vla.github.io/

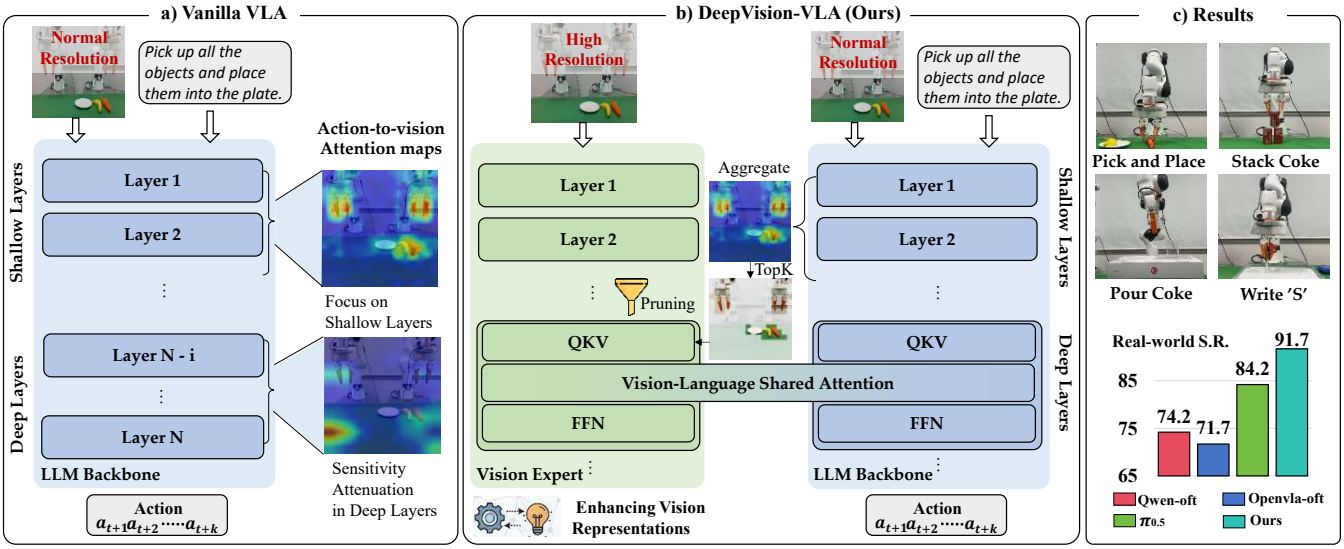


Figure 1: Overview. (a) In vanilla VLA models, attention to task-relevant visual tokens gradually weakens in deeper layers, leading to reduced sensitivity to visual information. (b) To address this issue, DeepVision-VLA introduces a Vision-Language Mixture-of-Transformers framework that injects multi-level visual features from a vision foundation model into deeper layers of the VLA backbone, strengthening visual representations for precise and complex manipulation. (c) As a result, DeepVision-VLA achieves superior performance across several real-world manipulation tasks.

Abstract

Vision-Language-Action (VLA) models have recently emerged as a promising paradigm for robotic manipulation, in which reliable action prediction critically depends on accurately interpreting and integrating visual observations conditioned on language instructions. Although recent works have sought to enhance the visual capabilities of VLA models, most approaches treat the LLM backbone as a black box, providing limited insight into how visual information is grounded into action generation. Therefore, we perform a systematic analysis of multiple VLA models across different action-generation paradigms and observe that sensitivity to visual tokens progressively decreases in deeper layers during action generation. Motivated by this observation, we propose **DeepVision-VLA**, built on a **Vision-Language Mixture-of-Transformers (VL-MoT)** framework. This framework enables shared attention between the vision foundation model and the VLA backbone, injecting multi-level visual features from the vision expert into deeper layers of the VLA backbone to enhance visual representations for precise and complex manipulation. In addition, we introduce **Action-Guided Visual Pruning (AGVP)**, which leverages shallow-layer attention to prune irrelevant visual tokens while preserving task-relevant ones, reinforcing critical visual cues for manipulation with minimal computational overhead. DeepVision-VLA outperforms prior state-of-the-art methods by 9.0% and 7.5% on simulated and real-world tasks, respectively, providing new insights for the design of visually enhanced VLA models.

1 Introduction

Driven by training on massive, internet-scale multimodal corpora [1, 26, 28, 29, 42], recent advances in Vision-Language Models (VLMs) have demonstrated exceptional proficiency in perception, reasoning, and instruction following [2, 4, 12, 22, 39]. Capitalizing on these strengths, Vision-Language-Action (VLA) models extend

these capabilities to robotics by directly mapping multimodal observations and natural language instructions to robot actions. Powered by billions of parameters and large-scale robot pre-training datasets [23, 40, 55], VLAs have exhibited remarkable potential for learning generalizable manipulation skills across diverse scenarios.

The robust control and generalization of VLA models are fundamentally contingent upon the precise interpretation and integration of visual observations [59]. Consequently, contemporary research has increasingly focused on fortifying the visual understanding and reasoning of VLAs. Existing approaches enhance the visual capabilities of VLA models from four main perspectives: (1) introducing visual prompts to improve the model's understanding of scenes and manipulated objects [18, 31, 49]; (2) designing auxiliary visual objectives to encourage the model to focus on task-critical entities [21, 46]; (3) incorporating additional visual modalities to provide complementary information [27, 49, 57, 61]; and (4) predicting future states to strengthen the model's ability to model the physical world [36, 37, 60]. Despite these advancements, existing paradigms typically treat the underlying Large Language Model (LLM) backbone in VLA models as a monolithic "black box", offering little insight into how visual information is propagated and utilized.

In this work, we move beyond treating the LLM backbone in VLA models as an opaque module and instead investigate how visual information is processed across its internal layers. Since the LLM is composed of stacked Transformer layers, a layer-wise analysis offers a natural and granular lens for understanding multimodal integration. We conduct a systematic investigation of several representative VLA architectures and action generation paradigms in two complementary stages. First, we qualitatively analyze action-to-visual attention maps and observe that while early layers maintain grounded attention on task-relevant objects, deeper layers often fail to focus effectively on these regions. To quantify the impact of this phenomenon on action generation, we introduce a layer-wise visual token dropout strategy. The results align with our qualitative findings: in deeper layers, action accuracy becomes increasingly insensitive to the masking of task-relevant visual regions, whereas shallow layers exhibit the opposite trend. We attribute this pattern to the prevalent serial architecture of current VLA models, where visual information is injected only at the first LLM layer and gradually attenuates as it propagates through Transformer layers.

Motivated by these findings, as illustrated in Figure 1, we introduce **Vision-Language Mixture-of-Transformers (VL-MoT)**, a novel VLA framework designed to enhance the sensitivity of deeper layers to task-relevant visual regions, thereby improving action prediction. In addition to the original visual encoder, our framework incorporates a dedicated **Vision Expert** (DINOv3, 0.8B) whose representations are fused with the VLA backbone via a MoT mechanism. Building on our observation that shallow layers naturally maintain effective visual grounding, we adopt an integration scheme that selectively couples the Vision Expert only with deeper VLA layers. Specifically, we extract multi-level visual features by sampling from the last few Transformer layers of the Vision Expert. This strategy empirically outperforms alternative integration approaches, such as sampling from the early layers or uniformly across all layers. A plausible explanation is that the later layers of the Vision Expert capture higher-level, semantically rich representations that are more invariant and object-centric, making them more compatible with task-relevant, action-conditioned features in the VLA model. Through this targeted integration, the Vision Expert collaborates with deeper VLA layers to reinforce action generation where the model is most susceptible to visual degradation, enabling more reliable and visually grounded robotic control.

However, naively integrating the full feature maps from the Vision Expert into the VLA backbone may introduce significant redundancy and irrelevant background information, potentially diluting task-critical signals. To address this, we propose an **Action-Guided Visual Pruning (AGVP)** strategy that refines information flow within our framework. Specifically, we leverage the robust grounding capabilities of shallow VLA layers to compute a saliency map by averaging attention weights from action tokens to visual tokens. This map identifies the most task-relevant regions, which we then use to prune the Query, Key, and Value across the Vision Expert's Transformer layers before integrating them into the deep VLA layers. This targeted pruning not only mitigates visual redundancy but also enables the Vision Expert to process higher-resolution inputs with minimal computational overhead. Our empirical results show that the increased visual granularity, focusing precisely on action-critical entities, leads to more stable manipulation.

We instantiate our framework as **DeepVision-VLA**, built upon a custom baseline, QwenVLA-OFT. This baseline leverages a Qwen3-VL backbone (4B) and adopts parallel action decoding with L1 regression output [24]. We systematically evaluate DeepVision-VLA across ten simulated tasks in RLBench [20] and four complex dual-arm real-world manipulation tasks. DeepVision-VLA achieves state-of-the-art (SOTA) performance, outperforming prior VLA methods by 9.0% in simulated settings and 7.5% in real-world settings. Our contributions are summarized as follows:

- We systematically analyze visual information utilized in current VLA models and identify a phenomenon where deeper LLM backbones become insensitive to task-relevant visual regions.
- We propose Vision-Language Mixture-of-Transformers (VL-MoT), a novel framework that improves action prediction by injecting multi-level visual features from a Vision Expert into deep VLA layers.
- We introduce Action-Guided Visual Pruning, a strategy that filters redundant visual information from the Vision Expert, providing deeper VLA with task-relevant visual signals.
- DeepVision-VLA establishes SOTA results in both simulation and real-world settings, demonstrating the effectiveness of our framework and providing more insights for the design of visually enhanced VLA models.

2 Related Work

Vision-Language-Action (VLA) Models. VLA models [3, 5, 6, 11, 19, 25, 33, 33, 34, 53, 54] are primarily driven by scaling robot demonstration data [8, 23, 40, 55] and adapting pretrained vision-language models (VLMs) [2, 12, 22, 39] for robotic control. These approaches directly model action sequences from visual observations and language instructions, demonstrating strong scalability and significant potential for generalization. Early work attempted to leverage the autoregressive capabilities of pretrained VLMs to generate robot

actions token by token [3, 7, 25, 62]. However, such formulations often suffer from action discontinuities and low execution frequency. Inspired by the success of diffusion policies [13, 58], recent efforts have explored diffusion-based [30, 33, 54] and flow-based VLA [6, 19] frameworks, which leverage the strong representation power of VLMs while introducing a dedicated action head to learn smooth and stable continuous action outputs. To further improve execution efficiency, several works [9, 11, 14] adopt a dual-system design, where a reasoning module is responsible for task planning, while a control module focuses on action generation. In addition, hierarchical architectures [43, 43] have been proposed to better handle high-level and abstract human instructions, typically leveraging an auxiliary VLM to decompose complex instructions into subgoals that guide the VLA for downstream instruction-following action generation. However, recent studies [21, 46, 59] suggest that when VLA models do not sufficiently develop their visual understanding during training, their action modeling performance can degrade noticeably, which may hinder precise manipulation in dynamic or cluttered environments. These findings suggest that preserving reliable, task-aware visual representations can be crucial for effective action generation [15, 50, 56].

Vision Improvement for VLA. As precise action generation critically depends on robust visual understanding and grounding [46, 59], a growing body of work has explored strategies to enhance VLA models from a visual perspective. One line of research augments the VLA input with additional visual cues to facilitate task comprehension, such as overlaying execution trajectories [18, 31] or highlighting target objects [17], demonstrating that simple prompt engineering can be surprisingly effective. Another approach introduces auxiliary visual supervision to encourage the model to attend to important image regions, for example, by reconstructing key objects in image [46] or anchoring the VLA's visual representations to strong teacher features [21], thereby improving the reliability of action generation. Beyond 2D inputs, incorporating richer visual modalities such as depth maps [32, 51, 57], 3D point clouds [27, 41, 61], or hand-drawn sketches [49] provides complementary spatial and geometric information, enabling the model to better reason about object shapes, distances, and occlusions. Finally, several methods adopt a reasoning-before-action paradigm [10, 36, 60], predicting future states or images to strengthen the model's understanding of physical dynamics, thereby enhancing the accuracy of manipulation. Despite these advancements, existing paradigms often treat the visual processing within VLA models as a black box, focusing primarily on the input state and the resulting actions while largely overlooking how visual information is internally utilized. In this work, we provide insights into this process and propose DeepVision-VLA, which integrates multi-level features from a visual foundation model into the deeper layers of the VLA via a Vision-Language Mixture-of-Transformers architecture, enhancing attention to task-relevant objects and improving action generation accuracy.

3 Methods

In this section, we first introduce the problem formulation and the general architecture of Vision-Language-Action (VLA) models in Sec. 3.1. Next, Sec. 3.2 analyzes how representative VLA architectures process and utilize visual information, providing key insights into their limitations. Building on this analysis, we propose the **Vision-Language Mixture-of-Transformers (VL-MoT)** framework, which improves action prediction performance by injecting multi-level Vision Expert knowledge into the deep layers of VLA models. Sec. 3.3 presents **DeepVision-VLA**, a concrete instantiation of this framework, and introduces the **Action-Guided Vision Pruning (AGVP)** strategy to further enhance the model's focus on task-relevant visual regions. The overall framework is shown in Figure 3. Finally, Sec. 3.4 describes the training and inference procedures of DeepVision-VLA.

3.1 Preliminaries

Problem Formulation. We consider a standard imitation learning setting for vision-language robotic manipulation. Given a dataset of expert demonstrations $\mathcal{D} = \{(\tau_i, l_i)\}_{i=1}^{N}$, where l_i denotes the language instruction and $\tau_i = \{(o_t, a_t)\}_{t=1}^{T_i}$ represents the corresponding trajectory consisting of visual observations and actions, the goal is to learn a policy π_θ that maps visual observations and task instructions to robot actions. At each time step t, the policy takes the current observation o_t together with the instruction l as input and predicts robot actions. Depending on the action representation, the policy may either predict the current action a_t or a short horizon of future actions $a_{t:t+H-1}$ under an action chunking formulation. Formally, this can be written as $\pi_\theta(a_t \mid o_{\le t}, l)$ or $\pi_\theta(a_{t:t+H-1} \mid o_{\le t}, l)$. The policy parameters are learned from demonstrations by solving

$$\theta^* = \arg\min_\theta \mathbb{E}_{(\tau,l)\sim\mathcal{D}} \left[\sum_{t=1}^{T} \ell\big(\pi_\theta(o_{\le t}, l), a_t\big) \right], \tag{1}$$

where $\ell(\cdot,\cdot)$ denotes the task-specific action supervision objective.

Vision-Language-Action Models. A typical VLA model consists of three primary components: a visual encoder, an LLM backbone, and an action decoder. The visual encoder extracts visual features from input observations, while the LLM backbone integrates these visual representations with the language instruction to perform multimodal reasoning and action conditioning. Finally, the action decoder maps the resulting representations to the robot action space for execution. Existing VLA models mainly differ in how actions are represented and learned. For instance, OpenVLA formulates action generation as *next-token prediction*, where discretized actions are generated autoregressively. In contrast, π_0 adopts *flow matching* to model continuous actions, while OpenVLA-OFT employs *parallel decoding*, predicting multiple future actions simultaneously using bidirectional attention for efficient chunk-level prediction. Despite these differences, all approaches rely on the LLM's ability to effectively interpret both visual observations and language instructions, which is critical for accurate action prediction. In this work, we focus on investigating how visual information influences the action prediction performance of VLA models.

3.2 Probing the Role of Vision in VLA Models

Given that the LLM backbone in VLAs is composed of stacked Transformer layers, we conduct a layer-wise investigation to understand how visual information is processed throughout action

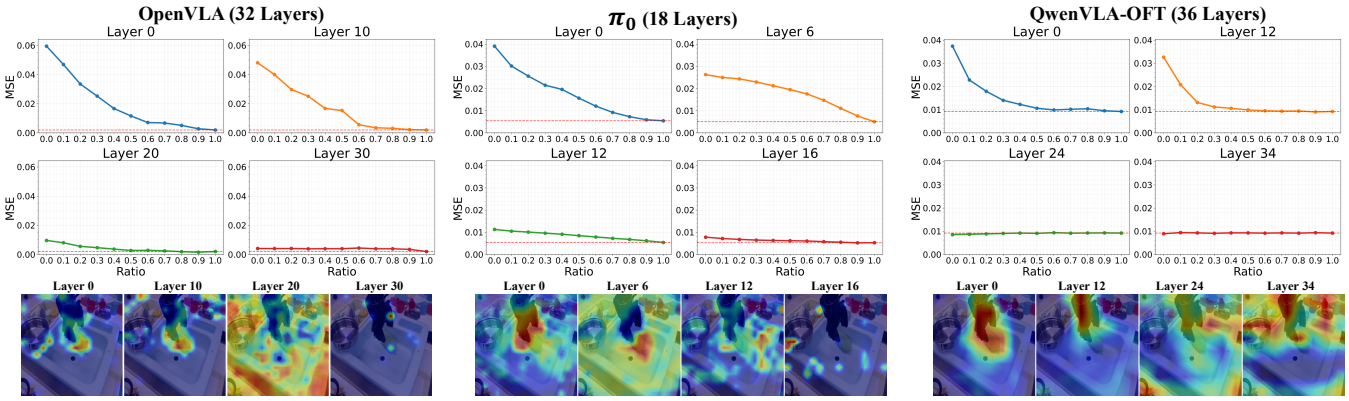


Figure 2: Layer-wise analysis of visual grounding in VLA models. Top: Effect of masking ROI visual tokens on action prediction error (MSE) at different layers. Masking in shallow layers significantly degrades performance, while the impact diminishes in deeper layers. Bottom: Action-to-vision attention maps across layers. Attention is concentrated on task-relevant regions in shallow layers but becomes increasingly diffuse in deeper layers, indicating weakened visual grounding.

prediction. Specifically, we first analyze action-to-vision attention maps to determine whether the model effectively attends to task-relevant objects in the scene, as such grounding provides a measure of reliable action learning. Next, to quantitatively evaluate the contribution of visual tokens to action performance, we perform a controlled, layer-wise ablation by masking critical visual tokens and observing the corresponding changes in action prediction accuracy. In the following, we describe our experimental setup and present the observed results along with their analysis.

Experimental Setup. We evaluate three representative VLA models that differ in their LLM backbones, model depths, and action generation paradigms: OpenVLA [25], π_0 [6], and a custom baseline QwenVLA-OFT, which adopts Qwen3-VL [2] as the backbone and performs parallel action prediction with an ℓ_1 regression objective. Our analysis is conducted on 1,500 randomly sampled trajectories from the BridgeV2 [52] dataset, which offers high-quality manipulation demonstrations with clear object layouts and consistent visual observations, making it particularly suitable for studying object-level visual sensitivity. For each image, we employ Grounding-DINO [35] to localize the regions of interest (ROIs), including the robot arm, the manipulated object, and their interaction area.

Action-to-Vision Attention. To better understand how VLA models ground action predictions in visual information, we visualize the attention from each action token to all visual tokens across LLM layers. At each layer, we average the attention weights from all action tokens to the visual tokens, producing a layer-wise action-to-vision attention map that captures how strongly the model relies on visual information at that depth. As shown in Figure 2 (bottom), all three VLA paradigms exhibit a consistent pattern: in shallow layers, action tokens attend predominantly to the visual ROIs localized for the task, including the manipulated object and the robot arm. In contrast, in deeper layers, attention becomes increasingly diffuse and extends to less relevant regions, indicating that the grounding of action tokens in visual information progressively diminishes along the LLM backbone.

Action Prediction Sensitivity to ROI Visual Tokens. While attention visualization provides qualitative insights into how models attend to visual regions, it does not directly quantify how much action prediction actually depends on those regions. To more rigorously measure the contribution of task-relevant visual information, we perform a layer-wise masking study on visual tokens corresponding to the ROIs. Specifically, for a selected layer in LLM, we identify the visual tokens associated with the ROIs and zero out a fraction r of them, effectively removing their information from the model, while keeping all non-ROI tokens unchanged. The resulting hidden states are then propagated through the remaining layers to produce the final action prediction. We measure the effect of this intervention using the mean squared error (MSE) between the predicted actions and the ground-truth actions. Since the masking is restricted to ROI tokens at a specific layer, the resulting change in performance directly reflects the extent to which grounded visual information contributes to action prediction at that depth. As shown in Figure 2 (top), all three models exhibit a consistent layer-wise pattern. Masking ROI tokens in early layers leads to a substantial increase in action MSE, indicating that these layers rely heavily on task-relevant visual information. In contrast, the effect of masking progressively decreases in deeper layers, and even the complete removal of ROI tokens in the deepest layers causes only minor changes in action prediction. These results indicate that task-relevant visual cues become increasingly underutilized in deeper layers, which partially undermines action reliability.

3.3 DeepVision-VLA

Based on the above analysis and the insight that increasing the sensitivity of deep VLA layers to task-relevant visual ROIs may improve action prediction accuracy, we propose the Vision-Language Mixture-of-Transformers (VL-MoT) framework. To facilitate understanding, we illustrate how the framework is applied to the QwenVLA-OFT baseline to construct DeepVision-VLA, as shown in Figure 3 (a). In this section, we begin by introducing the baseline model, followed by the architectural design of AVL-MoT, and finally present the Action-Guided Vision Pruning (AGVP) strategy.

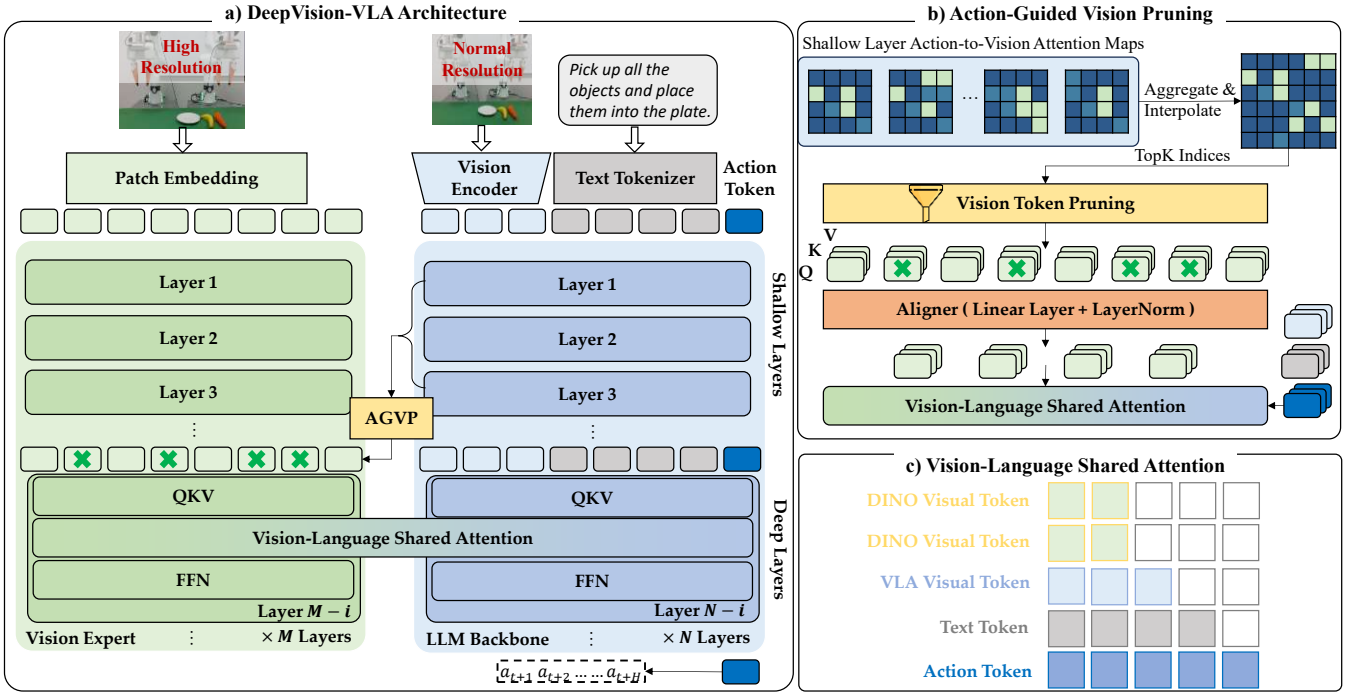


Figure 3: Framework. (a) A high-resolution Vision Expert is coupled with the LLM backbone through the proposed Vision–Language Mixture-of-Transformers (VL-MoT) framework, where deep LLM layers share attention with the Vision Expert to enhance visual grounding for action prediction. (b) Action-to-vision attention from shallow LLM layers is aggregated to identify task-relevant regions, which are used to prune Vision Expert tokens before fusion. (c) Vision Expert tokens use bidirectional attention to preserve their pretrained knowledge. VLA tokens apply causal attention to prompts and bidirectional attention to action tokens for parallel prediction.

3.3.1 ***QwenVLA-OFT Baseline.*** The baseline model is built upon Qwen3-VL (4B) [2] and adopts the parallel action prediction paradigm with an ℓ_1 regression objective as proposed in OpenVLA-OFT [24]. Different from OpenVLA-OFT, which applies bidirectional attention to all tokens, we restrict bidirectional attention to action tokens while keeping causal attention for prompt tokens to better preserve the pretrained behavior of Qwen3-VL. QwenVLA-OFT consists of three main components: a visual encoder, an LLM backbone, and an action decoder.

Visual Encoder. QwenVLA-OFT utilizes SigLIP2-Large (0.3B) as its visual encoder to extract expressive image representations. The encoder employs 2D-RoPE with interpolated absolute positional embeddings to provide spatial information for visual tokens. Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, it is partitioned into patches to produce dense visual features. To reduce token count and improve computational efficiency, neighboring 2×2 features are merged into a single token and projected to the LLM hidden dimension d via a two-layer MLP, resulting in $\mathbf{V} = E_{\text{vis}}(I) \in \mathbb{R}^{N_v\times d}$, where N_v is the number of visual tokens.

LLM Backbone. The visual features $\mathbf{V}$ are concatenated with tokenized language instructions and action tokens initialized as zero vectors to form the multimodal input sequence to the LLM. At layer ℓ, the hidden states are decomposed as $\mathbf{Z}^\ell = [\mathbf{Z}_V^\ell; \mathbf{Z}_L^\ell; \mathbf{Z}_A^\ell]$, where $\mathbf{Z}_V^\ell$, $\mathbf{Z}_L^\ell$, and $\mathbf{Z}_A^\ell$ denote the visual, language, and action embeddings, respectively. The LLM updates these hidden states layer by layer according to $\mathbf{Z}^\ell = \mathcal{F}^\ell(\mathbf{Z}^{\ell-1})$ for $\ell = 1, \ldots, L$, and the final action embedding $\mathbf{Z}_A^L$ is then fed to the action decoder.

Action Decoder. The action decoder maps the final action embeddings $\mathbf{Z}_A^L$ to the robot's action space using a two-layer MLP. Specifically, the predicted action vector $\mathbf{a} \in \mathbb{R}^n$, where n is the number of degrees of freedom (DoF) of the embodiment.

3.3.2 ***VL-MoT Framework.*** To enhance visual grounding in the deeper layers of the VLA, we introduce a **Vision Expert**, DINOv3. Its multi-level transformer features are injected into deep VLA layers via a shared-attention mechanism. This design allows the VLA to incorporate multi-level visual information into layers that would otherwise be insensitive to task-relevant visual ROIs, thereby improving the precision of action generation.

Vision Expert. We adopt the visual foundation model DINOv3 [45] because it provides spatially detailed representations [24, 25, 30] that are more fine-grained than the high-level semantic features typically extracted by existing VLA visual encoders, making it particularly suitable for precise manipulation tasks. Our approach, however, is not limited to this choice, and exploring alternative vision experts is left for future work.

VL-MoT Design. To integrate the multi-level knowledge of the Vision Expert into the deep VLA layers, we do not simply concatenate intermediate features and process them jointly through the LLM. Instead, we introduce a novel MoT architecture that directly

exposes the intermediate Query, Key, and Value (QKV) representations from the Vision Expert and integrates them with the corresponding QKV of the deep VLA layers via a shared-attention mechanism, as shown in Figure 3 (a). This enables cross-branch information exchange while preserving separate processing pathways, effectively reducing feature interference and stabilizing fusion training.

To align with the layers where visual grounding becomes less sensitive, the Vision Expert is connected only to the deepest n layers of the VLA, where n is determined based on the analysis in Sec. 3.2. For the Vision Expert, we select the last n transformer layers and use their QKV representations for feature injection. To formally describe this interaction, we consider a coupled layer between the Vision Expert and the LLM backbone. Here, $\mathbf{E}^k \in \mathbb{R}^{M \times d_e}$ represents the features from the k-th transformer layer of the Vision Expert, where M is the number of expert tokens and d_e is the expert hidden dimension. Similarly, $\mathbf{Z}^\ell \in \mathbb{R}^{N \times d}$ denotes the token representations of the LLM at layer ℓ, where N is the number of LLM tokens and d is the LLM hidden dimension. We then compute the QKV projections for the Vision Expert and the LLM tokens separately:

$$\begin{aligned} Q_E &= \mathbf{E}^k W_Q^E, \quad K_E = \mathbf{E}^k W_K^E, \quad V_E = \mathbf{E}^k W_V^E, \\ Q_Z &= \mathbf{Z}^\ell W_Q^Z, \quad K_Z = \mathbf{Z}^\ell W_K^Z, \quad V_Z = \mathbf{Z}^\ell W_V^Z. \end{aligned} \tag{2}$$

Since the hidden dimensions of the two branches may differ, the QKV representations of the Vision Expert tokens are each projected through a learnable linear layer to align with the LLM hidden dimension. The shared attention is then computed by concatenating the QKV representations from both branches:

$$Q = [Q_E; Q_Z], \quad K = [K_E; K_Z], \quad V = [V_E; V_Z]. \tag{3}$$

$$A = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right), \quad H = AV. \tag{4}$$

Finally, we split the output $H = [H_E; H_Z]$ back into the two branches and apply the remaining standard Transformer operations. Notably, we maintain the original bidirectional attention mechanism for visual expert tokens, as shown in Figure 3 (c), which better preserves pre-trained knowledge and provides more stable visual features for the VLA model.

Empirically, this VL-MoT configuration outperforms alternative feature selection strategies, such as using QKV features exclusively from the first n layers or uniformly sampling n layers across the entire Vision Expert. We attribute this design choice to the fact that the deeper layers of the Vision Expert encode high-level, semantically rich, and object-centric representations, which align naturally with the task-relevant, action-conditioned features in the VLA model and are particularly effective for fine-grained and complex manipulation.

3.3.3 ***Action-Guided Vision Pruning.*** To enable the VLA model to better focus on task-relevant objects, we propose Action-Guided Visual Pruning, as shown in Figure 3 (b), which filters redundant background features before integrating multi-level information from the Vision Expert into deeper VLA layers.

Based on our observation in Section 3.2 that shallow VLA layers provide reliable action-conditioned visual grounding, we use the attention from action tokens to visual tokens in these layers to identify ROI visual tokens. Let $\mathbf{A}^\ell \in \mathbb{R}^{N_a \times N_v}$ denote the action-to-vision attention map at layer ℓ, where N_a and N_v are the numbers of action and visual tokens, respectively. Since multiple action tokens jointly contribute to robot control, we first average over the action tokens as $\mathbf{m}^\ell = \frac{1}{N_a} \sum_{i=1}^{N_a} \mathbf{A}_{i,:}^\ell$ to obtain a per-layer attention map, and then aggregate over a set of shallow layers $\mathcal{L}_s$ to obtain the final attention map $\mathbf{m} = \frac{1}{|\mathcal{L}_s|} \sum_{\ell \in \mathcal{L}_s} \mathbf{m}^\ell$. This multi-layer averaging effectively filters out noise and produces stable task-relevant ROI visual tokens.

Compared with the input image used in the VLA, we leverage the strong feature extraction capability of the Vision Expert by feeding a higher-resolution image, allowing the model to capture more fine-grained object details. Consequently, we interpolate the obtained attention map to match the resolution of the Vision Expert. Denoting the interpolation operator by $\mathcal{I}(\cdot)$, the transferred attention map is $\tilde{\mathbf{m}} = \mathcal{I}(\mathbf{m}) \in \mathbb{R}^{N_d}$, where N_d is the number of Vision Expert image tokens. Since this attention map effectively indicates the importance of each region, we retain the top-K tokens with the highest importance, whose indices are given by $\mathcal{S}_K = \mathrm{TopK}(\tilde{\mathbf{m}}, K)$, and the corresponding tokens for sharing attention with deep VLA layers are $\bar{\mathbf{E}}^k = \mathbf{E}^k[\mathcal{S}_K]$. This approach not only effectively injects critical visual region information into deep VLA layers but also controls the computational budget while exploiting the high-resolution input of the Vision Expert.

3.4 Training and Inference

Training Recipe. Before pretraining DeepVision-VLA, we initialize the model with pretrained weights from Qwen3-VL [2] and DINOv3 [45], and then train the entire architecture in an end-to-end manner. Following prior work [11, 17, 33, 36], we curate a specialized pretraining dataset by carefully processing and filtering several large-scale cross-embodiment datasets, including Open X-Embodiment [40], DROID [23], and RoboMIND [55]. The resulting dataset contains over 400K trajectories, and DeepVision-VLA is trained on this dataset for one epoch. The fine-tuning settings for downstream tasks are described in the experimental section, as they differ between simulation and real-world environments.

Inference. At inference time, DeepVision-VLA first takes the current image observation and language instruction as input. These inputs are propagated through the shallow layers of the VLA, where action-to-vision attention maps are computed to identify task-relevant visual regions. Based on these cues, we prune the multi-level features from the Vision Expert and integrate them into the deeper VLA layers through the VL-MoT mechanism. Finally, after the deep layers complete the forward pass, the hidden states of the action tokens are fed into the action decoder to generate the final actions. Notably, this procedure for identifying key visual regions and enhancing VLA perception requires no additional annotations or external supervision, and the entire pipeline remains end-to-end executable after training.

4 Experiments

In Section 4.1, we systematically evaluate the manipulation capability of our proposed DeepVision-VLA in simulated environments. Subsequently, Section 4.2 presents an ablation study to validate the essential roles of the VL-MoT framework and AGVP strategy.

Table 1: Comparison of DeepVision-VLA and baselines on RLBench. All methods are trained in the multi-task setting [44], and we report mean success rates (S.R.).

Models	Close box	Close laptop lid	Toilet seat down	Sweep to dustpan	Close fridge	Phone on base	Umbrella out	Frame off hanger	Wine at rack	Water plants	Mean S.R. ↑
OpenVLA	0.60	0.35	0.75	0.55	0.85	0.20	0.30	0.15	0.20	0.05	0.40
SpatialVLA	0.80	0.70	0.85	0.20	0.80	0.15	0.25	0.40	0.15	0.30	0.46
CogACT	0.90	0.80	0.95	0.50	0.85	0.50	0.55	0.45	0.30	0.25	0.61
CoT-VLA	0.95	0.75	**1.00**	0.80	0.65	0.50	0.40	0.50	0.55	**0.50**	0.66
$\pi_{0.5}$	0.90	**0.95**	0.85	0.75	**1.00**	0.05	0.10	**0.80**	0.75	0.35	0.65
HybridVLA	0.85	**0.95**	**1.00**	0.90	**1.00**	0.50	0.50	0.70	0.50	**0.50**	0.74
QwenVLA-OFT	0.95	**0.95**	**1.00**	0.15	0.95	0.65	0.30	0.70	0.65	**0.50**	0.69
DeepVision-VLA	**1.00**	**0.95**	**1.00**	**0.95**	0.95	**0.75**	**0.65**	0.70	**0.85**	**0.50**	**0.83**

Finally, in Section 4.3, we further demonstrate the model's effectiveness through single-arm real-world manipulation tasks.

4.1 Simulation Experiment

Data Collection. We evaluate DeepVision-VLA across 10 manipulation tasks from the RLBench [20] benchmark based on the CoppeliaSim simulation environment, including 1) *Close box*, 2) *Close Laptop*, 3) *Toilet seat down*, 4) *Sweep to dustpan*, 5) *Close fridge*, 6) *Phone on base*, 7) *Take umbrella out*, 8) *Frame off hanger*, 9) *Wine at rack*, and 10) *Water plants*. All tasks are performed on a Franka Panda robot with a single front-view RGB observation. Following pre-defined waypoints and utilising the Open Motion Planning Library [48], we construct a training dataset in which each task comprises 100 trajectories, with the frame-sampling technique used in previous studies [44].

Training and evaluation protocol. We compare DeepVision-VLA against seven representative VLA baselines: OpenVLA [25], SpatialVLA [41], CogACT [30], CoT-VLA [60], $\pi_{0.5}$ [19], HybridVLA [33], and our custom QwenVLA-OFT. Each baseline is initialized with its officially released pretrained weights and fine-tuned end-to-end, strictly following the configurations and hyperparameters recommended in their original publications, except for QwenVLA-OFT, which we pretrain ourselves. The implementation details of QwenVLA-OFT are described in Section 3.3.1. For image processing, DeepVision-VLA adopts dual resolutions: 256×256 for the VLA branch and 512×512 for the Vision Expert, providing finer visual details to the Vision Expert branch. We fine-tune DeepVision-VLA for 300 epochs using the AdamW [38] optimizer on 8 NVIDIA H20 GPUs. Following standard evaluation protocols in prior works [16, 17], we report results from the final checkpoints across 20 rollout trials per task, repeated over three random seeds to account for stochastic variability.

Quantitative analysis. As shown in Table 1, DeepVision-VLA achieves an 83% mean success rate on RLBench, outperforming all baselines by a significant margin. It surpasses HybridVLA (74%), $\pi_{0.5}$ (65%), and CogACT (61%) by 9%, 18%, and 22%, respectively, and improves over our direct baseline QwenVLA-OFT (69%) by 14%, directly validating the effectiveness of our proposed Vision-Language Mixture-of-Transformers (VL-MoT) framework coupled with the Action-Guided Visual Pruning (AGVP) strategy. DeepVision-VLA attains the highest success rate on 8 out of 10 tasks, demonstrating strong robustness across diverse manipulation scenarios, with particularly large gains on visually challenging tasks. For instance, on *Sweep to Dustpan* and *Wine at Rack*, it improves over QwenVLA-OFT by 80% (0.95 vs. 0.15) and 31% (0.85 vs. 0.65), respectively. These improvements stem from its enhanced visual processing: by injecting multi-level DINOv3 features into deeper layers and pruning irrelevant visual tokens, the model maintains strong task-relevant visual grounding throughout the model, addressing a key limitation of standard VLAs and yielding substantial performance gains.

4.2 Ablation Study

To validate the key design choices of DeepVision-VLA, we conduct ablation experiments on four representative RLBench tasks: 1) *Close box*, 2) *Close laptop*, 3) *Sweep to dustpan*, and 4) *Phone on base*. These tasks are selected to cover both relatively simple and more challenging manipulation scenarios, while remaining representative and discriminative for evaluating visual grounding quality. All models are trained under the same setting and evaluated over 50 rollouts per task. We report the average success rate over the four tasks.

Importance of VL-MoT Architecture. We first study how different paradigms for incorporating Vision Expert features affect performance. Across the four evaluation tasks, the vanilla VLA baseline (QwenVLA-OFT) achieves a 65.5% success rate. Early fusion of DINOv3 features with the original visual tokens before projection improves performance to 73%, showing the benefit of external vision features, but remains limited by its shallow input-level integration. Aligning intermediate VLA features to frozen DINOv3 representations yields 67%, suggesting that preserving visual representations helps, though the aligned features are still generic and not optimized for manipulation. In contrast, our proposed VL-MoT framework achieves 88%, substantially outperforming all alternatives. This demonstrates that coupling the Vision Expert directly with deeper VLA layers, together with targeted feature pruning, is a more effective way to enhance action-relevant visual representations.

Vision Foundation Model Feature Selection. We next examine both the feature selection strategy and the choice of the Vision Foundation Model. Using only the first 16 layers of DINOv3 yields 61.5%, while uniformly sampling 16 layers across the whole model achieves 85%. The best result is obtained by using the last

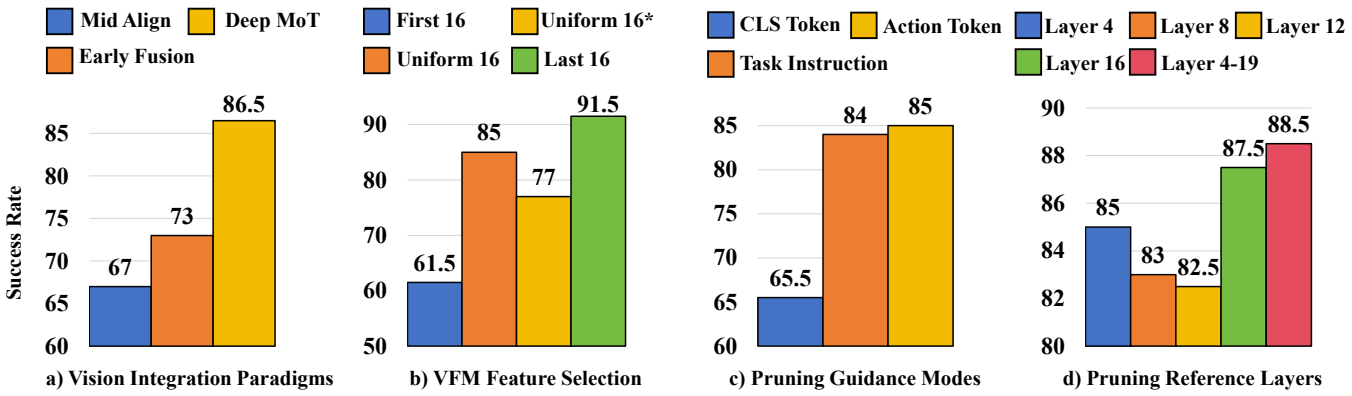


Figure 4: Ablation studies. (a) Comparison of different paradigms for integrating Vision Expert features into the VLA backbone. (b) Evaluation of different multi-level feature selection strategies from the Vision Expert. (c) Comparison of different guidance mechanisms for generating the visual pruning map in AGVP. (d) Analysis of different shallow-layer references used to compute the action-to-vision attention map for pruning.

Table 2: Comparison across real-world manipulation tasks. Step represents the atomic task within the overall task, and Avg denotes the average success rate.

Models	Stack coke cans	Write letter 'S'	Pick fruit to the plate		Pour coke to bottle		Avg
			Step 1 →	Step 2	Step 1 →	Step 2	
$\pi_{0.5}$	0.65	0.95	0.75	1.00	1.00	0.70	0.842
OpenVLA-OFT	0.50	0.85	0.70	0.80	0.75	0.70	0.717
QwenVLA-OFT (baseline)	0.50	0.80	0.85	0.90	0.70	0.70	0.742
DeepVision-VLA	**0.65**	**0.95**	**0.95**	**0.95**	**1.00**	**1.00**	**0.917**

16 layers, which reaches 88%, indicating that later DINOv3 features are the most effective for robotic manipulation. Replacing DINOv3 with SigLIP under the same uniformly sampled 16-layer setting reduces the success rate to 77%. This gap likely stems from their different pretraining objectives: SigLIP emphasizes image-text alignment, whereas DINOv3 provides stronger fine-grained spatial representations that are better suited for precise manipulation.

Guidance Mechanisms for Action-Guided Vision Pruning. We then evaluate different strategies for generating the guidance map used in AGVP. The baseline achieves 65.5%. Using the DINOv3 CLS token as the pruning cue brings no improvement, remaining at 65.5%, suggesting that global scene semantics alone are insufficient for manipulation-oriented feature selection. Using instruction-to-vision attention improves the success rate to 84%, showing that task-aware language guidance is beneficial, but its lack of reliable awareness of the robot arm and the arm-object interaction region limits its effectiveness for precise action prediction. Our action-to-vision attention guidance achieves the best result of 88%, indicating that shallow-layer action tokens provide more effective cues for identifying manipulation-relevant regions, as they naturally encode both task intent and action-conditioned visual grounding.

Impact of Reference Layers on AGVP. We analyze which shallow-layer attention maps provide the best pruning guidance. Using the attention map from the 4th, 8th, 12th, and 16th layers yields success rates of 85%, 69%, 82.5%, and 87.5%, respectively, with the 16th layer performing best among single-layer choices. Averaging attention maps across Layers 4–19 further improves performance to 88%, indicating that multi-layer guidance is more robust and alleviates noisy attention to irrelevant regions.

4.3 Real-World Experiment

Data Collection. We evaluate DeepVision-VLA on four complex single-arm real-world manipulation tasks using a Franka Research 3 robot equipped with an Intel RealSense D455 camera for RGB observations. The evaluated tasks encompass 1) *stack coke cans*, 2) *write letter 'S'*, 3) *pick fruit to the plate*, and 4) *pour coke to bottle*. To provide a more comprehensive assessment of model performance, Tasks 3s) and 4) are decomposed into two sequential steps. Task 3) is divided into a first step of picking and placing the banana, followed by a second step of picking and placing the carrot. Task 4) is evaluated independently on the initial grasp and the subsequent pouring action. Tasks 1) and 2) are defined as simple tasks and are assessed using a single comprehensive success score for the complete execution.

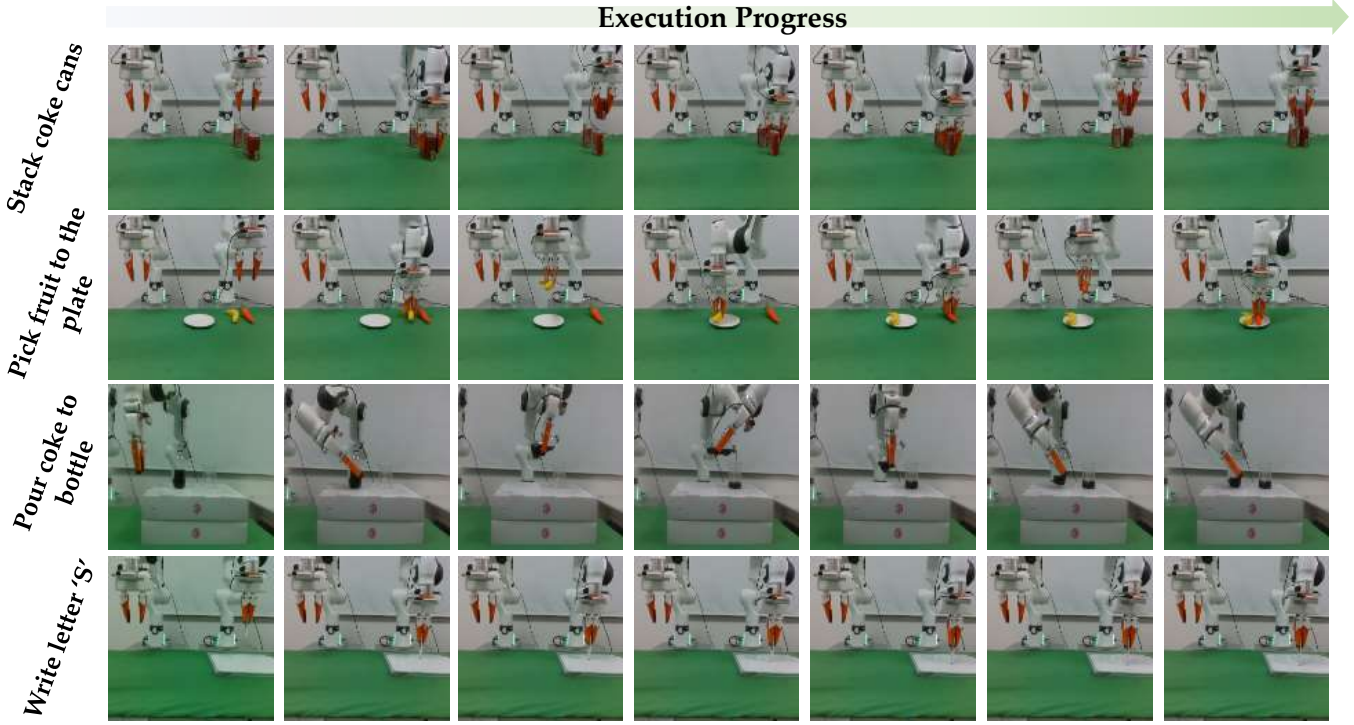


Figure 5: Visualization of task execution processes by real-world single-arm robots (from left to right).

Training and evaluation details. The training protocol for the DeepVision-VLA remains the same as in the simulation. We compare our method against three state-of-the-art VLA model baselines including $\pi_{0.5}$ [6], OpenVLA-OFT [24], and the custom QwenVLA-OFT [47]. For a fair evaluation, all baselines share the identical camera setup, with each task undergoing 20 rollouts under consistent test conditions.

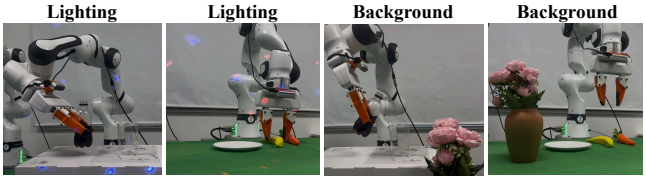
Quantitative and qualitative analysis. Tab. 2 shows that DeepVision-VLA achieves the highest overall performance across all real-world evaluations with an average success rate of 91.7%, effectively surpassing the strong baselines including $\pi_{0.5}$ (84.2%), QwenOFT (74.2%), and OpenVLA-OFT (71.7%), which demonstrates the highly robust physical interaction and precise manipulation capabilities of our DeepVision-VLA. For the two single-step tasks evaluated with a comprehensive score, DeepVision-VLA attains success rates of 65% for stacking coke cans and 95% for writing the letter 'S'. The whiteboard writing task particularly highlights the strengths of our architecture, as the robot must precisely draw instructed shapes requiring continuous, high-precision spatial tracking. Furthermore, DeepVision-VLA excels in multi-stage tasks evaluated via granular sub-stage metrics. During the "pick fruit to the plate" task, it maintained a consistent 95% success rate across both Step 1 and Step 2. More notably, in the "pour coke to bottle" task, our model achieved a perfect 100% success rate across both stages, outperforming the strongest baseline ($\pi_{0.5}$), which dropped to 70% in the second step. These multi-stage tasks necessitate strict visual focus on object boundaries and relative positioning. We attribute these improvements to our VL-MoT framework and AGVP strategy, which facilitate the integration of hierarchical visual features from the Vision Expert into the deeper layers of the VLA. This design enables the model to attend to critical object regions, leading to more accurate action generation. Figure 5 visualizes the execution process of DeepVision-VLA on real-world tasks. The smooth manipulation and successful task completion demonstrate that our model effectively attends to task-critical visual objects. For instance, the 100% success rate on the "pour Coke to Bottle" task highlights the model's robust object localization and execution stability.

4.4 Generalization Experiment

To evaluate the zero-shot generalization capacity of DeepVision-VLA, we conducted a series of rigorous experiments across two distinct environmental perturbations, comparing our model against the baseline QwenVLA-OFT. We focused on two representative tasks: the multi-step Pick fruit and the multi-step Pour coke.

Unseen background By introducingunseen elements (e.g., floral arrangements) into the workspace, we challenged the models' ability to distinguish task-relevant features from environmental noise. As reported in the "Background" row of Table **??**, while the baseline experienced a noticeable performance decay, DeepVision-VLA maintained high success rates (e.g., 0.90 in Pick fruit Step 1). This resilience stems from our visual enhancement mechanism, which enables the model to perform high-level scene reasoning and decouple the manipulated object from its surroundings.

Unseen lighting conditions The results under "Lighting" further validate our model's superiority in visual robustness. While baseline methods often struggle with altered shadow patterns and



Task	Pick fruit (Step 1)		Pick fruit (Step 2)	
Scenario	QwenVLA-OFT	**Ours**	QwenVLA-OFT	**Ours**
Original	0.85	**0.95**	0.90	**0.95**
Background	0.70 (-18%)	**0.90 (-5%)**	0.80 (-11%)	**0.90 (-5%)**
Lighting	0.70 (-18%)	**0.80 (-16%)**	0.70 (-22%)	**0.90 (-5%)**

Table 3: Generalization Scenarios. The visualizations show unseen test conditions, where Background and Lighting denote novel environmental layouts and varied illumination conditions, respectively. DeepVision-VLA exhibits robust visual enhancement, maintaining precise manipulation under these perturbations.

non-uniform illumination, DeepVision-VLA maintains near-perfect success rates (e.g., 1.00 in Pour coke). This suggests that our visual enhancement strategy effectively achieves a degree of photometric invariance, allowing the model to extract consistent structural information even when the raw pixel intensity fluctuates significantly.

5 Conclusion

In this work, we investigate the role of visual representations in Vision-Language-Action (VLA) models for robotic manipulation. Through a systematic analysis across multiple VLA architectures and action-generation paradigms, we observe that sensitivity to visual tokens progressively diminishes in deeper layers during action generation, limiting the effective utilization of visual information. Motivated by this finding, we propose DeepVision-VLA, a framework built upon a Vision-Language Mixture-of-Transformers (VL-MoT) architecture that introduces shared attention between a vision foundation model and the VLA backbone. By injecting multi-level visual features into deeper layers, DeepVision-VLA enhances visual representations and improves the grounding of perception in action generation. To further strengthen task-relevant visual cues, we introduce Action-Guided Visual Pruning (AGVP), which selectively removes irrelevant visual tokens based on shallow-layer attention while preserving critical visual information with minimal computational overhead. Extensive experiments on both simulated and real-world robotic manipulation tasks demonstrate that DeepVision-VLA consistently outperforms prior state-of-the-art methods. We hope this work provides new insights into the design of visually enhanced VLA models and inspires future research on integrating foundation vision models with robotic manipulation.

References

[1] Anas Awadalla, Le Xue, Oscar Lo, Manli Shu, Hannah Lee, Etash Guha, Sheng Shen, Mohamed Awadalla, Silvio Savarese, Caiming Xiong, et al. 2024. Mint-1t: Scaling open-source multimodal data by 10x: A multimodal dataset with one trillion tokens. *Advances in Neural Information Processing Systems* 37 (2024), 36805–36828.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631* (2025).

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

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

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

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. 2024. π_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164* (2024).

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817* (2022).

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

[9] Qingwen Bu, Hongyang Li, Li Chen, Jisong Cai, Jia Zeng, Heming Cui, Maoqing Yao, and Yu Qiao. 2024. Towards synergistic, generalized, and efficient dual-system for robotic manipulation. *arXiv preprint arXiv:2410.08001* (2024).

[10] Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. 2025. Worldvla: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539* (2025).

[11] Hao Chen, Jiaming Liu, Chenyang Gu, Zhuoyang Liu, Renrui Zhang, Xiaoqi Li, Xiao He, Yandong Guo, Chi-Wing Fu, Shanghang Zhang, et al. 2025. Fast-in-slow: A dual-system foundation model unifying fast manipulation within slow reasoning. *arXiv preprint arXiv:2506.01953* (2025).

[12] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. 2025. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811* (2025).

[13] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. 2025. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research* 44, 10-11 (2025), 1684–1704.

[14] Can Cui, Pengxiang Ding, Wenxuan Song, Shuanghao Bai, Xinyang Tong, Zirui Ge, Runze Suo, Wanqi Zhou, Yang Liu, Bofang Jia, et al. 2025. Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation. *arXiv preprint arXiv:2505.03912* (2025).

[15] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, et al. 2025. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626* (2025).

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

[17] Chenyang Gu, Jiaming Liu, Hao Chen, Runzhong Huang, Qingpo Wuwu, Zhuoyang Liu, Xiaoqi Li, Ying Li, Renrui Zhang, Peng Jia, Pheng-Ann Heng, and Shanghang Zhang. 2025. ManualVLA: A Unified VLA Model for Chain-of-Thought Manual Generation and Robotic Manipulation. arXiv:2512.02013 [cs.RO] https://arxiv.org/abs/2512.02013

[18] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. 2023. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv preprint arXiv:2311.01977* (2023).

[19] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. 2025. π_0.5: a Vision-Language-Action Model with Open-World Generalization. *arXiv preprint arXiv:2504.16054* (2025).

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

[21] Nikita Kachaev, Mikhail Kolosov, Daniil Zelezetsky, Alexey K Kovalev, and Aleksandr I Panov. 2025. Don't Blind Your VLA: Aligning Visual Representations for OOD Generalization. *arXiv preprint arXiv:2510.25616* (2025).

[22] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. 2024. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on*

Machine Learning.

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

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

[25] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. 2024. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246* (2024).

[26] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. 2024. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326* (2024).

[27] Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. 2026. Pointvla: Injecting the 3d world into vision-language-action models. *IEEE Robotics and Automation Letters* 11, 3 (2026), 2506–2513.

[28] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895* (2024).

[29] Qingyun Li, Zhe Chen, Weiyun Wang, Wenhai Wang, Shenglong Ye, Zhenjiang Jin, Guanzhou Chen, Yinan He, Zhangwei Gao, Erfei Cui, et al. 2024. Omnicorpus: A unified multimodal corpus of 10 billion-level images interleaved with text. *arXiv preprint arXiv:2406.08418* (2024).

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

[31] Xiaoqi Li, Jingyun Xu, Mingxu Zhang, Jiaming Liu, Yan Shen, Iaroslav Ponomarenko, Jiahui Xu, Liang Heng, Siyuan Huang, Shanghang Zhang, et al. 2025. Object-centric prompt-driven vision-language-action model for robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 27638–27648.

[32] Yixuan Li, Yuhui Chen, Mingcai Zhou, Haoran Li, Zhengtao Zhang, and Dongbin Zhao. 2025. QDepth-VLA: quantized depth prediction as auxiliary supervision for vision-language-action models. *arXiv preprint arXiv:2510.14836* (2025).

[33] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. 2025. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631* (2025).

[34] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. 2024. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864* (2024).

[35] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. 2024. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*. Springer, 38–55.

[36] Zhuoyang Liu, Jiaming Liu, Hao Chen, Jiale Yu, Ziyu Guo, Chengkai Hou, Chenyang Gu, Xiangju Mi, Renrui Zhang, Kun Wu, et al. 2026. LaST _{0}: Latent Spatio-Temporal Chain-of-Thought for Robotic Vision-Language-Action Model. *arXiv preprint arXiv:2601.05248* (2026).

[37] Zhuoyang Liu, Jiaming Liu, Jiadong Xu, Nuowei Han, Chenyang Gu, Hao Chen, Kaichen Zhou, Renrui Zhang, Kai Chin Hsieh, Kun Wu, et al. 2025. Mla: A multisensory language-action model for multimodal understanding and forecasting in robotic manipulation. *arXiv preprint arXiv:2509.26642* (2025).

[38] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).

[39] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. 2024. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525* (2024).

[40] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. 2024. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 6892–6903.

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

[42] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. 2022. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems* 35 (2022), 25278–25294.

[43] Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. 2025. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417* (2025).

[44] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. 2022. Perceiver-Actor: A Multi-Task Transformer for Robotic Manipulation. In *Proceedings of the 6th Conference on Robot Learning (CoRL)*.

[45] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. 2025. Dinov3. *arXiv preprint arXiv:2508.10104* (2025).

[46] Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. 2025. Reconvla: Reconstructive vision-language-action model as effective robot perceiver. *arXiv preprint arXiv:2508.10333* (2025).

[47] starVLA Contributors. 2025. StarVLA: A Lego-like Codebase for Vision-Language-Action Model Developing. GitHub repository. doi:10.5281/zenodo.18264214

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

[49] Priya Sundaresan, Quan Vuong, Jiayuan Gu, Peng Xu, Ted Xiao, Sean Kirmani, Tianhe Yu, Michael Stark, Ajinkya Jain, Karol Hausman, et al. 2024. Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches. In *8th Annual Conference on Robot Learning*.

[50] Yihe Tang, Wenlong Huang, Yingke Wang, Chengshu Li, Roy Yuan, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. 2025. Uad: Unsupervised affordance distillation for generalization in robotic manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3822–3831.

[51] Yalcin Tur, Jalal Naghiyev, Haoquan Fang, Wei-Chuan Tsai, Jiafei Duan, Dieter Fox, and Ranjay Krishna. 2026. Recurrent-Depth VLA: Implicit Test-Time Compute Scaling of Vision-Language-Action Models via Latent Iterative Reasoning. *arXiv preprint arXiv:2602.07845* (2026).

[52] 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. 2023. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*. PMLR, 1723–1736.

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

[54] Junjie Wen, Yichen Zhu, Minjie Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Xiaoyu Liu, Chaomin Shen, Yaxin Peng, and Feifei Feng. 2025. DiffusionVLA: Scaling Robot Foundation Models via Unified Diffusion and Autoregression. In *Forty-second International Conference on Machine Learning*. https://openreview.net/forum?id=VdwdU81Uzy

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

[56] Jiabing Yang, Yixiang Chen, Yuan Xu, Peiyan Li, Xiangnan Wu, Zichen Wen, Bowen Fang, Tao Yu, Zhengbo Zhang, Yingda Li, et al. 2026. UAOR: Uncertainty-aware Observation Reinjection for Vision-Language-Action Models. *arXiv preprint arXiv:2602.18020* (2026).

[57] Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. 2025. Depthvla: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375* (2025).

[58] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 2024. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954* (2024).

[59] Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. 2026. VLM4VLA: Revisiting Vision-Language-Models in Vision-Language-Action Models. *arXiv preprint arXiv:2601.03309* (2026).

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

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

[62] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*. PMLR, 2165–2183.