MoLe-VLA: Dynamic Layer-skipping Vision Language Action Model via Mixture-of-Layers for Efficient Robot Manipulation

Rongyu Zhang[1,2,3*], Menghang Dong[3*], Yuan Zhang[3], Liang Heng[3], Xiaowei Chi[4], Gaole Dai[3], Li Du[2], Dan Wang[4], Yuan Du[2†], Shanghang Zhang[3†]

[1]The Hong Kong Polytechnic University
[2]Nanjing University
[3]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University
[4]The Hong Kong University of Science and Technology
yuandu@nju.edu.cn, shanghang@pku.edu.cn

Abstract

Vision-Language-Action (VLA) models enable robotic systems to perform embodied tasks but face deployment challenges due to the high computational demands of the dense Large Language Models (LLMs), with existing early-exit-based sparsification methods often overlooking the critical semantic role of final layers in downstream tasks. Aligning with the recent breakthrough of the Shallow Brain Hypothesis (SBH) in neuroscience and the mixture of experts in model sparsification, we conceptualize each LLM layer as an expert and propose a **M**ixture-**o**f-**L**ay**e**rs **V**ision-**L**anguage-**A**ction model (***MoLe-VLA*** or simply ***MoLe***) architecture for dynamic LLM layer activation. Specifically, we introduce a *Spatial-Temporal Aware Router (STAR)* for MoLe to selectively activate only parts of the layers based on the robot's current state, mimicking the brain's distinct signal pathways specialized for cognition and causal reasoning. Additionally, to compensate for the cognition ability of LLM lost during the layer-skipping, we devise a *Cognitive self-Knowledge Distillation (CogKD)* to enhance the understanding of task demands and generate task-relevant action sequences by leveraging cognition features. Extensive experiments in RLBench simulations and real-world demonstrate the superiority of MoLe-VLA in both efficiency and performance, improving the mean success rate by 9.7% across ten simulation tasks while accelerating inference by 36.8% over OpenVLA.

Introduction

The rapid advancements in multimodal large language models (MLLMs) (Li et al. 2022; Alayrac et al. 2022; Radford et al. 2021b) have showcased their ability to integrate complex language and visual representations, inspiring the development of generalist robots and embodied agents capable of vision-language understanding, human interaction, and adaptive problem-solving in manipulation tasks. Early vision-language-action (VLA) models (Liu et al. 2024; Li et al. 2024a), such as RT-2 (Brohan et al. 2023) and OpenVLA (Kim et al. 2024), have demonstrated the feasibility of using MLLMs for end-to-end robotic control, enabling robust policies and emergent skills like generalization to unseen objects and novel commands. However, deploying MLLMs in real-world robotics poses significant challenges due to their high computational demands. For example, a 7B VLA model running on a commercial-grade GPU like the RTX 4090 generally achieves an inference frequency of 5-10 Hz, which falls significantly short of the 50-1000 Hz control frequency required by the Franka robotic arm.

Recent studies (Raposo et al. 2024; Dai et al. 2024b) have uncovered significant redundancy in LLM layers, particularly in robotic tasks, where homogeneous patterns across layers lead to high computational costs with limited performance gains. For instance, DeeR (Yue et al. 2024) demonstrated that using all 24 layers of the Flamingo (Li et al. 2023) model improves task success rates by only 3.2% compared to using six layers, while computational costs increase 4x on the Calvin LH-MTLC (Mees et al. 2022). Similarly, our analysis of OpenVLA with RLBench (James et al. 2020) simulator in Fig.1 (B) reveals that the cosine similarity between consecutive layer outputs exceeds *90%*, while features from the first and last layers differ significantly. This suggests the potential for skipping adjacent layers to reduce computation but also highlights the limitations of early-exit strategies (Del Corro et al. 2023), as shown in Fig.1 (B), where discarding deeper layers risks losing critical semantic information. In addition, recent breakthrough of the Shallow Brain Hypothesis (SBH) (Suzuki, Pennartz, and Aru 2023) in neuroscience also posits that biological cognition achieves efficient reasoning by balancing deep hierarchical processing with parallel cortico-subcortical loops acting as shortcuts, rather than relying exclusively on layer-by-layer signal transmission. Building upon these facts, we propose a dynamic layer activation paradigm for VLA models as visualized in Fig.1 (A-D). Our approach operationalizes SBH through two principles: ❶ **Hierarchical Semantic Encoding**: Task-critical features are distributed across layers, necessitating dynamic selection rather than fixed-depth processing; ❷ **Neuro-Inspired Sparsification**: Mirroring the brain's depth-parallelism trade-off, we activate layers only when their encoded semantics are task-relevant.

[*]These authors contributed equally.
[†]Corresponding authors.

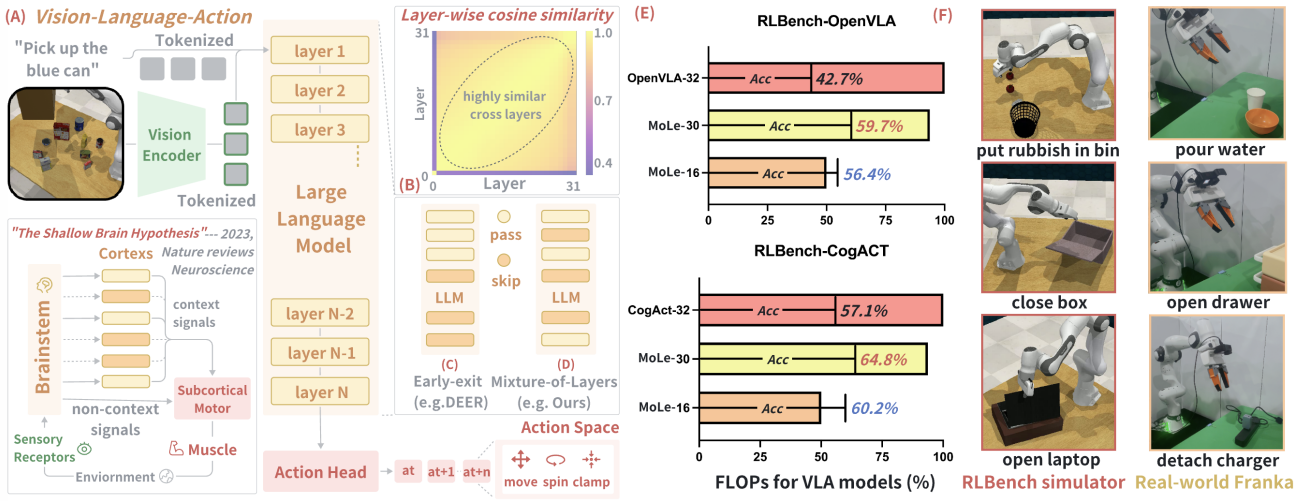

Figure 1: Overview of the proposed MoLe-VLA: (A) The VLA model and The Shallow Brain Hypothesis. (B) Cosine similarity across outputs of each LLM layer. (C) The early-exit strategy. (D) The Mixture-of-Layers mechanism. (E) Comparison of FLOPs and accuracy of MoLe with varying numbers of skipped layers against its backbones. (F) Experimental results.

In this paper, we introduce a **M**ixture-**o**f-**L**aye**r**s **V**ision-**L**anguage-**A**ction model (***MoLe-VLA***) incorporating a novel layer-selection router at the input stage of LLMs for its structural sparsity. Inspired by the routing mechanism in Mixture-of-Experts (MoE) (Lin et al. 2024; Zhang et al. 2024a,b), which enables horizontal expert-wise activation within a single LLM layer, we extend this concept vertically to achieve layer-wise activation. Specifically, we treat each LLM layer as an expert and utilize a biologically inspired router to manage layer skipping, mimicking the brain's selective activation of cortico-subcortical loops. Unlike Mixture-of-Depth (MoD) (Raposo et al. 2024), which assigns input tokens to different experts and risks token-wise inconsistencies due to varying perception levels across layers, our proposed MoLe dynamically selects the most relevant layers while processing input features holistically.

Moreover, traditional MoE or MoD routers that rely on simple linear layers lack the perception ability in 3D space and the capacity to discern the nuanced context essential for effective embodiment manipulation. Therefore, we propose a lightweight (with less than 0.1% additional parameters) yet effective *Spatial-Temporal Aware Router (STAR)* that explicitly preserves spatial-visual patterns and task-contextual dynamics through modality-specific processing, while enabling adaptive cross-modal fusion in a single step. This design synergistically leverages the spatial structure of visual inputs and task-phase awareness from language instructions (e.g., "first grasp then place") to optimize layer selection for VLA tasks. STAR integrates a end-to-end trained Transformer module to flexibly model temporal dependencies in task instructions. By combining these essential properties into a unified representation, STAR dynamically activates the most relevant layers by generating softmax probabilities for each layer and selecting the top-k layers with the highest probabilities. By fully leveraging spatial-temporal information, STAR ensures effective adaptation to the dynamic nature of embodied intelligence tasks, achieving optimal performance with reduced computational overhead.

Nonetheless, skipping certain layers inevitably reduces the cognitive expressiveness (Li et al. 2024a) of the model integrate visual and linguistic information, reason about their relationships, and make context-aware decisions or predictions to guide actions. To address this, we propose *Cognitive self-Knowledge Distillation (CogKD)*, where the full-layer backbone serves as the teacher and the MoLe layer-skipping model as the student. CogKD leverages the learnable *cognition token*, which is the last token in the LLM output sequence that aggregates global context similarly to the [CLS] token in BERT (Devlin et al. 2019) and naturally fuses visual and language information. By leveraging the similarity between the cognition token and other tokens, we dynamically identify task-relevant tokens of interest (ToIs) and concentrate the distillation process on them. This targeted approach eliminates the need for exhaustive one-to-one token-wise alignment, greatly reducing computational overhead while enhancing the transfer of essential semantic and cognitive features. As a result, the student model efficiently preserves critical capabilities even with aggressive layer skipping, supporting both effectiveness and scalability in complex robot manipulation.

The effectiveness of MoLe is demonstrated in real-world and RLBench simulations, as shown in Fig.1 (F), based on various VLA models, including OpenVLA, π_0, and CogAct. Extensive robotic experiments show that MoLe improves the mean success rate by 9.7% while accelerating inference by 36.8% over OpenVLA under the simulation scenario. The key contributions of MoLe are summarized as:

- Inspired by the SBH, we develop the MoLe-VLA, which mimics the signal flow of the human brain by enabling dynamic layer activation through a router tailored for VLA, as illustrated in Fig.1 (E).

- We propose STAR that leverages spatial-temporal cues to enable context-aware layer activation, improving efficiency and adaptability in complex tasks.
- We introduce CogKD that adaptively transfers key knowledge from the full teacher to the sparse student, recovering information lost during layer-skipping.

Related Works

Vision language action model. The remarkable success of VLMs (Karamcheti et al. 2024; Radford et al. 2021a) has driven the rapid development of VLA models (Kim et al. 2024; Li et al. 2024c), which extend VLMs by incorporating action generation. VLA models (Kim et al. 2024; Awadalla et al. 2023) first aim to bridge the gap between perception and action with the auto-regressive-based method, enabling machines to generate and execute actions based on sequence de-tokenization. (Black et al. 2024; Li et al. 2024a), on the other hand, leverage a diffusion-based action head to generate action sequences. Combining their strengths, HybridVLA (Liu et al. 2025) unifies both autoregressive and diffusion policies within a single LLM. However, the dense structure of LLMs hinders efficient deployment on commercial-grade GPUs in real-world applications, whereas MoLe-VLA acts as a plug-in module that achieves model sparsity without introducing significant parameter overhead.

Efficient multimodal large language models. With the advancement of VLA models, research on improving inference efficiency has focused on three key strategies: efficient architecture design, model compression, and dynamic networks. Liu et al. (Liu et al. 2024) leverage the Mamba model (Gu and Dao 2024) to enable efficient fine-tuning and inference, achieving pose prediction speeds 7× faster than existing robotic MLLMs. Wang et al. (Wang et al. 2022) utilizes a lightweight model with only 93M parameters while retaining 98.4% of its performance and delivering a 2.2× speedup. Yue et al. (Yue et al. 2024) propose a dynamic inference framework with multi-exit architectures, allowing early computation termination. However, existing early-exit methods often neglect the semantic importance of final layers. To address this, we propose a layer-skipping mechanism based on dynamic networks and knowledge distillation, reducing redundancy while enhancing performance.

Sparse mixture-of-experts. Activation sparsity has been extensively studied, with sparse MoE architectures proving highly efficient in LLMs by activating only a small portion of parameters during inference (Li et al. 2024b; Shazeer et al. 2017). In the LLMs and VLMs era, MoE has become a widely adopted and effective architecture (Dai et al. 2024a). For example, (Lin et al. 2024) achieves performance comparable to LLaVA-1.5-7B on various visual understanding benchmarks and even surpasses LLaVA-1.5-13B on the object hallucination benchmark, using only 3B sparsely activated parameters. Additionally, (Raposo et al. 2024) employs a router to dynamically choose between computational paths, such as a standard block's computation or a residual connection. While our model shares similarities with MoD, we differ by employing a router to select all standard blocks.

Methods

Preliminary: Mixture-of-Experts

The MoE paradigm enhances model capacity while maintaining computational efficiency via conditional computation. For an input $\boldsymbol{x} \in \mathbb{R}^d$, a standard MoE is defined as:

$$\text{MoE}(\boldsymbol{x}) = \sum_{i=1}^{N_e} G_i(\boldsymbol{x}) \cdot E(\boldsymbol{x}), \tag{1}$$

where N_e is the number of experts, $E : \mathbb{R}^d \to \mathbb{R}^d$ represents the i-th expert network, and $G(\boldsymbol{x}) = \{G_1(\boldsymbol{x}), \ldots, G_{N_e}(\boldsymbol{x})\}$ is the gating function satisfying $\sum_{i=1}^{N_e} G_i(\boldsymbol{x}) = 1$. The gating weights are computed as:

$$G(\boldsymbol{x}) = \text{Softmax}(\mathbf{W}_g \cdot \boldsymbol{x} + \mathbf{b}_g), \tag{2}$$

where $\mathbf{W}_g \in \mathbb{R}^{N_e \times d}$ and $\mathbf{b}_g \in \mathbb{R}^{N_e}$ are learnable parameters. To improve efficiency, sparse gating with top-k selection is applied. To address load imbalance, where inputs are routed to few experts, a load balance loss $\mathcal{L}_{\text{lb}}$ is introduced:

$$\mathcal{L}_{\text{lb}} = \frac{1}{N_e} \sum_{i=1}^{N_e} \left(\frac{\sum_{n=1}^{N} v_i(\boldsymbol{x}_n)}{\sum_{n=1}^{N} v_i(\boldsymbol{x}_n) + \epsilon} \right)^2, \tag{3}$$

where $v_i(\boldsymbol{x}_n) = 1$ if the i-th expert is selected for input $\boldsymbol{x}_n$ by the top-k gating, and $v_i(\boldsymbol{x}_n) = 0$ otherwise. This loss encourages balanced experts and improves efficiency.

Mixture-of-Layers: MoLe-VLA

Vision language action model. Tasked with a language instruction $\boldsymbol{l}$ with a length L, a robot receives an observation o_t from sensors (*e.g.*, RGB image from the camera) at timestep t to predict the action space of a gripper with 7 degrees of freedom (DoF) to execute:

$$\mathbf{a}_t^* = [\Delta x, \Delta y, \Delta z, \Delta\phi, \Delta\theta, \Delta\psi, g], \tag{4}$$

where $\Delta x, \Delta y$, and Δz are the relative translation offsets of the end effector, $\Delta\phi, \Delta\theta$, $\Delta\psi$ denote the rotation changes, and $g \in \{0, 1\}$ indicates the gripper open/close state.

Our basic VLA model mainly consists of a vision encoder $\mathcal{E}$, an MLLM π, and an action module $\mathcal{A}$. The vision encoder $\mathcal{E}$ comprises DINO-v2 (Oquab et al. 2023) and Siglip (Zhai et al. 2023), which encodes an input image o_t into a sequence of informative tokens v_t. For multimodal fusion, an MLLM is established on top of the visual representations generated by the vision encoder $\mathcal{E}$, which functions as an effective multimodal feature extractor π, formalized as:

$$\boldsymbol{f}_t = \pi(\boldsymbol{l}, \mathcal{E}(o_t)), \tag{5}$$

where the output $\boldsymbol{f}_t$ represents the hidden state sequence from the last layer of our MLLM at timestep t, corresponding to the cognition token. This serves as a condition for the subsequent action module to interpret and derive the desired actions. Following CogAct (Li et al. 2024a), our action module $\mathcal{A}$ takes the cognition feature $\boldsymbol{e}_t^c$ extracted from the output feature $\boldsymbol{f}_t$ as input and predicts the final actions $\mathbf{a}_t^*$.

Our vision, language, and action modules are trained end-to-end by minimizing the mean squared error between the

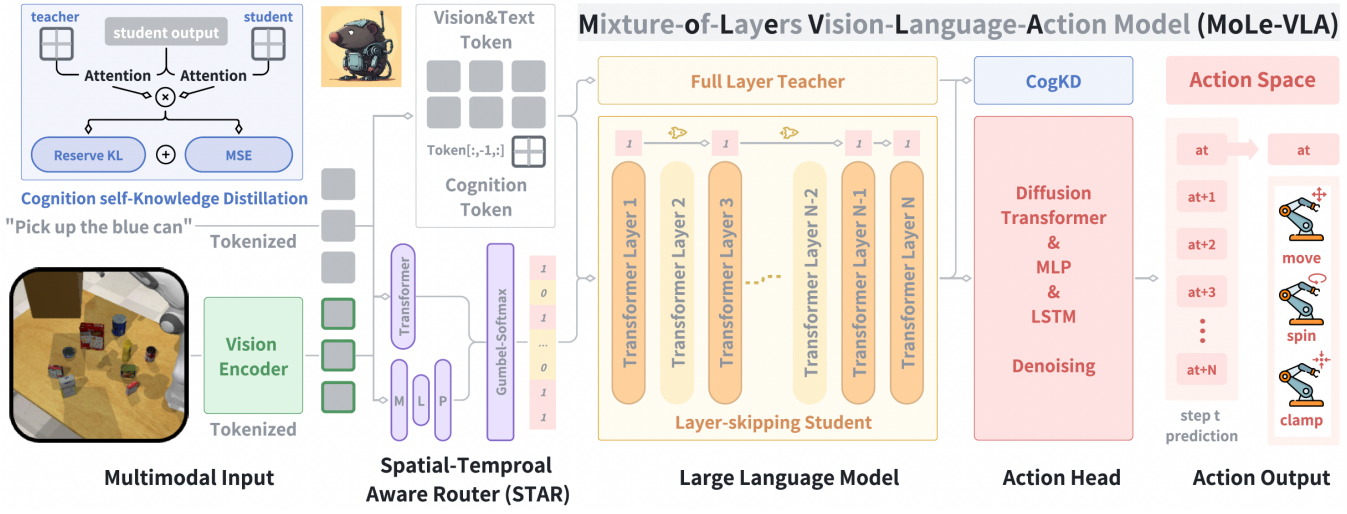

Figure 2: **The overall framework of MoLe-VLA.** Our proposed *Mixture of Layers (MoLe)* architecture consists of a *Spatial-Temporal Aware Router (STAR)* and a devised *Cognitive self-Knowledge Distillation (CogKD)* for VLA models.

predicted noises from the action module and the ground truth noises. Taking the diffusion head as an example, the loss function is defined as:

$$\mathcal{L}_{task} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1),i} ||\hat{\epsilon}^i - \epsilon|| \tag{6}$$

where $\hat{\epsilon}^i$ is the predicted noise for the noisy action a_t^* at the i's denoising step, and ϵ is the corresponding ground truth.

Layer-skipping via MoLe router. We propose MoLe-VLA to improve the efficiency of LLM in robotic tasks, where many transformer layers are underutilized due to the simpler reasoning demands of robotics tasks. MoLe employs a lightweight router to adaptively skip non-essential transformer layers during inference, reducing computational costs while maintaining performance. As shown in Fig.2, for a given MLLM π with K layers, the MoLe router processes the input embeddings $\boldsymbol{x}_k \in \mathbb{R}^{b \times n \times d}$ and generates a binary gating vector $G_{mol}(\boldsymbol{x}) = \{G_k\}_{k=1}^K$, where $G_k \in [0, 1]$. To ensure efficiency, only the top-k values in $G_{mol}(\boldsymbol{x})$ are set to 1, determining which layers π_k are executed with the hidden feature $\boldsymbol{h}_k$ while the rest are skipped:

$$\boldsymbol{h}_k = G_k \cdot \pi_k(\boldsymbol{h}_{k-1}) + (1 - G_k) \cdot \boldsymbol{h}_{k-1}. \tag{7}$$

Unlike traditional MoE routers that allocate tokens to experts, the MoLe router skips entire layers, avoiding redundant computations. This improves inference efficiency and responsiveness, making MoLe particularly suited for real-time robotic tasks like manipulation and navigation that require lightweight and adaptive processing.

Spatial-Temporal Aware Router

We propose the STAR routing mechanism, which preserves spatial-visual patterns and task-aware dynamics through modality-specific processing and adaptive cross-modal fusion, optimizing layer selection for VLA tasks. Given visual features $\boldsymbol{v}_t \in \mathbb{R}^{b \times n_{img} \times d}$ and textual features $\boldsymbol{l} \in \mathbb{R}^{b \times n_{text} \times d}$, both modalities are projected into a shared latent space using a learnable matrix $\mathbf{W}_p \in \mathbb{R}^{d \times d_1}$:

$$\boldsymbol{h}_{img} = \boldsymbol{v}_t \cdot \mathbf{W}_p, \quad \boldsymbol{h}_{text} = \boldsymbol{l} \cdot \mathbf{W}_p. \tag{8}$$

We compute spatial routing weights $\mathbf{S} \in \mathbb{R}^{b \times N_e}$ from $\boldsymbol{h}_{img}$ to process image patches to capture geometric relationships and object layouts:

$$\mathbf{S} = \mathbf{W}_s^{(2)} \cdot \varphi(\mathbf{W}_s^{(1)} \cdot \boldsymbol{h}_{img} + \mathbf{b}_s^{(1)}), \tag{9}$$

where φ indicates GELU, while spatial pooling preserves translation invariance for visual reasoning. Concurrently, temporal routing weights $\mathbf{T} \in \mathbb{R}^{b \times N_e}$ are derived from $\boldsymbol{h}_{text}$ using a Transformer module to extract instruction-aware temporal dynamics, where "temporal" corresponds to task execution phases implied by language instructions (e.g., "first grasp then place"):

$$\mathbf{T} = \mathbf{W}_t \cdot \Phi(\text{Transformer}(\boldsymbol{h}_{text})). \tag{10}$$

The Transformer models step dependency in textual commands, while average pooling Φ distills phase-shift signals. A learnable sharpness controller $\alpha \in [0, 1]$ adaptively adjusts routing focus based on instruction complexity:

$$\alpha = \sigma(\mathbf{W}_\tau^\top \cdot \boldsymbol{h}_{text}^{[CLS]} + b_\tau), \tag{11}$$

where σ is the sigmoid function and higher α emphasizes spatial experts for precise grasp pose estimation, while lower α activates geometry-text alignment experts for task-conditioned adaptation. Last but not least, the final gating weights $\mathbf{G} \in \mathbb{R}^{b \times N_e}$ combine both modalities $\mathbf{S}$ and $\mathbf{T}$ through a dynamic and sharpened blending:

$$\mathbf{G} = \mathbf{S} + \mathbf{T}. \tag{12}$$

By integrating spatial and temporal information, our method enables the router to select LLM layers, optimizing performance for VLA tasks adaptively. The approach is efficient, requiring only $\mathcal{O}(N_e(d_2 + N_{text}^2))$ FLOPs per sample compared to $\mathcal{O}(N_e d)$ in standard MoE frameworks, where $d \gg N_{text}, d_2$. This design ensures high adaptability and computational efficiency.

Cognitive self-Knowledge Distillation

To mitigate cognition loss from layer-skipping, we employ self-distillation, using the original model as teacher and MoLe as student. Following token-wise distillation approaches (Yang et al. 2024; Zhang et al. 2024c, 2023), the student reconstructs each token from the teacher: given $\boldsymbol{f}^{(t)} \in \mathbb{R}^{n\times d}$ and $\boldsymbol{f}^{(s)} \in \mathbb{R}^{n\times d_s}$, token reconstruction is performed as:

$$\mathcal{L}_{\text{mimic}} = \frac{1}{N}\left\|\boldsymbol{f}^{(t)} - \mu(\boldsymbol{f}^{(s)})\right\|_2^2, \tag{13}$$

where $\mu(\cdot)$ is the projection layer. However, Eq.13 distills all tokens equally, which may be suboptimal for complex manipulation tasks where certain visual tokens are more relevant to the text.Therefore, we leverage the *cognition token* $\boldsymbol{e}_t^c \in \mathbb{R}^{1\times d}$, which is the **last token** in the output sequence of the LLM. that naturally encapsulates the global task semantics formed by the fusion of language instructions and visual information under multimodal inputs, to distill adaptively. Each model has its own cognition token ($\boldsymbol{e}t^{c,(t)}$ for the teacher and $\boldsymbol{e}_t^{c,(s)}$ for the student). We identify tokens of interest (ToIs) $\boldsymbol{M}$ by computing their similarity to the cognition token during distillation:

$$\boldsymbol{M}^{(i)} = \eta(\boldsymbol{e}^{c,(i)}\boldsymbol{f}^{(s)}), i \in \{s, t\}, \tag{14}$$

where η denotes the Sigmoid function. Next, we utilize the *intersection* of ToIs generated by the teacher and student cognition tokens to decide the distillation degree of each token, where $\boldsymbol{M} = \boldsymbol{M}^{(t)} \odot \boldsymbol{M}^{(s)}$, because the distillation tokens should consist of the ones both important to the teacher and student. Therefore, the Eq.13 can be updated as:

$$\mathcal{L}_{\text{cog-mimic}} = \frac{1}{N}\left\|\boldsymbol{M}\odot\boldsymbol{f}^{(t)} - \mu(\boldsymbol{M}\odot\boldsymbol{f}^{(s)})\right\|_2^2, \tag{15}$$

where $\odot$ denotes element-wise matrix multiplication. Furthermore, we introduce the *Reverse-KL* (Gu et al. 2023) paired with our cognition token, with prioritizing mode coverage of the teacher distribution while suppressing spurious student modes. The reason is that for VLA distillation, reverse-KL loss prevents overconfident predictions on out-of-distribution visual-language pairs while preserving semantic diversity, offering better generalization than forward-KL. Therefore, we obtain $\mathcal{L}_{\text{cog-reversekl}}$ to enhance the distribution constraint following the manner before to:

$$\mathcal{L}_{\text{cog-reversekl}} = (\boldsymbol{M}\odot\boldsymbol{f}^{(s)})\log\left(\frac{\boldsymbol{M}\odot\boldsymbol{f}^{(s)}}{\boldsymbol{M}\odot\boldsymbol{f}^{(t)}}\right). \tag{16}$$

Finally, our eventual CogKD loss can be formulated as:

$$\mathcal{L}_{\text{cog}} = (1-\lambda_1)\mathcal{L}_{\text{cog-mimic}} + \lambda_1\mathcal{L}_{\text{cog-reversekl}}, \tag{17}$$

where λ_1 is set to 0.5 for balancing the losses.

Optimization Objective

For the update of the teacher model, we initialize both models with pre-trained parameters and use the exponential moving average (EMA) to update the teacher model with the update weight $\alpha = 0.999$. Our final training objective can be formulated with the combination of $\mathcal{L}_{task}$, $\mathcal{L}_{cog}$ and $\mathcal{L}_{lb}$:

$$\mathcal{L}_{MoLe} = \mathcal{L}_{task} + \lambda_2\mathcal{L}_{cog} + \lambda_3\mathcal{L}_{lb}, \tag{18}$$

where λ_2 and λ_3 are two hyperparameters which are set to 0.5 and 0.1 by default.

Experiments

Implementation Details

Simulation and real-world deployment. *1) RLBench* includes 10 diverse tabletop tasks performed with a Franka Panda robot and a front-view camera. These tasks range from object manipulation to environment interaction, such as: *Close box*, *Close laptop lid*, *Toilet seat down*, *Put rubbish in bin*, *Sweep to dustpan*, *Close fridge*, *Phone on base*, *Take umbrella out of stand*, *Frame off hanger*, and *Change clock*. Task data are generated using predefined waypoints and the Open Motion Planning Library (Sucan, Moll, and Kavraki 2012). Following prior work (Jia et al. 2024), each task includes 100 training trajectories sampled using a frame-based approach and evaluated in 25 trials per task.

2) Real-world deployment is evaluated on the Franka Research 3 robot equipped with a 3D-printed UMI gripper (Chi et al. 2024). A GoPro 9 camera mounted on the wrist captures real-world visual observations. We collect 50 demonstrations across three tasks, including *detach charger*, *pull drawer*, and *pour water*. A single agent is trained across all tasks and evaluated in 10 trials per task within the training workspace. The success rate is determined through human assessment and serves as the evaluation metric.

Baselines. We compare *MoLe* with three state-of-the-art VLA methods across two action generation paradigms: 1) *Autoregressive models*, including OpenVLA (Kim et al. 2024), which uses LLaMA for discrete action prediction, and 2) *Diffusion-based models*, such as *CogAct* (Li et al. 2024a) and π_0 (Black et al. 2024), which predicts action chunks via a diffusion head. Additionally, we evaluate several VLA efficiency baselines all based on CogAct backbone: *DeeR* (Yue et al. 2024), which enables early exits in LLMs; *MoD* (Raposo et al. 2024), which allocates input tokens dynamically across layers; and *Random-skip*, which skips LLM layers randomly. RoboMamba (Liu et al. 2024) with Mamba-2.8B LLM backbone is also selected to demonstrate the efficiency of MoLe. For a fair comparison, *DeeR* using single-phase training and full model loading. We integrate MoLe using a default **50% layer-skip**.

Training and evaluation details. All baselines are trained using the same task configuration for fair comparison. It should be noted that we conduct **mixed-task, one-time training** and evaluate each test task separately. Each method's official pre-trained parameters are loaded, following their respective training settings. For MoLe-VLA, the single-view RGB input is resized to 224×224, and the robot state is aligned with the predicted actions (7-DOF end-effector poses). The model is trained with a batch size of 64 and 8 diffusion steps per sample, using pre-trained weights for the vision and language modules. MoLe-VLA is **fine-tuned** on LLM backbones that have been pre-trained with large-scale robotic data, using end-to-end training with a constant learning rate of 2×10^{-5} for 1,000 iterations. Training is conducted on 8 NVIDIA A800 GPUs in approximately 1.5 hours using PyTorch's Fully Sharded Data Parallel (FSDP) framework, while the frequency is evaluated on GeForce RTX 4090D.

Methods	Action Head	Backbone	Put Rubbish in Bin	Close Box	Close Laptop Lid	Take Umbrella out of Stand	Close Fridge
RLBench							
OpenVLA (CoRL'24)	Auto-regressive	LLaMA2-7B	13.0%(±0.03)	78.0%(±0.04)	78.0%(±0.07)	*31.0%*(±0.03)	78.0%(±0.10)
MoLe-OpenVLA (Ours)	Auto-regressive	LLaMA2-7B	22.0%(±0.10)	*79.0%*(±0.09)	*76.0%*(±0.07)	36.0%(±0.08)	**89.0%**(±0.09)
π_0 (Arxiv'24)	Diffusion	Paligemma-2.6B	33.0%(±0.05)	76.0%(±0.06)	66.0%(±0.10)	27.0%(±0.01)	50.0%(±0.08)
MoLe-π_0 (Ours)	Diffusion	Paligemma-2.6B	56.0%(±0.02)	76.0%(±0.03)	66.0%(±0.03)	48.0%(±0.04)	50.0%(±0.16)
CogAct (Arxiv'24)	Diffusion	LLaMA2-7B	**58.0%**(±0.05)	80.0%(±0.06)	*34.0%*(±0.03)	44.0%(±0.07)	78.0%(±0.03)
Random-skip-CogAct	Diffusion	LLaMA2-7B	16.0%(±0.04)	*80.0%*(±0.07)	**80.0%**(±0.07)	32.0%(±0.05)	84.0%(±0.04)
MoD-CogAct (Arxiv'24)	Diffusion	LLaMA2-7B	*53.0%*(±0.08)	81.0%(±0.02)	73.0%(±0.03)	**43.0%**(±0.03)	*90.0%*(±0.08)
DeeR-CogAct (NeruIPS'24)	Diffusion	LLaMA2-7B	35.0%(±0.11)	74.0%(±0.15)	74.0%(±0.10)	*49.0%*(±0.15)	79.0%(±0.18)
MoLe-CogAct (Ours)	Diffusion	LLaMA2-7B	52.0%(±0.15)	**72.0%**(±0.10)	68.0%(±0.09)	*49.0%*(±0.08)	94.0%(±0.04)

Methods	Sweep to Dustpan	Phone on Base	Change Clock	Toilet Seat Down	Take Frame off Hanger	Mean Acc.% ↑	Speed ↑
OpenVLA (CoRL'24)	*61.0%*(±0.05)	21.0%(±0.08)	*16.0%*(±0.05)	80.0%(±0.03)	11.0%(±0.03)	46.7% (-)	6.8 Hz (-)
MoLe-OpenVLA (Ours)	**83.0%**(±0.08)	16.0%(±0.05)	18.0%(±0.12)	**100.0%**(±0.00)	45.0%(±0.03)	56.4% (9.7% ↑)	9.3 Hz (2.5 Hz ↑)
π_0 (Arxiv'24)	60.0%(±0.05)	44.0%(±0.09)	16.0%(±0.05)	88.0%(±0.05)	40.0%(±0.07)	50.0% (-)	13.8 Hz (-)
MoLe-π_0 (Ours)	60.0%(±0.02)	32.0%(±0.13)	32.0%(±0.05)	88.0%(±0.04)	44.0%(±0.04)	55.4% (5.4% ↑)	18.4 Hz (4.8 Hz ↑)
CogAct (Arxiv'24)	47.0%(±0.04)	*55.0%*(±0.03)	17.0%(±0.03)	95.0%(±0.07)	63.0%(±0.02)	57.1% (-)	9.8 Hz (-)
Random-skip-CogAct	65.0%(±0.08)	24.0%(±0.09)	12.0%(±0.07)	91.0%(±0.09)	28.0%(±0.02)	51.2% (5.9% ↓)	13.4 Hz (3.6 Hz ↑)
MoD-CogAct (Arxiv'24)	18.0%(±0.15)	34.0%(±0.12)	21.0%(±0.07)	*99.0%*(±0.02)	**53.0%**(±0.11)	56.5% (0.6% ↓)	13.3 Hz (3.5 Hz ↑)
DeeR-CogAct (NeruIPS'24)	35.0%(±0.29)	**52.0%**(±0.16)	28.0%(±0.06)	*94.0%*(±0.12)	*68.0%*(±0.08)	*58.8%* (1.7% ↑)	12.5 Hz (2.7 Hz ↑)
MoLe-CogAct (Ours)	*44.0%*(±0.15)	42.0%(±0.09)	**32.0%**(±0.13)	**100.0%**(±0.00)	48.0%(±0.17)	**60.2%** (3.1% ↑)	13.3 Hz (3.5 Hz ↑)

Table 1: **Performance comparison** with existing VLA models across ten tasks in RLBench settings.

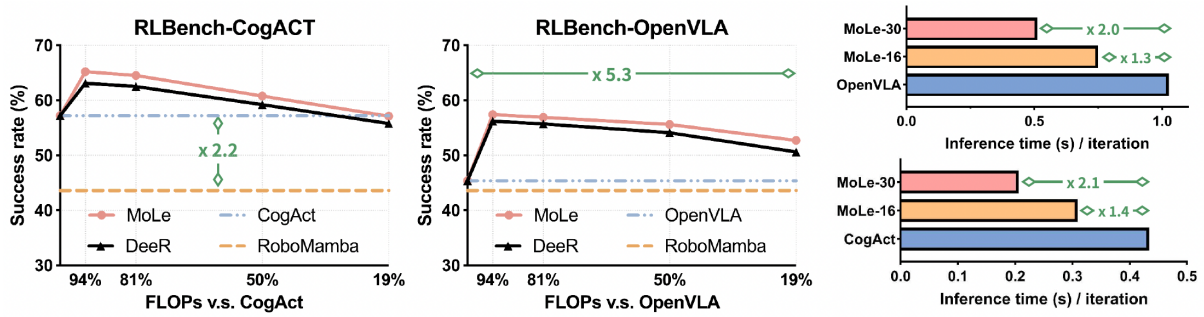

Figure 3: **Efficiency analysis with FLOPs and inference time.** (Left) Success rate *v.s.* the FLOPs reduction compared to model backbone with differnt number of layer-skipping. (Right) Inference time per iteration for different layers.

Quantitative Results

Performance enhancement The performance of MoLe-VLA against SOTA VLA models across ten RLBench tasks is shown in Tab.**??** with four repetitions. Specifically, MoLe-OpenVLA improves success rate by *9.7%* with a 2.5 Hz acceleration. As for π_0, MoLe also improves its success by 5.4% while further accelerates its inference to a optimal 18.4 Hz. MoLe-CogAct achieves the highest mean success rate of *60.2%* among all the other efficient baselines, surpassing CogAct by *3.1%*. Compared with efficiency baselines, MoLe-VLA not only consistently outperforms other methods but also achieves the lowest variance (*±0.01*), while MoD and DeeR exhibit unstable performance with variances of *±0.03* and *±0.05* on CogAct due to overlooking semantic layers, leading to token-wise perception inconsistencies. Although random-skip is faster than MoLe, the significant performance gap (9.0%) demonstrates the effectiveness of MoLe-VLA in both model performance and efficiency.

Efficiency analysis We analyze success rate changes with increasing skipped layers in Fig.3. MoLe achieves similar success rates compared to the full-layer backbone while only *19%* of the FLOPs. Notably, MoLe-OpenVLA significantly outperforms the original OpenVLA by a large margin. Furthermore, detailed statistics on model efficiency are provided in Tab.2. While DeeR offers more dynamic layer-skipping, it incurs extra optimization overhead and auxiliary action computation. In contrast, MoLe achieves superior efficiency, requiring just *0.309s*.

MoLe with quantization analysis We highlight the efficiency of MoLe under 8-bit quantization as shown in Tab.2. MoLe achieves a higher success rate of 58.8% with an inference frequency of 15.7 Hz, while utilizing only 55% of the GPU memory compared to CogAct, which achieves just 9.8 Hz. This demonstrates MoLe's ability to maintain superior performance with significantly lower computational costs after quantization for computational-constrained real-world applications with commercial-grade GPUs.

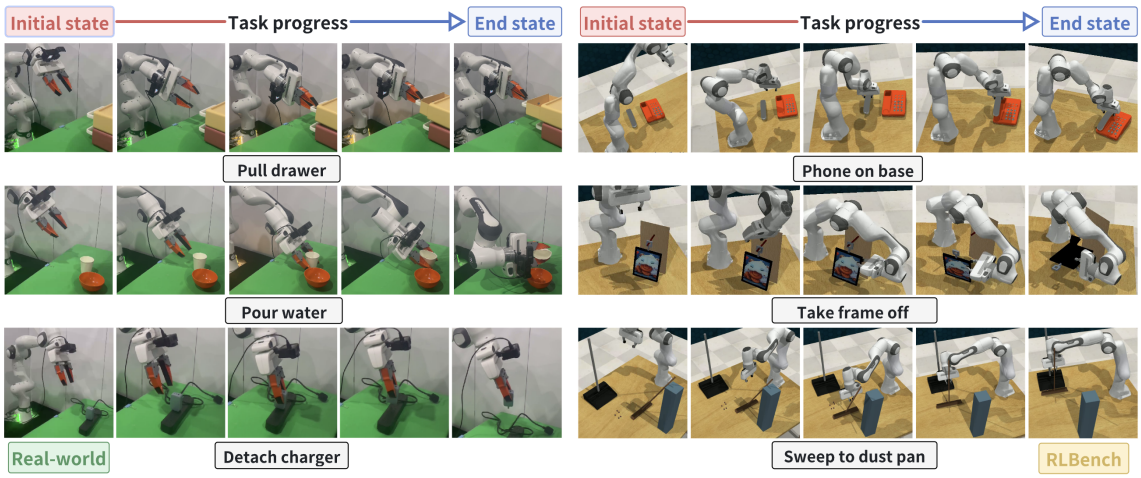

Figure 4: **The qualitative results of MoLe-VLA**, including the manipulation progress and the task completion end state.

Methods	Time ↓	FLOPs ↓	Mean ↑
CogAct	0.434 s	1935.8 G	57.2%
DeeR	0.337 s	997.4 G	59.2%
MoLe	0.309 s	985.8 G	60.8%
Precision	**Speed ↑**	**Memory ↓**	**Mean ↑**
CogAct-FP16	9.8 Hz	16055 MB	57.2%
MoLe-INT8	15.7 Hz	8887 MB	58.8%

Table 2: **Inference analysis** on RLBench with quantization.

Methods	Mixture-of-Layers		CogKD Loss			Mean↑
RLBench	Linear	STAR	MSE	KL	R-KL	
Ex_0	✗	✗	✗	✗	✗	46.7%
Ex_{1-1}	✓	✗	✗	✗	✗	42.6%
Ex_{1-2}	✗	✓	✗	✗	✗	45.6%
Ex_{2-1}	✗	✓	✓	✗	✗	51.4%
Ex_{2-2}	✗	✓	✗	✓	✗	49.6%
Ex_{2-3}	✗	✓	✗	✗	✓	53.8%
Ex_{2-4}	✗	✓	✓	✗	✓	**56.4%**

Table 3: **Ablation study** on RLBench based on OpenVLA.

Methods	Detach charger	Pull drawer	Pour water	Mean↑	Speed↑
CogAct	60.0%	60.0%	80.0%	66.7%	9.8 Hz
MoLe-Cog	70.0%	60.0%	80.0%	70.0%	13.3 Hz
π_0	50.0%	50.0%	60.0%	53.3%	13.8 Hz
MoLe-π_0	60.0%	50.0%	70.0%	60.0%	18.4 Hz

Table 4: **Success rate for real-world** evaluated with Franka.

Ablation study Tab.3 demonstrates the effectiveness of our *STAR* and *CogKD* in RLBench based on OpenVLA. Adapting MoLe with the traditional linear router Ex_{1-1} reduces accuracy from 46.7% to 42.6%, while our proposed STAR Ex_{1-2} recovers 3.0% with only the same size of 13.17 MB as the linear router. Further improvements are achieved with tailored CogKD loss variants, particularly when combining MSE and Reverse KL losses Ex_{2-4}, which reaches a 56.4% mean success rate. These results highlight our methods' superiority and verify that CogKD effectively transfers context knowledge in auto-regressive VLA.

Quantitative results in real-world tasks. We conducted experiments involving interactions with various real-world objects, as summarized in Tab.4. The results show that MoLe consistently delivers strong performance across three tasks compared to CogAct and π_0. Notably, in the challenging *pour water* task, which demands precise 3D position and rotation predictions, MoLe shows superiority and matches CogAct with the success rate of *80%*. In addition, MoLe significantly outperforms π_0 with only *53.33%* mean success. These results highlight that MoLe preserves the ability to understand 3D spatial scenes and make accurate predictions with a *50%* computation reduction in LLM.

Qualitative Results

As shown in Fig.4, we visualize the manipulation process for three real-world and RLBench simulation tasks. Our method accurately predicts continuous 7-DoF end-effector poses, enabling precise task execution along planned trajectories. For instance, in the *pour water* task, MoLe-VLA successfully grasps the cup, lifts the can, positions it above the bowl, and smoothly rotates the gripper to control water flow.

Conclusions

We proposed ***MoLe-VLA***, a method inspired by the Shallow Brain Hypothesis, to optimize VLA models for robotics. MoLe dynamically activates key LLM layers with a specially devised STAR router, reducing redundancy while preserving essential information. To address performance loss from layer skipping, we developed CogKD to enhance efficiency and cognitive capacity. Experiments on real-world and RLBench environments show that MoLe reduces computational costs, enabling efficient and adaptable robot.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62476011) and (625B2090), Beijing Natural Science Foundation (L252060), and also in part by RGC GRF 15200321, 15201322, 15230624, 15239925, ITC ITF-ITS/056/22MX, ITS/052/23MX, and PolyU 1-CDKK, G-SAC8, K-ZYAP..

References

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.

Awadalla, A.; Gao, I.; Gardner, J.; Hessel, J.; Hanafy, Y.; Zhu, W.; Marathe, K.; Bitton, Y.; Gadre, S.; Sagawa, S.; Jitsev, J.; Kornblith, S.; Koh, P. W.; Ilharco, G.; Wortsman, M.; and Schmidt, L. 2023. OpenFlamingo: An Open-Source Framework for Training Large Autoregressive Vision-Language Models. arXiv:2308.01390.

Black, K.; Brown, N.; Driess, D.; Esmail, A.; Equi, M.; Finn, C.; Fusai, N.; Groom, L.; Hausman, K.; Ichter, B.; et al. 2024. Pi-0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*.

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

Chi, C.; Xu, Z.; Pan, C.; Cousineau, E.; Burchfiel, B.; Feng, S.; Tedrake, R.; and Song, S. 2024. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*.

Dai, D.; Deng, C.; Zhao, C.; Xu, R. X.; Gao, H.; Chen, D.; Li, J.; Zeng, W.; Yu, X.; Wu, Y.; Xie, Z.; Li, Y. K.; Huang, P.; Luo, F.; Ruan, C.; Sui, Z.; and Liang, W. 2024a. DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models. arXiv:2401.06066.

Dai, X.; Li, J.; Liu, X.; Yu, A.; and Lui, J. C. 2024b. Cost-Effective Online Multi-LLM Selection with Versatile Reward Models. *arXiv preprint arXiv:2405.16587*.

Del Corro, L.; Del Giorno, A.; Agarwal, S.; Yu, B.; Awadallah, A.; and Mukherjee, S. 2023. Skipdecode: Autoregressive skip decoding with batching and caching for efficient llm inference. *arXiv preprint arXiv:2307.02628*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 4171–4186.

Gu, A.; and Dao, T. 2024. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv:2312.00752.

Gu, Y.; Dong, L.; Wei, F.; and Huang, M. 2023. MiniLLM: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

James, S.; Ma, Z.; Arrojo, D. R.; and Davison, A. J. 2020. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2): 3019–3026.

Jia, Y.; Liu, J.; Chen, S.; Gu, C.; Wang, Z.; Luo, L.; Lee, L.; Wang, P.; Wang, Z.; Zhang, R.; et al. 2024. Lift3d foundation policy: Lifting 2d large-scale pretrained models for robust 3d robotic manipulation. *arXiv preprint arXiv:2411.18623*.

Karamcheti, S.; Nair, S.; Balakrishna, A.; Liang, P.; Kollar, T.; and Sadigh, D. 2024. Prismatic VLMs: Investigating the Design Space of Visually-Conditioned Language Models. arXiv:2402.07865.

Kim, M. J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.; Lam, G.; Sanketi, P.; Vuong, Q.; Kollar, T.; Burchfiel, B.; Tedrake, R.; Sadigh, D.; Levine, S.; Liang, P.; and Finn, C. 2024. OpenVLA: An Open-Source Vision-Language-Action Model. arXiv:2406.09246.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, 12888–12900. PMLR.

Li, Q.; Liang, Y.; Wang, Z.; Luo, L.; Chen, X.; Liao, M.; Wei, F.; Deng, Y.; Xu, S.; Zhang, Y.; Wang, X.; Liu, B.; Fu, J.; Bao, J.; Chen, D.; Shi, Y.; Yang, J.; and Guo, B. 2024a. CogACT: A Foundational Vision-Language-Action Model for Synergizing Cognition and Action in Robotic Manipulation. arXiv:2411.19650.

Li, T.; Wen, Z.; Li, Y.; and Lee, T. S. 2024b. Emergence of Shape Bias in Convolutional Neural Networks through Activation Sparsity. *Advances in Neural Information Processing Systems*, 36.

Li, X.; Liu, M.; Zhang, H.; Yu, C.; Xu, J.; Wu, H.; Cheang, C.; Jing, Y.; Zhang, W.; Liu, H.; Li, H.; and Kong, T. 2024c. Vision-Language Foundation Models as Effective Robot Imitators. arXiv:2311.01378.

Li, X.; Liu, M.; Zhang, H.; Yu, C.; Xu, J.; Wu, H.; Cheang, C.; Jing, Y.; Zhang, W.; Liu, H.; et al. 2023. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*.

Lin, B.; Tang, Z.; Ye, Y.; Huang, J.; Zhang, J.; Pang, Y.; Jin, P.; Ning, M.; Luo, J.; and Yuan, L. 2024. MoE-LLaVA: Mixture of Experts for Large Vision-Language Models. arXiv:2401.15947.

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

Liu, J.; Liu, M.; Wang, Z.; An, P.; Li, X.; Zhou, K.; Yang, S.; Zhang, R.; Guo, Y.; and Zhang, S. 2024. RoboMamba: Efficient Vision-Language-Action Model for Robotic Reasoning and Manipulation. arXiv:2406.04339.

Mees, O.; Hermann, L.; Rosete-Beas, E.; and Burgard, W. 2022. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3): 7327–7334.

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

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021a. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021b. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.

Raposo, D.; Ritter, S.; Richards, B.; Lillicrap, T.; Humphreys, P. C.; and Santoro, A. 2024. Mixture-of-Depths: Dynamically allocating compute in transformer-based language models. arXiv:2404.02258.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; and Dean, J. 2017. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. arXiv:1701.06538.

Sucan, I. A.; Moll, M.; and Kavraki, L. E. 2012. The open motion planning library. *IEEE Robotics & Automation Magazine*, 19(4): 72–82.

Suzuki, M.; Pennartz, C. M.; and Aru, J. 2023. How deep is the brain? The shallow brain hypothesis. *Nature Reviews Neuroscience*, 24(12): 778–791.

Wang, T.; Zhou, W.; Zeng, Y.; and Zhang, X. 2022. EfficientVLM: Fast and Accurate Vision-Language Models via Knowledge Distillation and Modal-adaptive Pruning. arXiv:2210.07795.

Yang, Z.; Li, Z.; Zeng, A.; Li, Z.; Yuan, C.; and Li, Y. 2024. ViTKD: Feature-based knowledge distillation for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1379–1388.

Yue, Y.; Wang, Y.; Kang, B.; Han, Y.; Wang, S.; Song, S.; Feng, J.; and Huang, G. 2024. DeeR-VLA: Dynamic Inference of Multimodal Large Language Models for Efficient Robot Execution. arXiv:2411.02359.

Zhai, X.; Mustafa, B.; Kolesnikov, A.; and Beyer, L. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, 11975–11986.

Zhang, R.; Cheng, A.; Luo, Y.; Dai, G.; Yang, H.; Liu, J.; Xu, R.; Du, L.; Du, Y.; Jiang, Y.; et al. 2024a. Decomposing the neurons: Activation sparsity via mixture of experts for continual test time adaptation. *arXiv preprint arXiv:2405.16486*.

Zhang, R.; Luo, Y.; Liu, J.; Yang, H.; Dong, Z.; Gudovskiy, D.; Okuno, T.; Nakata, Y.; Keutzer, K.; Du, Y.; et al. 2024b. Efficient deweahter mixture-of-experts with uncertainty-aware feature-wise linear modulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16812–16820.

Zhang, Y.; Chen, W.; Lu, Y.; Huang, T.; Sun, X.; and Cao, J. 2023. Avatar knowledge distillation: self-ensemble teacher paradigm with uncertainty. In *Proceedings of the 31st ACM international conference on multimedia*, 5272–5280.

Zhang, Y.; Huang, T.; Liu, J.; Jiang, T.; Cheng, K.; and Zhang, S. 2024c. FreeKD: Knowledge distillation via semantic frequency prompt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15931–15940.