VLA-GSE: BOOSTING PARAMETER-EFFICIENT FINE-TUNING IN VLA WITH GENERALIZED AND SPECIALIZED EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action (VLA) models inherit rich visual-semantic priors from pre-trained vision-language backbones, but adapting them to robotic control remains challenging. Full fine-tuning (FFT) is prone to overfitting on downstream robotic data and catastrophic forgetting of pretrained vision-language capabilities. Parameter-efficient fine-tuning (PEFT) better preserves pre-trained knowledge, yet existing PEFT methods still struggle to adapt effectively to robot control tasks. To address this gap, we propose **VLA-GSE**, a parameter-efficient VLA fine-tuning framework that improves control adaptation while retaining PEFT's knowledge preservation advantage. Specifically, VLA-GSE (Generalized and Specialized Experts) is initialized by spectrally decomposing the frozen backbone, assigning leading singular components to generalized experts (shared experts) and disjoint residual components to specialized experts (routed experts). This decomposition improves adaptation capacity under a fixed trainable-parameter budget. Across comparable-budget comparison, VLA-GSE updates only 2.51% of full model parameters and consistently outperforms strong FFT and PEFT baselines. It achieves 81.2% average zero-shot success on LIBERO-Plus, preserves pre-trained VLM capability comparably to LoRA on multimodal understanding benchmarks, and improves real-world manipulation success under multiple distribution shifts.

1 INTRODUCTION

Vision-language-action (VLA) models have emerged as a compelling paradigm for general-purpose robotic control, where pre-trained vision-language models (VLMs) are adapted into embodied policies (Firoozi et al., 2025; Ma et al., 2024; Brohan et al., 2023; Kim et al., 2025b; Zhang et al., 2026; Wu et al., 2026; Chen et al., 2025). By inheriting strong multimodal priors from foundation VLMs (Dai et al., 2023), these models map visual observations and language instructions to low-level actions, enabling substantial progress in long-horizon manipulation and boarder generalization (Gao et al., 2025; Chen et al., 2024a; Li et al., 2024b; Mees et al., 2022; Ahn et al., 2023; Huang et al., 2025; Shridhar et al., 2023; Black et al., 2024; Jiang et al., 2025; Chen et al., 2024b).

Despite this progress, robust VLM-to-VLA adaptation remains fundamentally challenging. Full fine-tuning (FFT) offers high expressivity but often overfits limited robotic data and degrades pre-trained vision-language competence through catastrophic forgetting (Yang et al., 2026; Li et al., 2024a; Wen et al., 2025; Hancock et al., 2025; Dey et al., 2024; Yang et al., 2025; Yu et al., 2026). Parameter-efficient fine-tuning (PEFT), such as low-rank adaptation (LoRA), is more stable in preserving pre-trained capability (Hancock et al., 2025), yet existing PEFT approaches frequently under-adapt when precise control is required. This raises a central question: *can PEFT be made sufficiently expressive for precise robot-control adaptation while retaining its knowledge-preservation advantage?*

To address this question, we propose **VLA-GSE** (**G**eneralized and **S**pecialized **E**xperts), a parameter-efficient VLA fine-tuning framework designed to strengthen adaptation without sacrificing the knowledge retention behavior of PEFT. VLA-GSE is built from a spectral decomposition of the frozen backbone: leading singular components initialize generalized experts (shared experts), while disjoint residual components initialize specialized experts (routed experts). This decomposition

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

concentrates shared capacity in always-available experts, reduces redundancy in routed adaptation, and enables more flexible expert composition for targeted control behaviors.

This design introduces two optimization issues. First, spectral initialization induces large singular-value imbalance across specialized experts, which leads to uneven update magnitudes and weakly trained experts. We address this with *expert-wise gradient scale balancing*, which compensates for spectral imbalance and harmonizes optimization scales. Second, because routing is input-dependent, the equivalent weight becomes sample-dependent and can drift from the frozen backbone at initialization. We address this with a *backbone weight adjustment* mechanism that preserves the backbone-equivalent weight in expectation.

Empirically, VLA-GSE updates only 2.51% of full model parameters yet consistently outperforms strong baselines under a comparable trainable-parameter budget. On LIBERO-Plus, it achieves 81.2% average zero-shot success, exceeding FFT and the strongest PEFT baseline by 6.3 and 4.4 percentage points, respectively. On multimodal understanding benchmarks, VLA-GSE preserves pre-trained VLM capability comparably to LoRA and outperforms both FFT and co-trained VLA baselines. In real-world evaluation across four manipulation tasks and four distribution shifts, VLA-GSE reaches 82.5% success, 16.7 percentage points above FFT.

2 Related Work and Background

Finetuning VLMs into VLAs. Recent VLA models adapt pre-trained VLMs into robot policies by retaining the multimodal backbone and training additional action modules on robot trajectories (Zhang et al., 2026; Wu et al., 2026; Gao et al., 2025). Some recent architectures, such as OpenVLA-OFT-style designs, further improve inference efficiency via parallel decoding. Since FFT methods are memory consuming, LoRA is explored in VLA fine-tuning works (Kim et al., 2025b; Hancock et al., 2025; Omaisan & Mohamed, 2025). In particular, OpenVLA (Kim et al., 2025b) is among the earliest VLA works to adopt LoRA in VLA fine-tuning for lower training memory consumption. VLM2VLA (Hancock et al., 2025) provides evidence that LoRA better preserves pre-trained VLM knowledge during VLA adaptation. Although LoRA-based methods better preserve pre-trained vision-language knowledge than FFT, they often lack sufficient adaptation capacity for precise embodied control. Our method explicitly targets both goals: preserving pretrained multimodal knowledge while enabling stronger downstream control adaptation.

Parameter-Efficient Fine-Tuning. PEFT has become a practical alternative to FFT mainly in LLM finetuning, with LoRA as the canonical example (Hu et al., 2022). Recent work shows that adapter structure and initialization critically affect adaptation quality: SVD-based methods such as PiSSA, MiLoRA, KaSA, and GOAT exploit different spectral components to better preserve pretrained knowledge or expose task-relevant subspaces (Meng et al., 2024; Wang et al., 2024b;a; Fan et al., 2025), while MoE-style methods such as MoLoRA, AdaMoLE, and HydraLoRA improve capacity through routed low-rank experts (Zadouri et al., 2024; Liu & Luo, 2024; Tian et al., 2024). However, existing PEFT-based VLA finetuning methods are still largely limited to LoRA-style adaptation (Hancock et al., 2025; Omaisan & Mohamed, 2025), and PEFT remains under-explored in VLA finetuning, where VLA models require stronger adaptation for precise embodied control. Motivated by this difference, VLA-GSE is tailored to the control setting, combining a shared generalized expert with routed specialized experts to support both transferable adaptation and input-specific control refinement. We further introduce gradient scale balancing to stabilize optimization across experts initialized from different spectral components.

3 Method

Overview. Our model follows a standard VLM-to-VLA pipeline, where a pre-trained VLM encodes image observations and language instructions, and an OpenVLA-OFT-style action head is used to generate robot actions (Kim et al., 2025a). The action head is fully fine-tuned for control and we apply PEFT only to the VLM backbone: for a pre-trained weight $W_0 \in \mathbb{R}^{m\times n}$, we learn a structured low-rank update while keeping most backbone parameters frozen. On top of this foundation, we propose VLA-GSE, which decomposes backbone adaptation into an always-on generalized expert and multiple routed specialized experts. The design is tailored to VLA finetuning, where the model must

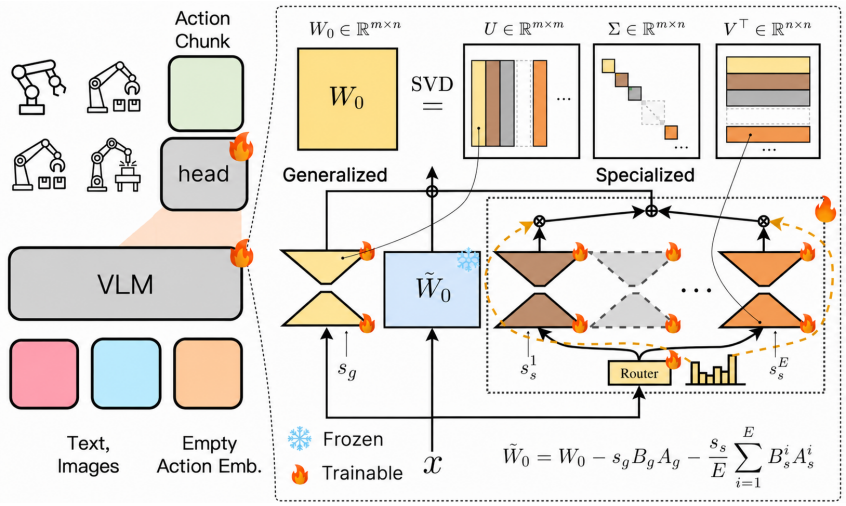

Figure 1: **Overview of the VLA-GSE framework.** (Top Right) VLA-GSE uses an SVD-based adaptive priors scheme to initialize generalized experts and multiple specialized experts based on sorted singular value segments. (Bottom Right) During the forward pass, the output is formed by summing the input transformations induced by the adjusted frozen pre-trained weight $\tilde{W}_0$, the generalized expert, and the dynamically selected top-k specialized experts. Unlike prior MoE LoRA, the generalized expert is always active and initialized from the SVD-based priors. GSE is only applied for the pretrained weight W_0 in the VLM.

both preserve pretrained vision-language knowledge and support precise robot control adaptation. VLA-GSE addresses this through generalized-specialized decomposition, expert-wise gradient scale balancing, and backbone weight adjustment.

3.1 Generalized and Specialized Expert Initialization via SVD

We initialize VLA-GSE from the SVD of the frozen pre-trained weight matrix $W_0 \in \mathbb{R}^{m\times n}$: $W_0 = U\Sigma V^\top$. Sorting singular values in descending order induces a spectral decomposition of W_0, from which we allocate the leading components to the *generalized expert* and the *specialized experts*, as illustrated in Figure 1.

Generalized Expert. We allocate the largest r_g singular values to the generalized expert. This expert is tasked with capturing common knowledge and mitigating redundancy in routed experts. Its initialization is formulated as follows:

$$B_g = \sqrt{\frac{1}{s_g}} U_{1:r_g} \Sigma_{1:r_g}^{1/2} \in \mathbb{R}^{m\times r_g}, \quad A_g = \sqrt{\frac{1}{s_g}} \Sigma_{1:r_g}^{1/2} (V_{1:r_g})^\top \in \mathbb{R}^{r_g\times n}, \quad W_g = s_g B_g A_g, \quad (1)$$

where $U_{1:r_g}$, $\Sigma_{1:r_g}$, and $V_{1:r_g}$ correspond to the top segment of the SVD, s_g is the scaling factor for the generalized expert, and W_g is the generalized expert's effective weight. This design is motivated by the observation that leading singular components capture the dominant energy of the pre-trained weight matrix, which makes the always-on adaptation more effective (Meng et al., 2024).

Specialized Experts. The remaining dominant singular components are evenly partitioned to initialize E specialized experts. Under the adaptive prior strategy, we consider the SVD components indexed by $[r_g + 1, r_g + dE]$ and assign each expert a contiguous block of rank d. Formally, let $\mathcal{I}_i \triangleq \{r_g + (i-1)d + 1, \ \ldots, \ r_g + id\}, i \in \{1, 2, \ldots, E\}$. The i-th specialized expert is then initialized by $U_i = U_{:,\mathcal{I}_i}, \Sigma_i = \Sigma_{\mathcal{I}_i,\mathcal{I}_i}, V_i^\top = V_{\mathcal{I}_i,:}^\top$. Using these designated segments, the initialization is formulated as:

$$B_s^i = \sqrt{\frac{1}{s_s^i}} U_i \Sigma_i^{1/2} \in \mathbb{R}^{m\times d}, \quad A_s^i = \sqrt{\frac{1}{s_s^i}} \Sigma_i^{1/2} V_i^\top \in \mathbb{R}^{d\times n}, \quad W_s^i = s_s^i B_s^i A_s^i, \quad (2)$$

where s_s^i is the i-th expert's scaling factor that normalizes each expert's scale based on the spectral magnitude of its assigned segment, and W_s^i is the effective weight of the i-th specialized expert.

Specialized Expert Selection Method During the forward pass, a router $W_z \in \mathbb{R}^{E\times n}$ maps the input x to routing logits $z(x)$, which determine a sparse subset of specialized experts for x: $p^i(x) = \frac{\exp(z^i(x))}{\sum_{j=1}^{E}\exp(z^j(x))}, z(x) = W_z x$. Let $\Omega_k(x)$ denote the indices of the top-k $p^i(x)$. The routing weights $w^i(x)$ are explicitly defined by applying a softmax function exclusively over the selected top-k logits:

$$w^i(x) = \begin{cases} \frac{\exp(z^i(x))}{\sum_{j\in\Omega_k(x)}\exp(z^j(x))}, & \text{if } i \in \Omega_k(x), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Load Balancing Auxiliary Loss To prevent the sparse router over-selecting a small subset of specialized experts, we introduce an auxiliary load-balancing loss to encourage balanced expert utilization across GSE blocks. For the l-th GSE block, let $f_i^{(l)}$ denote the fraction of tokens assigned to expert i, and let $P_i^{(l)}$ denote its average routing probability over the batch. We define

$$\mathcal{L}_{\text{bal}}^{(l)} = E\sum_{i=1}^{E} f_i^{(l)} P_i^{(l)}, \qquad \mathcal{L}_{\text{final}} = \|\hat{a} - a\|_1 + \alpha\sum_{l=1}^{L}\mathcal{L}_{\text{bal}}^{(l)}. \tag{4}$$

The first term in $\mathcal{L}_{\text{final}}$ is the action regression loss and the second term regularizes expert usage. Detailed explanation of load balancing auxiliary loss is shown in Appendix G

3.2 Gradient Scale Balancing.

To avoid optimization imbalance across specialized experts, we analyze their gradient scales at initialization. Let g_A^i and g_B^i denote the localized gradients of the i-th specialized expert with respect to A_s^i and B_s^i, respectively. To characterize the expected gradient scale, we introduce the left and right second-moment matrices of the gradient: $G_L \triangleq \mathbb{E}[gg^\top], G_R \triangleq \mathbb{E}[g^\top g]$. Here, G_L and G_R describe the second-moment structure of g in the left and right subspaces, respectively. The following theorem gives the exact initialization-time gradient scale for each expert.

Theorem 1 (Expert-wise gradient scale balancing)**.** *Let $w^i(x)$ denote the input-dependent routing coefficient multiplying the contribution of the i-th specialized expert in the aggregated equivalent weight for input x. Under the idealized balanced-routing condition at initialization, define $\rho_w := \mathbb{E}_x\left[(w^i(x))^2\right]$, and assume that ρ_w is identical across all specialized experts $i \in \{1,\dots,E\}$. Then, for the i-th specialized expert, the expected squared Frobenius norms of its localized gradients satisfy*

$$\mathbb{E}\left[\|g_A^i\|_F^2\right] = \rho_w\, s_s^i\, \alpha_i^L, \qquad \mathbb{E}\left[\|g_B^i\|_F^2\right] = \rho_w\, s_s^i\, \alpha_i^R, \tag{5}$$

where

$$\alpha_i^L = \mathrm{Tr}\left(\Sigma_i U_i^\top G_L U_i\right), \qquad \alpha_i^R = \mathrm{Tr}\left(\Sigma_i V_i^\top G_R V_i\right). \tag{6}$$

Therefore, under balanced routing, the optimization scale of expert i is determined jointly by its spectral magnitude Σ_i, the projected second-order gradient energy on its corresponding singular subspaces, and a shared routing second-moment factor ρ_w. Moreover, if there exist constants $\kappa_L, \kappa_R > 0$ such that[1]

$$\alpha_i^L = \kappa_L\,\mathrm{Tr}(\Sigma_i), \qquad \alpha_i^R = \kappa_R\,\mathrm{Tr}(\Sigma_i), \qquad \forall i \in \{1,\dots,E\}, \tag{7}$$

then choosing

$$s_s^i = s_{\text{base}} \cdot \frac{C}{\mathrm{Tr}(\Sigma_i)} \tag{8}$$

equalizes both $\mathbb{E}[\|g_A^i\|_F^2]$ and $\mathbb{E}[\|g_B^i\|_F^2]$ across all specialized experts, since the additional factor ρ_w is shared by all experts under the balanced-routing assumption. Here s_{base} is a global base scaling hyperparameter and C is a normalization constant, e.g., $C = \frac{1}{E}\sum_{j=1}^{E}\mathrm{Tr}(\Sigma_j)$.

Theorem 1 shows that the trace-inverse scaling rule in Eq. (8) is justified whenever the projected second-order gradient statistics are approximately uniform across experts, as specified in Eq. (7). The detailed proof is provided in Appendix B.

[1]This is satisfied, for example, if $G_L \propto I$ and $G_R \propto I$. For a related, but distinct, isotropic gradient-covariance idealization, see Jastrzębski et al. (2017).

3.3 BACKBONE WEIGHT ADJUSTMENT

Unlike LoRA-style zero perturbation at initialization, VLA-GSE induces a non-zero low-rank perturbation at initialization, which would otherwise shift the weight away from W_0 before fine-tuning. To preserve the pre-trained backbone as much as possible at initialization, we compensate for the non-zero perturbation introduced by the generalized and specialized experts by adjusting the frozen backbone weight from W_0 to $\tilde{W}_0$. Let $W_{eq}(x)$ denote the input-dependent equivalent weight of the GSE block. Then, it follows that

$$y = W_{eq}(x)x, \qquad W_{eq}(x) = \tilde{W}_0 + s_g B_g A_g + \sum_{i=1}^{E} w^i(x)\, s_s^i B_s^i A_s^i. \tag{9}$$

In principle, one may desire an exact sample-wise equality: $W_{eq}(x) = W_0, \forall x$. However, this is generally impossible because the selection of specialized experts varies with x through $w^i(x)$. Accordingly, we aim to maintain consistency with the pretrained weight *in expectation* under the routing distribution at initialization. At random initialization, the router is symmetric across specialized experts in expectation. Since they also satisfy $\sum_{i=1}^{E} w^i(x) = 1$, we have $\mathbb{E}_x[w^i(x)] = \frac{1}{E}, \forall i \in \{1, \ldots, E\}$. It follows that the expected weight of the specialized experts is

$$\mathbb{E}_x\left[\sum_{i=1}^{E} w^i(x)\, s_s^i B_s^i A_s^i\right] = \sum_{i=1}^{E} \mathbb{E}_x[w^i(x)]\, s_s^i B_s^i A_s^i = \frac{1}{E}\sum_{i=1}^{E} s_s^i B_s^i A_s^i. \tag{10}$$

This motivates defining the expected residual shift introduced at initialization:

$$W_{res} = s_g B_g A_g + \frac{1}{E}\sum_{i=1}^{E} s_s^i B_s^i A_s^i, \tag{11}$$

where W_{res} denotes the total expected offset of the initialized GSE weight relative to the original backbone. We then compensate for this offset by redefining the adjusted frozen weight as

$$\tilde{W}_0 = W_0 - W_{res} = W_0 - s_g B_g A_g - \frac{1}{E}\sum_{i=1}^{E} s_s^i B_s^i A_s^i. \tag{12}$$

Under the backbone weight adjustment in Eq. 12, $W_{eq}(x)$ is aligned with W_0 in expectation at initialization, where the detailed proof is shown in Appendix C. The full initialization and forward-pass procedure is provided in Appendix E.

4 SIMULATION EXPERIMENTS

4.1 SETTINGS

Model Architecture and Initialization. In our experiments, we adopt Qwen3-VL-4B-Instruct as the foundation VLM backbone. To efficiently adapt the VLM into a VLA model, we apply our proposed VLA-GSE framework. Specifically, the low-rank dimension is set to $r = 16$. Each GSE block consists of 8 rank-2 experts in total, which explicitly includes 1 generalized expert to anchor the foundational representations, and 7 specialized experts utilizing a Top-2 gating strategy. Scaling across the architecture, this configuration deploys a total of 356 generalized experts and 2492 specialized experts. The initialization strictly follows the SVD-based approach. For the action head, we use the MLP architecture and parallel decoding method from OpenVLA-OFT (Kim et al., 2025a).

Optimization and Hyperparameters. The model is trained from Qwen3-VL-4B-Instruct for a total of 80,000 optimization steps. The batch size per GPU is set to 16, resulting in an effective total batch size of 128 across 8 NVIDIA A100 GPUs. To ensure stable adaptation while preventing the catastrophic forgetting of pre-trained visual-semantic knowledge, we apply decoupled learning rates: the learning rate for the GSE parameters in VLM is set to 1×10^{-5}, while the newly initialized action MLP head uses a learning rate of 1×10^{-4}. Furthermore, the auxiliary load-balancing loss weight α is set to 0.01. We set the generalized scaling factor as $s_g = 2$ and set the i-th specialized

Table 1: LIBERO-Plus benchmark zero-shot performance. Results are shown in success rate (%).

Model	Camera	Robot	Language	Light	Background	Noise	Layout	Total
OpenVLA (Kim et al., 2025b)	0.8	3.5	23.0	8.1	34.8	15.2	28.5	15.6
WorldVLA (Cen et al., 2025)	0.1	27.9	41.6	43.7	17.1	10.9	38.0	25.0
NORA (Hung et al., 2025)	2.2	37.0	65.1	45.7	58.6	12.8	62.1	39.0
UniVLA (Bu et al., 2025)	1.8	46.2	69.6	69.0	81.0	21.2	31.9	42.9
π_0 (Black et al., 2024)	13.8	6.0	58.8	85.0	81.4	79.0	68.9	53.6
π_0-Fast (Pertsch et al., 2025)	65.1	21.6	61.0	73.2	73.2	74.4	68.8	61.6
OpenVLA-OFT (Kim et al., 2025a)	56.4	31.9	79.5	88.7	93.3	75.8	74.2	69.6
RIPT-VLA (Tan et al., 2025)	55.2	31.2	77.6	88.4	91.6	73.5	74.2	68.4
ABot-M0 (Yang et al., 2026)	60.4	67.9	86.4	96.2	91.6	86.4	**82.6**	80.5
VLANeXt (Wu et al., 2026)	**71.6**	63.4	81.4	93.7	91.8	**89.5**	77.7	80.1
VLA-GSE (Ours)	64.4	**68.5**	**88.8**	**97.3**	**97.3**	79.4	**82.6**	**81.2**

scaling factor as $s_s^i = \frac{2\sum_{j=1}^{E}\mathrm{Tr}(\Sigma_j)}{E\cdot\mathrm{Tr}(\Sigma_i)}$. In this way, the scaling of each specialized expert is inversely proportional to its spectral magnitude. A comprehensive summary of the detailed hyperparameter settings and model configurations is provided in Appendix Table 9.

Parameter Efficiency. Training VLA-GSE is highly parameter-efficient. Among the 4,551.85M total model parameters, we freeze 4,437.82M non-GSE parameters from the base VLM and optimize only 114.04M parameters (2.51% of the full model). Specifically, the trainable parameters consist of 48.41M parameters in the GSE modules and 65.62M parameters in the action head.

4.2 LIBERO-Plus Benchmark Results

We use LIBERO-Plus (Fei et al., 2025) as the simulation benchmark, which evaluates VLA models' generalization ability across seven perturbation dimensions. VLA-GSE and the controlled baselines in Table 2 use the combined training sets of all four LIBERO suites (Liu et al., 2023). Table 1 shows that VLA-GSE achieves the best overall zero-shot performance, obtaining 81.2% average success and outperforming the strongest prior baselines, including ABot-M0 (80.5%) and VLANeXt (80.1%). The improvement is broad rather than isolated: VLA-GSE delivers the strongest average robustness on Robot, Language, Light, and Background, while matching the best Layout performance. At the suite level, it attains 90.3% on Spatial and 86.2% on Object, and remains stable on the more challenging Goal and Long suites with 74.2% and 74.1%, respectively. Overall, these results indicate that VLA-GSE improves not only aggregate success rate, but also robustness under diverse test-time perturbations. The full suite-wise results of VLA-GSE are provided in Appendix D.5.

Table 2 compares fine-tuning strategies in LIBERO-Plus zero-shot generalization using the Qwen3-VL-4B-Instruct backbone. For a fair comparison, all PEFT methods use roughly 2.51% (114M) trainable parameters and are independently tuned for optimal performance. Notably, FFT (74.9%) underperforms several PEFT baselines, consistent with the degradation of pretrained capabilities in Table 3 and prior work on *catastrophic forgetting* (Hancock et al., 2025; Yu et al., 2026). VLA-GSE achieves the highest reported overall success rate of 81.2%. It exceeds FFT and the strongest PEFT baseline, GOAT, by 6.3 and 4.4 percentage points, respectively, in the reported Total score. The category-wise point estimates also favor VLA-GSE over the compared fine-tuning baselines.

We qualitatively illustrate the attention maps in the challenging **dark lighting** setting of LIBERO-Plus. Figure 2 shows the last transformer layer attention maps for different models under the instruction: "Pick up the black bowl between the plate and the ramekin and place it on the plate." Both FFT and LoRA are fine-tuned from the same Qwen3-VL-4B-Instruct backbone. VLA-GSE exhibits concentrated attention on the **graspable edges** of the correct target object and successfully completes the task, indicating that it not only identifies the target specified by the dense spatial instruction, but also captures the manipulation affordance required for grasping, namely the appropriate edge positions for the gripper. In contrast, the baselines show either scattered or mislocalized attention, leading to incorrect object selection or failed grasping. These results suggest that GSE better preserves the backbone's visual-semantic grounding while acquiring task-specific manipulation behavior. More qualitative examples are provided in Figure 4.

Table 2: Finetuning method comparison based on LIBERO-Plus benchmark zero-shot performance. Results are shown in success rate (%).

Model	Params (%)	Camera	Robot	Language	Light	Background	Noise	Layout	Total
FFT (Full Fine-Tuning)	100.00	45.8	60.5	86.9	95.9	94.7	74.1	79.2	74.9
LoRA (Hu et al., 2022)	2.58	44.3	53.6	82.1	92.9	91.1	67.1	67.2	69.2
rsLoRA (Kalajdzievski, 2023)	2.52	46.0	58.9	85.1	93.6	92.4	71.6	76.5	73.1
DoRA (Liu et al., 2024)	2.54	49.7	61.0	86.6	95.7	94.4	73.4	78.6	75.3
PiSSA (Meng et al., 2024)	2.54	48.3	59.4	86.0	95.4	94.0	72.8	78.1	74.5
MoLoRA (Zadouri et al., 2024)	2.56	52.2	61.5	87.5	96.0	95.0	74.2	79.0	76.2
AdaMoLE (Liu & Luo, 2024)	2.53	50.2	60.8	87.8	95.8	94.4	73.1	78.6	75.5
HydraLoRA (Tian et al., 2024)	2.55	51.3	60.1	86.4	95.4	94.1	73.0	78.2	75.2
MiLoRA (Wang et al., 2025)	2.52	51.1	60.1	86.3	94.9	93.9	73.0	77.7	75.0
GOAT (Fan et al., 2025)	2.54	54.5	61.8	87.7	96.2	95.2	74.6	79.3	76.8
VLA-GSE (Ours)	2.51	**64.4**	**68.5**	**88.8**	**97.3**	**97.3**	**79.4**	**82.6**	**81.2**

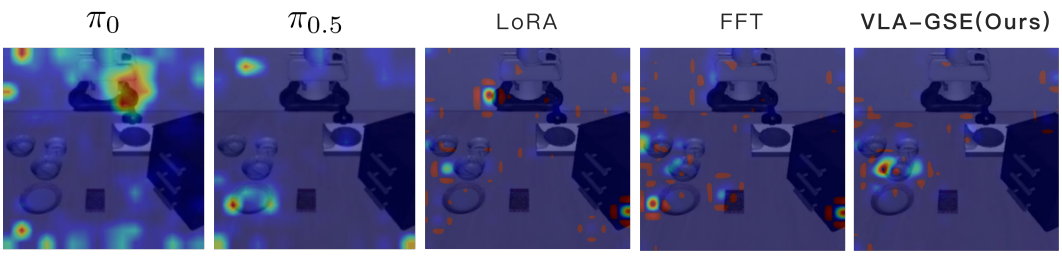

Figure 2: Attention map visualizations for different models under the LIBERO-Plus **dark lighting** environment. The instruction is "Pick up the black bowl between the plate and the ramekin and place it on the plate." Our VLA-GSE method exhibits precise attention on the **graspable edges** of the target object, whereas π_0, $\pi_{0.5}$, FFT, and LoRA show scattered attention or focus on incorrect objects.

4.3 Knowledge Retention after VLA Fine-Tuning in Terms of Multimodal Understanding

We further evaluate the base VLM and VLA-adapted variants on a diverse set of multimodal understanding benchmarks. As shown in Table 3, FFT leads to a clear and consistent degradation across benchmarks, indicating severe catastrophic forgetting of pretrained vision–language capability during VLM-to-VLA transfer. A similar trend is also observed for co-trained VLA models such as $\pi_{0.5}$, whose multimodal understanding performance remains far from that of strong general-purpose VLMs, suggesting that preserving VLM knowledge is still challenging under FFT embodied training.

In contrast, both Qwen3-VL-LoRA and VLA-GSE remain consistently close to the base Qwen3-VL-4B-Instruct on general multimodal benchmarks, showing substantially better preservation of pretrained multimodal knowledge than FFT. Importantly, VLA-GSE achieves a retention level comparable to standard LoRA-based PEFT, which suggests that its stronger embodied performance does not come from sacrificing the original vision–language capability. Instead, VLA-GSE preserves pretrained VLM knowledge similarly well to LoRA, while providing stronger adaptation capacity for downstream robotic control as shown in Table 2.

4.4 Ablation Study

To evaluate the contributions of our core designs, we conduct an ablation study on the LIBERO-Plus zero-shot Long suite. The "w.o. SVD Initialization" variant instead uses a standard LoRA-style initialization, with each B_s^i Gaussian initialized and each A_s^i zero initialized. For the expert routing ablations, we maintain a constant total number of experts to ensure a fair comparison of model capacity. Specifically, the "w.o. Generalized Experts" setting replaces the generalized expert with an additional specialized expert, forcing the model to rely entirely on routing mechanisms. Conversely, the "w.o. Specialized Experts" variant replaces all specialized experts with generalized ones. The "w.o. Auxiliary Loss" variant removes the load-balancing objective during training. Finally, the "w.o. Gradient Scaling" variant replaces the expert-wise trace-inverse scaling in Eq. 8 with a constant scaling factor, setting all s_s^i to 2.

Table 3: Multimodal understanding evaluation. Comparison of VLMs and VLAs across multimodal understanding benchmarks.

Method	#Params	MMMU	MMStar	OCR Bench	MMB-en	Doc VQA	Info VQA	AI2D	RealWorld QA
Prismatic VLM Family									
Prismatic VLM (Karamcheti et al., 2024)	7B	35.0	38.8	32.0	66.2	17.5	19.7	54.6	30.8
OpenVLA (Kim et al., 2025b)	7B	26.3	0	0	0	0	0	0	0
ECoT (Zawalski et al., 2025)	7B	26.6	0	0.01	3.7	0	0	0	25.6
Gemma-3 Family									
Gemma-3-4B-IT (Gemma Team, 2025)	4B	39.3	37.1	70.2	68.6	68.8	40.9	70.5	44.0
Gemma-3-12B-IT (Gemma Team, 2025)	12B	46.0	46.3	75.0	76.9	80.6	50.4	78.5	50.6
VLM2VLA-AT (Hancock et al., 2025)	12B	45.9	45.2	65.5	70.9	74.6	44.8	74.1	44.5
VLM2VLA (Hancock et al., 2025)	12B	42.7	48.0	63.9	68.5	78.4	46.2	74.0	43.3
Other Baseline VLAs									
MolmoAct (Lee et al., 2025)	7B	28.4	1.2	52.7	55.1	58.7	41.9	2.0	8.6
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	3B	24.0	21.7	6.8	6.8	4.6	7.7	27.0	2.7
Qwen3-VL Family									
Qwen3-VL-4B-Instruct (Bai et al., 2025)	4B	**53.2**	**69.8**	**88.1**	**85.1**	**95.3**	**80.3**	**84.1**	**70.9**
Qwen3-VL-FFT (Bai et al., 2025)	4B	35.6	31.4	43.2	50.8	55.1	37.6	48.3	33.7
Qwen3-VL-LoRA (Bai et al., 2025)	4B	51.8	68.1	83.3	83.0	94.1	79.2	83.0	69.8
VLA-GSE (Ours)	4B	51.1	67.8	85.6	82.4	92.8	78.5	83.4	70.1

Table 4: Ablation study on Long suite of LIBERO-Plus benchmark zero-shot performance. Results are shown in success rate (%).

Model	Camera	Robot	Language	Light	Background	Noise	Layout	Total
w.o. SVD Initialization	25.8	41.9	80.8	84.8	88.1	51.0	70.3	60.9
w.o. Generalized Experts	25.1	50.2	84.4	95.7	85.9	62.5	81.9	67.2
w.o. Specialized Experts	24.0	48.8	82.8	88.7	90.6	52.1	71.4	63.1
w.o. Auxiliary Loss	32.2	59.7	90.0	91.8	96.8	62.9	85.4	72.0
w.o. Gradient Scaling	32.5	59.5	90.2	91.6	96.5	63.1	85.6	72.1
VLA-GSE (Ours)	**35.8**	**61.0**	**91.0**	**94.9**	**98.3**	**65.2**	**87.3**	**74.1**

Our full GSE framework achieves the highest overall success rate of 74.1%, outperforming all tested ablated variants. The most severe degradation occurs without SVD initialization. Replacing either expert type also causes notable drops, supporting the utility of both adaptation paths. These controls approximately preserve the expert pool and trainable capacity, but not active paths or per-token FLOPs; they therefore do not isolate expert roles independently of computation. Removing the auxiliary loss reduces overall success to 72.0%, while replacing trace-inverse scaling with a constant factor gives 72.1%, indicating modest task benefits from both components. Appendix H.6 reports complementary update-balancing diagnostics and their scope.

5 Real-World Experiments

To further validate our framework, we conduct extensive real-world experiments on the AgileX PiPER robot. The robot is equipped with a fixed front-view camera and a wrist camera for visual observation. As illustrated in Figure 3, the real-world benchmark consists of four tasks: *Pour water into bowl*, *Pick spoon into plate*, *Put carrot into bowl*, and *Take pen out of pen container*. Each task is further evaluated under four types of distribution shifts: lighting changes, language variations, cluttered layout variations, and background tablecloth changes. For each generalization dimension, we collect 15 test cases, resulting in 60 trials per task.

Table 5 summarizes the real-world generalization results. VLA-GSE achieves the best performance on all 4 tasks, yielding the highest average success rate of 82.5%. Among the baselines in this table, $\pi_{0.5}$ is the strongest, followed by Qwen3-VL-FFT, Qwen3-VL-LoRA, and π_0, while OpenVLA-OFT performs the worst. Notably, Qwen3-VL-LoRA consistently underperforms Qwen3-VL-FFT across all tasks, which aligns with the trend observed in Table 2: parameter-efficient adaptation improves

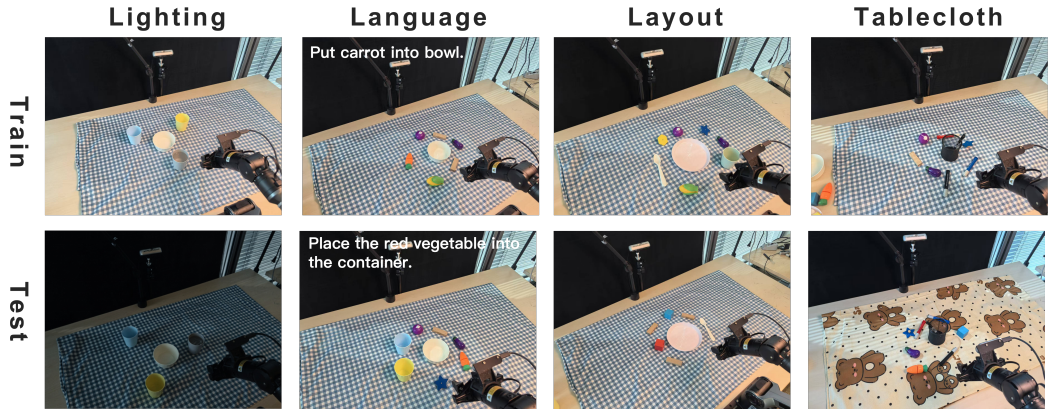

Figure 3: **Real-world generalization settings.** The four train-test pairs correspond to variations in *lighting*, *language*, *clutter layout*, and *background tablecloth*, respectively. For *language variations*, the training instruction is *"Put carrot into bowl."* At test time, we replace it with *"Place the red vegetable into the container."*

Table 5: Results on **real-world** generalization. Each task contains 60 evaluation episodes, including 4 OOD settings: variations in *lighting*, *language*, *clutter layout*, and *background tablecloth*. Results are reported in success rate (%).

Model	Pour water into bowl	Pick spoon into plate	Put carrot into bowl	Take pen out of pen container	Average
OpenVLA-OFT (Kim et al., 2025a)	38.3	41.7	48.3	43.3	42.9
π_0 (Black et al., 2024)	51.7	55.0	60.0	56.7	55.8
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	73.3	76.7	75.0	71.7	74.2
Qwen3-VL-LoRA (StarVLA, 2026)	58.3	61.7	66.7	60.0	61.7
Qwen3-VL-FFT (StarVLA, 2026)	63.3	66.7	70.0	63.3	65.8
VLA-GSE (Ours)	**81.7**	**85.0**	**83.3**	**80.0**	**82.5**

transferability, but FFT remains more effective than vanilla low-rank adaptation for acquiring robust action grounding. By contrast, VLA-GSE delivers a clear margin over all baselines, indicating that its generalized-specialized decomposition is better suited for handling real-world distribution shifts in illumination, background, clutter layout, and instruction phrasing. We also compare real-world inference time between VLA-GSE and baselines in Appendix D.4.

6 Limitations

While the additional backbone and decoder results suggest compatibility, task adaptation remains robotics-specific, and cross-backbone and real-world studies use single training seeds. A similar caution applies to mechanism attribution: the scaling argument relies on initialization-time assumptions, and expert-role ablations do not match active FLOPs. Together, these boundaries motivate multi-seed, cross-platform evaluation and matched-compute controls.

7 Conclusion

We present VLA-GSE, a parameter-efficient framework for adapting pretrained VLMs to robotic control while retaining multimodal knowledge. Its SVD-based decomposition combines an always-active generalized expert and routed specialists, with trace-inverse scaling and expectation-based backbone adjustment. The reported simulation and real-world comparisons favor this design over the evaluated FFT and PEFT baselines, while multimodal performance remains comparable to LoRA. These findings motivate organizing trainable capacity for shared and task-dependent adaptation, rather than focusing on parameter count alone, as a practical route toward more effective VLA fine-tuning.

Reproducibility Statement

The implementation is available in the anonymous repository at `https://anonymous.4open.science/r/VLA-GSE/`. Section 3 specifies the method, and Appendix E provides initialization and forward-pass pseudocode. Simulation and real-world evaluation settings are described in Sections 4 and 5; Appendix D.6 and Table 9 provide training configurations, hyperparameters, and real-world task descriptions. Proofs of the gradient-scale result and expectation-based backbone weight adjustment are provided in Appendices B and C, respectively. Additional experimental protocols, measured training costs, uncertainty analyses, and diagnostic controls are provided in Appendix H.

AI Use Statement

LLMs were used for language polishing in selected sections of this paper. An AI assistant was also used for manuscript preparation, including conversion to the ICLR LaTeX template, anonymity and pagination checks, organizing and summarizing previously reported experimental results, checking numerical transcription, summary calculations, and reference metadata, and drafting the reproducibility and AI-use statements. An AI-based paper-feedback tool was used to obtain feedback on methodology and presentation; automated feedback was not treated as empirical evidence. The authors take responsibility for the final content of this work, including all AI-assisted text and artifacts.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 287–318, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025. doi: 10.48550/arXiv.2511.21631.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.010.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 2165–2183, 2023.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to act anywhere with task-centric latent actions, 2025. URL https://arxiv.org/abs/2505.06111.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards autoregressive action world model, 2025. URL https://arxiv.org/abs/2506.21539.

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

Xiaoyu Chen, Junliang Guo, Tianyu He, Chuheng Zhang, Pushi Zhang, Derek Cathera Yang, Li Zhao, and Jiang Bian. Igor: Image-goal representations are the atomic control units for foundation models in embodied ai. *arXiv preprint arXiv:2411.00785*, 2024b.

Xiaoyu Chen, Hangxing Wei, Pushi Zhang, Chuheng Zhang, Kaixin Wang, Yanjiang Guo, Rushuai Yang, Yucen Wang, Xinquan Xiao, Li Zhao, et al. Villa-x: enhancing latent action modeling in vision-language-action models. *arXiv preprint arXiv:2507.23682*, 2025.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin CM Burchfiel, and Shuran Song. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.026.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Sombit Dey, Jan-Nico Zaech, Nikolay Nikolov, Luc Van Gool, and Danda Pani Paudel. Revla: Reverting visual domain limitation of robotic foundation models. *arXiv preprint arXiv:2409.15250*, 2024.

Chenghao Fan, Zhenyi Lu, Sichen Liu, Chengfeng Gu, Xiaoye Qu, Wei Wei, and Yu Cheng. Make lora great again: Boosting lora with adaptive singular values and mixture-of-experts optimization alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/fan25g.html.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, Brian Ichter, Danny Driess, Jiajun Wu, Cewu Lu, and Mac Schwager. Foundation models in robotics: Applications, challenges, and the future. *The International Journal of Robotics Research*, 44(5), 2025. doi: 10.1177/02783649241281508.

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

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025. doi: 10.48550/arXiv.2503.19786.

Asher J. Hancock, Xindi Wu, Lihan Zha, Olga Russakovsky, and Anirudha Majumdar. Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting. *arXiv preprint arXiv:2509.22195*, 2025.

Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, and Yuntao Chen. Dita: Scaling diffusion transformer for generalist vision-language-action policy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7686–7697, October 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. *arXiv preprint arXiv:2507.16815*, 2025.

Chia-Yu Hung, Qi Sun, Pengfei Hong, Amir Zadeh, Chuan Li, U-Xuan Tan, Navonil Majumder, and Soujanya Poria. Nora: A small open-sourced generalist vision language action model for embodied tasks, 2025. URL https://arxiv.org/abs/2504.19854.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in SGD. *arXiv preprint arXiv:1711.04623*, 2017. URL https://arxiv.org/abs/1711.04623v1. arXiv v1.

Yuhua Jiang, Shuang Cheng, Yan Ding, Feifei Gao, and Biqing Qi. AsyncVLA: Asynchronous flow matching for Vision-Language-Action models, 2025. URL https://arxiv.org/abs/2511.14148.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023. URL https://arxiv.org/abs/2312.03732.

Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic VLMs: Investigating the design space of visually-conditioned language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23123–23144. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/karamcheti24a.html.

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

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713, 2025b.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025. doi: 10.48550/arXiv.2508.07917.

Wei Li, Renshan Zhang, Rui Shao, Jie He, and Liqiang Nie. CogVLA: Cognition-aligned vision-language-action models via instruction-driven routing & sparsification. *Advances in Neural Information Processing Systems*, 2025.

Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024a. URL https://arxiv.org/abs/2412.14058v1. arXiv v1.

Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. In *International Conference on Learning Representations*, 2024b.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating Real-World Robot Manipulation Policies in Simulation, 2024c. URL https://arxiv.org/abs/2405.05941.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/8c3c666820ea055a77726d66fc7d447f-Abstract-Datasets_and_Benchmarks.html.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32100–32121. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24bn.html.

Zefang Liu and Jiahua Luo. Adamole: Fine-tuning large language models with adaptive mixture of low-rank adaptation experts. In *Conference on Language Modeling*, 2024.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai. *arXiv preprint arXiv:2405.14093*, 2024.

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

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 121038–121072, 2024. doi: 10.52202/079017-3846. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/db36f4d603cc9e3a2a5e10b93e6428f2-Abstract-Conference.html.

NVIDIA GEAR Team. GR00T N1.6: An improved open foundation model for generalist humanoid robots. https://research.nvidia.com/labs/gear/gr00t-n1_6/, 2025. Accessed: 2026-04-28.

Abdullah Yahya Abdullah Omaisan and Ibrahim Sheikh Mohamed. Towards accessible physical ai: Lora-based fine-tuning of vla models for real-world robot control. *arXiv preprint arXiv:2512.11921*, 2025. URL https://arxiv.org/abs/2512.11921.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models, 2025. URL https://arxiv.org/abs/2501.09747.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Moritz Reuss, Hongyi Zhou, Marcel Rühle, Ömer Erdinç Yağmurlu, Fabian Otto, and Rudolf Lioutikov. Flower: Democratizing generalist robot policies with efficient vision-language-flow models. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3736–3761. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/reuss25a.html.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, 2023.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing, 2026. URL https://arxiv.org/abs/2604.05014.

Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, Siyang Qin, Reeve Ingle, Emanuele Bugliarello, Sahar Kazemzadeh, Thomas Mesnard, Ibrahim Alabdulmohsin, Lucas Beyer, and Xiaohua Zhai. PaliGemma 2: A family of versatile VLMs for transfer, 2024. URL https://arxiv.org/abs/2412.03555.

Shuhan Tan, Kairan Dou, Yue Zhao, and Philipp Krähenbühl. Interactive post-training for vision-language-action models, 2025. URL https://arxiv.org/abs/2505.17016.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. Hydralora: An asymmetric lora architecture for efficient fine-tuning. *arXiv preprint arXiv:2404.19245*, 2024.

Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. BridgeData V2: A dataset for robot learning at scale, 2024. URL https://arxiv.org/abs/2308.12952v3. arXiv v3, 17 January 2024.

Fan Wang, Juyong Jiang, Chansung Park, Sunghun Kim, and Jing Tang. Kasa: Knowledge-aware singular-value adaptation of large language models. *arXiv preprint arXiv:2412.06071*, 2024a.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. *arXiv preprint arXiv:2406.09044*, 2024b.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. Milora: Harnessing minor singular components for parameter-efficient llm finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4823–4836, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.248. URL https://aclanthology.org/2025.naacl-long.248/.

Junjie Wen, Yichen Zhu, Minjie Zhu, Zhibin Tang, Jinming Li, Zhongyi Zhou, Xiaoyu Liu, Chaomin Shen, Yaxin Peng, and Feifei Feng. Diffusionvla: Scaling robot foundation models via unified diffusion and autoregression. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66558–66574, 2025.

Xiao-Ming Wu, Bin Fan, Kang Liao, Jian-Jian Jiang, Runze Yang, Yihang Luo, Zhonghua Wu, Wei-Shi Zheng, and Chen Change Loy. Vlanext: Recipes for building strong vla models. *arXiv preprint arXiv:2602.18532*, 2026.

Shuai Yang, Hao Li, Bin Wang, Yilun Chen, Yang Tian, Tai Wang, Hanqing Wang, Feng Zhao, Yiyi Liao, and Jiangmiao Pang. Instructvla: Vision-language-action instruction tuning from understanding to manipulation. *arXiv preprint arXiv:2507.17520*, 2025.

Yandan Yang, Shuang Zeng, Tong Lin, Xinyuan Chang, Dekang Qi, Junjin Xiao, Haoyun Liu, Ronghan Chen, Yuzhi Chen, Dongjie Huo, Feng Xiong, Xing Wei, Zhiheng Ma, and Mu Xu. Abot-m0: Vla foundation model for robotic manipulation with action manifold learning. *arXiv preprint arXiv:2602.11236*, 2026.

Bin Yu, Shijie Lian, Xiaopeng Lin, Yuliang Wei, Zhaolong Shen, Changti Wu, Yuzhuo Miao, Xinming Wang, Bailing Wang, Cong Huang, and Kai Chen. Twinbrainvla: Unleashing the potential of generalist vlms for embodied tasks via asymmetric mixture-of-transformers. *arXiv preprint arXiv:2601.14133*, 2026.

Ted Zadouri, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. Pushing mixture of experts to the limit: Extremely parameter efficient moe for instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3157–3181. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/zawalski25a.html.

Jianke Zhang, Xiaoyu Chen, Qiuyue Wang, Mingsheng Li, Yanjiang Guo, Yucheng Hu, Jiajun Zhang, Shuai Bai, Junyang Lin, and Jianyu Chen. Vlm4vla: Revisiting vision-language-models in vision-language-action models. *arXiv preprint arXiv:2601.03309*, 2026.

Jinliang Zheng, Jianxiong Li, Zhihao Wang, Dongxiu Liu, Xirui Kang, Yuchun Feng, Yinan Zheng, Jiayin Zou, Yilun Chen, Jia Zeng, Ya-Qin Zhang, Jiangmiao Pang, Jingjing Liu, Tai Wang, and Xianyuan Zhan. X-vla: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. *arXiv preprint arXiv:2510.10274*, 2025.

A Technical Appendices and Supplementary Material

A.1 Impact Statement

This paper presents work whose goal is to advance the field of machine learning. We do not identify any specific impacts of this work that require particular emphasis here.

B Proof of Theorem 1

Proof. For the i-th specialized expert, its effective contribution to the aggregated equivalent weight is $w^i(x)W_s^i$, where $W_s^i = s_s^i B_s^i A_s^i$. Let $g = \frac{\partial \mathcal{L}_{\text{final}}}{\partial W_{eq}(x)}$ denote the gradient of the loss with respect to the aggregated equivalent weight. By the chain rule, the localized gradients with respect to A_s^i and B_s^i are

$$g_A^i = \frac{\partial \mathcal{L}_{\text{final}}}{\partial A_s^i} = \frac{\partial \big(w^i(x)W_s^i\big)}{\partial A_s^i} \frac{\partial \mathcal{L}_{\text{final}}}{\partial \big(w^i(x)W_s^i\big)} = w^i(x)\, s_s^i (B_s^i)^\top g, \tag{13}$$

$$g_B^i = \frac{\partial \mathcal{L}_{\text{final}}}{\partial B_s^i} = \frac{\partial \mathcal{L}_{\text{final}}}{\partial \big(w^i(x)W_s^i\big)} \frac{\partial \big(w^i(x)W_s^i\big)}{\partial B_s^i} = w^i(x)\, s_s^i g (A_s^i)^\top. \tag{14}$$

Using the initialization

$$B_s^i = \sqrt{\frac{1}{s_s^i}}\, U_i \Sigma_i^{1/2}, \qquad A_s^i = \sqrt{\frac{1}{s_s^i}}\, \Sigma_i^{1/2} V_i^\top, \tag{15}$$

we obtain

$$g_A^i = w^i(x)\, s_s^i \left(\sqrt{\frac{1}{s_s^i}}\, U_i \Sigma_i^{1/2}\right)^\top g = w^i(x)\, \sqrt{s_s^i}\, \Sigma_i^{1/2} U_i^\top g, \tag{16}$$

$$g_B^i = w^i(x)\, s_s^i g \left(\sqrt{\frac{1}{s_s^i}}\, \Sigma_i^{1/2} V_i^\top\right)^\top = w^i(x)\, \sqrt{s_s^i}\, g V_i \Sigma_i^{1/2}. \tag{17}$$

We first analyze the squared Frobenius norm of g_A^i:

$$\begin{aligned} \|g_A^i\|_F^2 &= \mathrm{Tr}\big((g_A^i)^\top g_A^i\big) \\ &= \mathrm{Tr}\left(\left(w^i(x)\, \sqrt{s_s^i}\, \Sigma_i^{1/2} U_i^\top g\right)^\top \left(w^i(x)\, \sqrt{s_s^i}\, \Sigma_i^{1/2} U_i^\top g\right)\right) \\ &= (w^i(x))^2 s_s^i\, \mathrm{Tr}\big(g^\top U_i \Sigma_i U_i^\top g\big)\,. \end{aligned} \tag{18}$$

Taking expectation over the training batch distribution and using the cyclic property of the trace gives

$$\mathbb{E}\big[\|g_A^i\|_F^2\big] = s_s^i\, \mathbb{E}\big[(w^i(x))^2 \mathrm{Tr}\big(g^\top U_i \Sigma_i U_i^\top g\big)\big]\,. \tag{19}$$

To analyze the expected squared gradient norm, we consider the idealized balanced-routing condition and define

$$\rho_w := \mathbb{E}\big[(w^i(x))^2\big]\,. \tag{20}$$

Under balanced routing, ρ_w is identical across experts and thus does not depend on i. Therefore, when taking expectations of $\|g_A^i\|_F^2$ or $\|g_B^i\|_F^2$, the routing term contributes only the same multiplicative factor ρ_w for all experts. As a result, it does not affect the relative scaling across experts, and the trace-inverse scaling rule remains unchanged. Absorbing this shared routing second-moment factor, we obtain

$$\begin{aligned} \mathbb{E}\big[\|g_A^i\|_F^2\big] &= \rho_w\, s_s^i\, \mathrm{Tr}\big(U_i \Sigma_i U_i^\top\, \mathbb{E}[g g^\top]\big) \\ &= \rho_w\, s_s^i\, \mathrm{Tr}\big(\Sigma_i U_i^\top G_L U_i\big)\,. \end{aligned} \tag{21}$$

Define

$$\alpha_i^L = \mathrm{Tr}\big(\Sigma_i U_i^\top G_L U_i\big)\,. \tag{22}$$

Then Eq. (21) becomes

$$\mathbb{E}\big[\|g_A^i\|_F^2\big] = \rho_w \, s_s^i \, \alpha_i^L. \tag{23}$$

We next analyze g_B^i. From Eq. (17),

$$\begin{aligned}\|g_B^i\|_F^2 &= \mathrm{Tr}\big((g_B^i)^\top g_B^i\big) \\ &= \mathrm{Tr}\left(\left(w^i(x)\,\sqrt{s_s^i}\,gV_i\Sigma_i^{1/2}\right)^\top \left(w^i(x)\,\sqrt{s_s^i}\,gV_i\Sigma_i^{1/2}\right)\right) \\ &= (w^i(x))^2 s_s^i \,\mathrm{Tr}\big(\Sigma_i V_i^\top g^\top g V_i\big)\,. \end{aligned} \tag{24}$$

Taking expectation yields

$$\begin{aligned}\mathbb{E}\big[\|g_B^i\|_F^2\big] &= s_s^i \,\mathbb{E}\big[(w^i(x))^2 \mathrm{Tr}\big(\Sigma_i V_i^\top g^\top g V_i\big)\big] \\ &= \rho_w \, s_s^i \,\mathrm{Tr}\big(\Sigma_i V_i^\top \mathbb{E}[g^\top g] V_i\big) \\ &= \rho_w \, s_s^i \,\mathrm{Tr}\big(\Sigma_i V_i^\top G_R V_i\big)\,. \end{aligned} \tag{25}$$

Define

$$\alpha_i^R = \mathrm{Tr}\big(\Sigma_i V_i^\top G_R V_i\big)\,. \tag{26}$$

Then Eq. (25) becomes

$$\mathbb{E}\big[\|g_B^i\|_F^2\big] = \rho_w \, s_s^i \, \alpha_i^R. \tag{27}$$

Equations (23) and (27) prove the first part of the theorem under the idealized balanced-routing assumption at initialization. Since ρ_w is shared across experts, it does not affect the relative scaling across experts, and thus does not change the trace-inverse scaling rule derived below.

We now prove the trace-inverse scaling rule. Assume there exist constants $\kappa_L, \kappa_R > 0$ such that[2]

$$\alpha_i^L = \kappa_L \,\mathrm{Tr}(\Sigma_i), \qquad \alpha_i^R = \kappa_R \,\mathrm{Tr}(\Sigma_i), \qquad \forall i \in \{1, \dots, E\}. \tag{28}$$

Substituting Eq. (28) into Eqs. (23) and (27), we obtain

$$\mathbb{E}\big[\|g_A^i\|_F^2\big] = \rho_w s_s^i \kappa_L \,\mathrm{Tr}(\Sigma_i), \tag{29}$$

$$\mathbb{E}\big[\|g_B^i\|_F^2\big] = \rho_w s_s^i \kappa_R \,\mathrm{Tr}(\Sigma_i). \tag{30}$$

Therefore, if we choose

$$s_s^i = s_{\mathrm{base}} \cdot \frac{C}{\mathrm{Tr}(\Sigma_i)}, \tag{31}$$

then it follows that

$$\mathbb{E}\big[\|g_A^i\|_F^2\big] = \rho_w s_{\mathrm{base}} C \kappa_L, \tag{32}$$

$$\mathbb{E}\big[\|g_B^i\|_F^2\big] = \rho_w s_{\mathrm{base}} C \kappa_R, \tag{33}$$

which are independent of i. Hence, the expected squared Frobenius norms of the localized gradients are equalized across all specialized experts. This proves the theorem. □

Proposition 2 (Isotropic global gradient variance implies projected uniformity)**.** *A sufficient condition for the projected-uniformity condition in Eq.* (28) *is that the global gradient second moments are isotropic, namely*

$$G_L = \mathbb{E}[gg^\top] = c_L I, \qquad G_R = \mathbb{E}[g^\top g] = c_R I, \tag{34}$$

for some constants $c_L, c_R > 0$. Under this assumption, for every specialized expert i,

$$\alpha_i^L = c_L \,\mathrm{Tr}(\Sigma_i), \qquad \alpha_i^R = c_R \,\mathrm{Tr}(\Sigma_i), \tag{35}$$

where

$$\alpha_i^L = \mathrm{Tr}\big(\Sigma_i U_i^\top G_L U_i\big)\,, \qquad \alpha_i^R = \mathrm{Tr}\big(\Sigma_i V_i^\top G_R V_i\big)\,. \tag{36}$$

Therefore, Eq. (28) *holds with $\kappa_L = c_L$ and $\kappa_R = c_R$.*

[2] A sufficient condition is $G_L \propto I$ and $G_R \propto I$. These are assumptions on uncentered second moments. Related isotropic covariance models, which are distinct from this condition, are discussed by Jastrzębski et al. (2017).

Proof. Since U_i and V_i are composed of orthogonal singular vectors, they satisfy $U_i^\top U_i = I$ and $V_i^\top V_i = I$. Substituting $G_L = c_L I$ into α_i^L gives

$$\alpha_i^L = \mathrm{Tr}\big(\Sigma_i U_i^\top (c_L I) U_i\big) = c_L \,\mathrm{Tr}\big(\Sigma_i U_i^\top U_i\big) = c_L \,\mathrm{Tr}(\Sigma_i). \tag{37}$$

Similarly, substituting $G_R = c_R I$ into α_i^R yields

$$\alpha_i^R = \mathrm{Tr}\big(\Sigma_i V_i^\top (c_R I) V_i\big) = c_R \,\mathrm{Tr}\big(\Sigma_i V_i^\top V_i\big) = c_R \,\mathrm{Tr}(\Sigma_i). \tag{38}$$

Hence the projected-uniformity condition is satisfied exactly with $\kappa_L = c_L$ and $\kappa_R = c_R$. This isotropic case is an analytically tractable sufficient condition, not a claim that the empirical moments are exactly isotropic. Related covariance-based SGD idealizations are discussed by Jastrzębski et al. (2017). □

C Proof of Expectation-Based Backbone Weight Adjustment

Proof. Recall that the effective weight of the proposed module is

$$W_{eq}(x) = \tilde{W}_0 + s_g B_g A_g + \sum_{i=1}^{E} w^i(x)\, s_s^i B_s^i A_s^i, \tag{39}$$

where $\tilde{W}_0$ is defined as

$$\tilde{W}_0 = W_0 - s_g B_g A_g - \frac{1}{E}\sum_{i=1}^{E} s_s^i B_s^i A_s^i. \tag{40}$$

Under the uniform-routing assumption at initialization, each specialized expert is selected with equal probability in expectation. Hence,

$$\mathbb{E}_x[w^i(x)] = \frac{1}{E}, \qquad i = 1, \ldots, E. \tag{41}$$

Therefore, taking expectation over the input-dependent routing gives

$$\begin{aligned}
\mathbb{E}_x[W_{eq}(x)] &= \tilde{W}_0 + s_g B_g A_g + \mathbb{E}_x\left[\sum_{i=1}^{E} w^i(x)\, s_s^i B_s^i A_s^i\right] \\
&= \tilde{W}_0 + s_g B_g A_g + \sum_{i=1}^{E} \mathbb{E}_x[w^i(x)]\, s_s^i B_s^i A_s^i \\
&= \tilde{W}_0 + s_g B_g A_g + \frac{1}{E}\sum_{i=1}^{E} s_s^i B_s^i A_s^i.
\end{aligned} \tag{42}$$

Substituting the definition of $\tilde{W}_0$ yields

$$\mathbb{E}_x[W_{eq}(x)] = W_0. \tag{43}$$

Thus, the effective weight $W_{eq}(x)$ is aligned with the original backbone weight W_0 in expectation at initialization. □

D More Experiment Results

D.1 More Qualitative Evaluation in Generalization Scenarios

As a comprehensive supplement to the qualitative evaluation presented in the main text, Figure 4 provides an extensive visualization of model attention maps across a wider range of out-of-distribution (OOD) generalization scenarios. Specifically, we evaluate the visual grounding robustness of all models under seven challenging axes of variation: Camera Viewpoints, Object Layouts, Robot Initial States, Light Conditions, Language Instructions, Sensor Noise, and Background Textures. All attention maps are taken from the last transformer layer of the models.

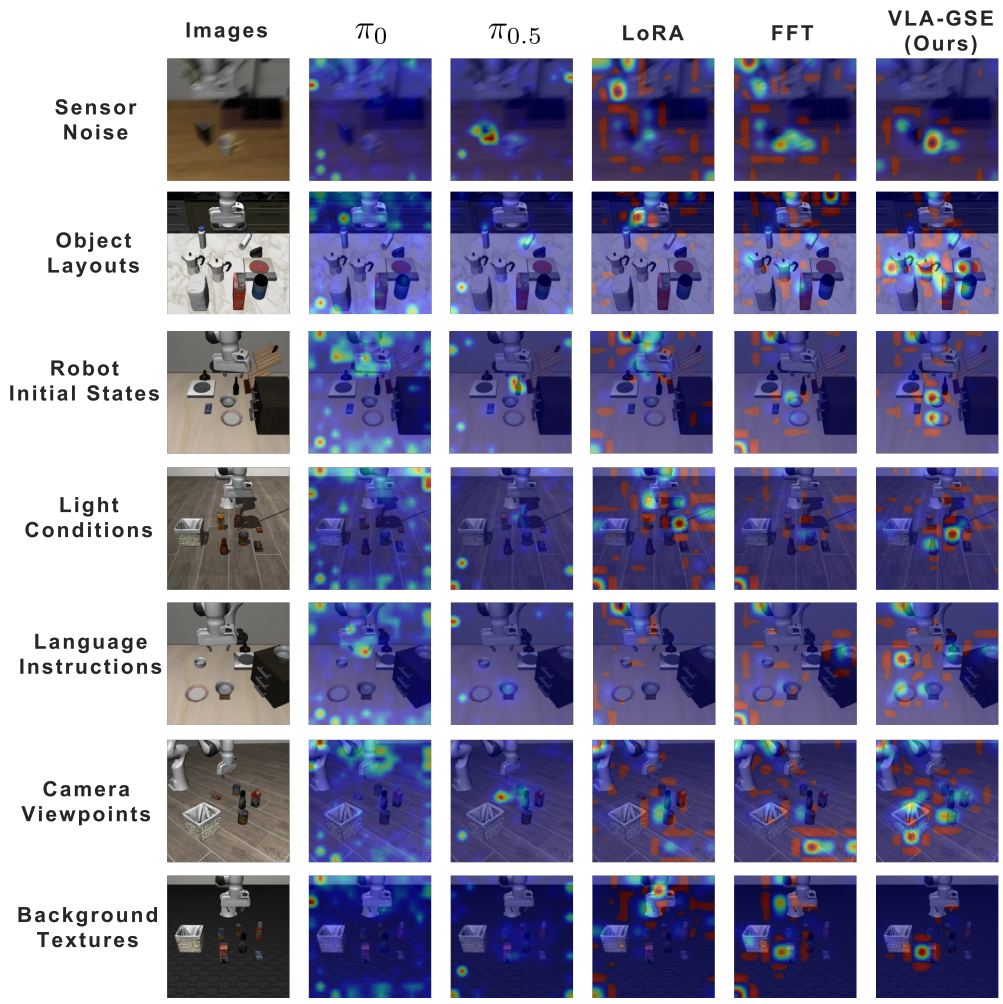

Figure 4: Extended attention-map examples under variations in Camera Viewpoints, Object Layouts, Robot Initial States, Light Conditions, Language Instructions, Sensor Noise, and Background Textures. All maps are taken from the last transformer layer. In these selected examples, GSE attention is generally more localized around target objects than that of the baselines (π_0, $\pi_{0.5}$, LoRA, and FFT). These qualitative visualizations do not by themselves measure task success or knowledge retention.

Table 6: LIBERO simulation benchmark results. Success rates (SR, %) across the four LIBERO task suites.

Method	Spatial	Object	Goal	Long	Average
Diffusion Policy (Chi et al., 2023)	78.3	92.5	68.3	50.5	72.4
Dita (Hou et al., 2025)	97.4	94.8	93.2	83.6	92.3
π_0 (Black et al., 2025)	96.8	98.8	95.8	85.2	94.2
$\pi_{0.5}$ (Physical Intelligence et al., 2025)	98.8	98.2	98.0	92.4	96.9
FLOWER (Reuss et al., 2025)	97.5	99.1	96.1	94.9	96.9
GR00T-N1.6 (NVIDIA GEAR Team, 2025)	97.7	98.5	97.5	94.4	97.0
OpenVLA-OFT (Kim et al., 2025a)	97.6	98.4	97.9	94.5	97.1
CogVLA (Li et al., 2025)	98.6	98.8	96.6	95.4	97.4
VLANeXt (Wu et al., 2026)	**99.0**	99.2	96.6	94.6	97.4
X-VLA (Zheng et al., 2025)	98.2	98.6	97.8	**97.6**	98.1
VLA-GSE (Ours)	98.8	**99.8**	**98.2**	96.8	**98.4**

In the cases shown in Figure 4, GSE focuses attention on relevant target objects and graspable edges under varied environmental conditions. These examples illustrate useful visual grounding, but they are not estimates of success over the full evaluation distribution and do not establish invariance to every perturbation. The aggregate success rates in Tables 1 and 2, together with the failure analysis in Appendix H.9, provide the complementary evidence needed to assess robustness, particularly under sensor noise.

The baselines show less localized or distracted attention in some illustrated cases. Such behavior can arise from imperfect grounding, task adaptation, or observation corruption, and does not by itself diagnose catastrophic forgetting. In particular, Table 3 shows that LoRA retains general multimodal performance comparably to GSE despite its weaker robotic performance in Table 2. We therefore interpret these attention maps as qualitative grounding evidence, rather than proof that all baselines have lost pretrained knowledge. They complement, rather than replace, the quantitative success rates and failure counts.

D.2 Comparison of VLA Models on LIBERO Results

Table 6 reports the simulation results on LIBERO. VLA-GSE achieves the best overall performance, reaching an average success rate of 98.4%. Even after including strong recent baselines such as FLOWER, GR00T-N1.6, VLANeXt, and X-VLA, our method still ranks first overall, outperforming the strongest baseline, X-VLA, by 0.3 points in average success rate. VLA-GSE also establishes new best results on Object (99.8%) and Goal (98.2%), while remaining highly competitive on Spatial (98.8%) and Long (96.8%).

These results suggest that GSE provides a strong balance between fine-grained task adaptation and robustness across heterogeneous task suites. In particular, compared with X-VLA, the gains on Object and Goal are +1.2 and +0.4 points, respectively, indicating that the proposed generalized-specialized expert decomposition improves task-specific adaptation without sacrificing cross-suite generalization. Although VLANeXt attains the best Spatial score and X-VLA achieves the best Long score, VLA-GSE delivers the highest average performance, demonstrating the strongest overall trade-off across the four LIBERO suites. Overall, the results show that parameter-efficient adaptation with generalized and specialized experts can remain competitive with, and often surpass, strong recent VLA baselines without requiring full-model fine-tuning.

D.3 Comparison of Fine-Tuning Methods on LIBERO Results

The frozen control keeps Qwen3-VL-4B-Instruct fixed and trains the same OFT action head. Its 4.7% average success shows that pretrained features plus a trained head alone are insufficient for these control tasks. This establishes the need for backbone adaptation, not the relative superiority of GSE; the latter is evaluated through the FFT, LoRA, and GOAT comparisons.

Table 7: Fine-tuning method comparison on the standard LIBERO benchmark. Results are reported as success rate (%) across the four task suites. All PEFT methods are configured with comparable trainable parameter budgets, following Table 2. The frozen control trains only the OFT action head; its unreported parameter fraction is marked by a dash.

Method	Params (%)	Spatial	Object	Goal	Long	Average
Frozen VLM + OFT	–	6.2	8.6	4.0	0.0	4.7
FFT (Full Fine-Tuning)	100.00	97.8	98.8	96.4	94.2	96.8
LoRA (Hu et al., 2022)	2.58	94.4	96.8	90.4	78.4	90.0
MiLoRA (Wang et al., 2025)	2.52	97.0	98.2	95.0	84.2	93.6
PiSSA (Meng et al., 2024)	2.54	96.0	97.4	93.6	89.8	94.2
rsLoRA (Kalajdzievski, 2023)	2.52	96.4	98.0	94.0	90.8	94.8
HydraLoRA (Tian et al., 2024)	2.55	96.6	98.0	94.2	91.2	95.0
DoRA (Liu et al., 2024)	2.54	96.6	98.2	94.4	91.6	95.2
AdaMoLE (Liu & Luo, 2024)	2.53	96.8	98.2	94.8	91.8	95.4
MoLoRA (Zadouri et al., 2024)	2.56	97.2	98.4	95.2	92.4	95.8
GOAT (Fan et al., 2025)	2.54	97.4	98.6	95.8	93.0	96.2
VLA-GSE (Ours)	2.51	**98.8**	**99.8**	**98.2**	**96.8**	**98.4**

Table 7 reports the results on the standard LIBERO benchmark. Overall, the ranking is broadly consistent with that on LIBERO-Plus: stronger parameter-efficient fine-tuning methods on zero-shot generalization also tend to achieve better performance on the in-distribution LIBERO benchmark. Vanilla low-rank adaptation, such as LoRA, already provides a competitive baseline, while more advanced variants, including DoRA, AdaMoLE, MoLoRA, and GOAT, further improve the average success rate, indicating that a richer adaptation space is beneficial even under the standard simulation setting.

Among all baselines, FFT remains highly competitive, achieving an average success rate of 96.8%. Nevertheless, VLA-GSE still delivers the strongest overall performance, reaching 98.4% average success rate and ranking first on all four suites. Compared with FFT, our method improves Spatial from 97.8% to 98.8%, Object from 98.8% to 99.8%, Goal from 96.4% to 98.2%, and Long from 94.2% to 96.8%, corresponding to gains of 1.0, 1.0, 1.8, and 2.6 points, respectively. The largest improvement is observed on the Long suite, suggesting that VLA-GSE is particularly effective for long-horizon manipulation requiring sustained temporal consistency and accurate multi-step action composition.

Compared with the strongest PEFT baseline, GOAT, VLA-GSE improves the average success rate from 96.2% to 98.4%, yielding a margin of 2.2 points. This advantage is again most pronounced on the more challenging Goal and Long suites, where our method outperforms GOAT by 2.4 and 3.8 points, respectively. Notably, Object is already close to saturation for most competitive methods, whereas Long still exhibits a clear performance spread. This pattern indicates that long-horizon sequential manipulation remains the main differentiating factor once short-horizon perception and control are well handled. Overall, these results provide additional evidence that explicitly combining generalized transferable adaptation with specialized experts yields a more effective fine-tuning mechanism for VLAs than both conventional LoRA-style methods and existing LoRA-MoE variants.

D.4 Real-World Inference Time Comparison

We further compare the real-world inference efficiency of different policies by measuring the wall-clock latency required to generate one action chunk. All methods are evaluated with batch size 1 under the same deployment pipeline on the same real-robot workstation equipped with an NVIDIA RTX 5080 GPU. We measure the elapsed time from receiving the current observation to returning the full action chunk, and report the average latency over 120 runs.

As shown in Figure 5, Qwen3-VL-FFT achieves the lowest latency, while VLA-GSE is the second fastest method among the compared policies. The most controlled comparison is between VLA-GSE and Qwen3-VL-FFT, since both are built on the same Qwen3-VL-4B backbone and mainly differ

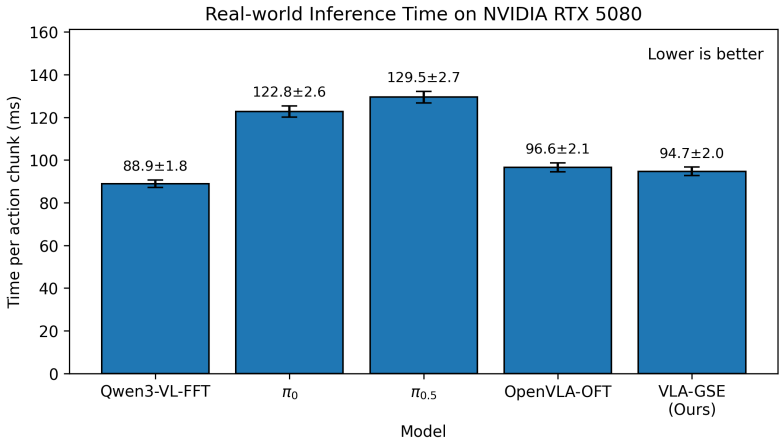

Figure 5: Real-world inference latency comparison on an NVIDIA RTX 5080 GPU. We report the wall-clock time for generating one action chunk with batch size 1. Error bars denote standard deviation over 120 runs. Lower is better.

in the adaptation strategy. Under this matched setting, the latency gap indicates that the proposed generalized-specialized expert design introduces only a modest additional inference overhead.

We note, however, that absolute latency comparisons across different policy families should be interpreted with caution. In particular, the lower latency of VLA-GSE relative to π_0 and $\pi_{0.5}$ is not solely due to the proposed adapter design, but also reflects differences in the action-generation paradigm: VLA-GSE adopts OFT-style parallel decoding, whereas π_0 and $\pi_{0.5}$ rely on flow-matching-based generation, which is inherently more expensive at inference time. Similarly, the lower latency of VLA-GSE relative to OpenVLA-OFT should not be over-interpreted as a pure architectural advantage of GSE, since OpenVLA-OFT is built on a larger 7B backbone, whereas VLA-GSE uses a 4B backbone. Therefore, these cross-model comparisons are best viewed as end-to-end deployment measurements rather than clean isolation of module-level efficiency.

Overall, the main takeaway from Figure 5 is that, relative to the matched Qwen3-VL-FFT baseline, VLA-GSE improves policy performance while incurring only a small additional inference-time cost, while remaining practically efficient for real-world deployment.

D.5 Suite-wise Results on LIBERO-Plus

For compact comparison, tables in the main paper reports only the averaged LIBERO-Plus performance of VLA-GSE. Here, we provide its detailed zero-shot results on the four LIBERO suites: *Spatial*, *Object*, *Goal*, and *Long*.

As shown in Table 8, VLA-GSE achieves strong performance across all four suites, with the best overall results on *Spatial* (90.3) and competitive performance on the more challenging *Goal* and *Long* suites. Notably, the suite-wise breakdown shows that the strong averaged performance in the main paper is not dominated by a single suite, but is supported by consistently high robustness across diverse distribution shifts, including camera, robot embodiment, language, lighting, background, noise, and layout variations. These results further demonstrate that VLA-GSE generalizes well across different task structures and evaluation conditions in LIBERO-Plus.

D.6 Training Details, Hyperparameters, and Task Description

Training Details and Hyperparameters Table 9 details the comprehensive set of hyperparameters and training configurations used for fine-tuning the Qwen3-VL-4B-Instruct VLM with our proposed VLA-GSE framework.

Real-World Task Description and Hyperparameters For the real-world experiments, we evaluate four representative long-horizon manipulation tasks: *pour water into bowl*, which requires stable

Table 8: Suite-wise zero-shot performance of VLA-GSE on LIBERO-Plus. Results are shown in success rate (%).

Suite	Camera	Robot	Language	Light	Background	Noise	Layout	Total
Spatial	89.5	80.6	89.6	94.8	96.4	89.2	95.2	90.3
Object	80.5	65.3	95.1	100.0	98.4	86.6	85.1	86.2
Goal	51.7	67.2	79.3	99.6	96.1	76.5	62.8	74.2
Long	35.8	61.0	91.0	94.9	98.3	65.2	87.3	74.1
Average	64.4	68.5	88.8	97.3	97.3	79.4	82.6	81.2

container grasping and controlled pouring; *pick spoon into plate*, which involves precise grasping and object placement; *put carrot into bowl*, which tests visually grounded pick-and-place under object-level spatial variation; and *take pen out of pen container*, which requires extracting a slender pen from a cylindrical container. For each task, we collect 70 in-distribution demonstrations via teleoperation. Each task contains 60 evaluation episodes, covering four out-of-distribution (OOD) generalization settings: variations in *lighting*, which change scene illumination conditions; *language*, which rephrase the task instruction; *clutter layout*, which modify the arrangement of surrounding objects; and *background tablecloth*, which alter the background appearance of the workspace. These settings are designed to evaluate the robustness of the learned policy under realistic visual and linguistic distribution shifts.

During training, we set the learning rate for the VLM and GSE parameters to 1×10^{-5}, and the learning rate for the action MLP head to 1×10^{-4}. The policy for *pour water into bowl* is trained for 8000 iterations, whereas the policies for the other three tasks are each trained for 6000 iterations. Other real-world task training details and hyperparameters are consistent with Table 9.

E VLA-GSE Algorithm Pseudo-code

In this section, we provide the complete pseudo-code for the VLA-GSE framework. Algorithm 1 explicitly outlines the two primary operational phases of our approach. The first phase is the SVD-based adaptive initialization, where the pre-trained weight matrix is decomposed. The dominant singular values are allocated to the generalized expert to anchor foundational knowledge, while the subsequent singular components are partitioned among the specialized experts. A crucial element of this phase is the backbone weight adjustment scheme via residual subtraction, which preemptively deducts the expected initial contribution of the newly added experts from the pre-trained backbone. The second phase details the forward pass, which aggregates the adjusted frozen weights, the continuously active generalized expert, and the dynamically routed top-k specialized experts.

F Catastrophic Forgetting under VLM-to-VLA Full-Parameter Fine-Tuning

A growing body of recent work has pointed out that directly adapting a pretrained VLM into a VLA by full-parameter fine-tuning can easily induce *catastrophic forgetting*. Although the VLM-to-VLA paradigm is attractive because it transfers rich visual-semantic priors from internet-scale pretraining to embodied control, the optimization objective in robot learning is substantially narrower than that of the original VLM. As a result, the backbone can over-specialize to low-level action prediction and partially lose the open-world understanding ability that originally motivated using a VLM backbone in the first place.

Several recent studies provide direct evidence for this phenomenon. Hancock et al. (2025) argue that catastrophic forgetting in VLM-to-VLA transfer is fundamentally related to the distribution mismatch between web-scale VLM pretraining data and robotics fine-tuning data. They show that learning to generate actions often diminishes the VLM's original reasoning and multimodal understanding ability, which in turn harms semantic generalization and instruction following. Their solution is primarily *data-centric*: by re-expressing actions as language, they reduce representational mismatch and enable LoRA-based adaptation without heavily perturbing the pretrained backbone.

Table 9: Hyperparameters and Configuration for VLA-GSE Fine-tuning

Hyperparameter / Setting	**Value**
Architecture & VLA-GSE Configuration	
Base VLM Backbone	Qwen3-VL-4B-Instruct
Action Head	OpenVLA-OFT MLP (Kim et al., 2025a)
Rank (r)	16
Rank of Generalized Expert (r_g)	2
Rank of Specialized Expert (d)	2
Total Experts per GSE Block	8
Generalized Experts	1
Top-k Routing	2
Router Initialization	Random Gaussian Noise
Initialization Type	SVD-based GSE
Total Generalized Experts in Model	356
Total Specialized Experts in Model	2492
Training & Optimization	
Training Dataset	LIBERO Benchmark (all 4 suites)
Total Optimization Steps	80,000
Batch Size per GPU	16
Total Effective Batch Size	128
Learning Rate (VLM / GSE Params)	1×10^{-5}
Learning Rate (Action MLP Head)	1×10^{-4}
Auxiliary Loss Weight (α)	0.01
Generalized Scaling Factor (s_g)	2
Specialized Scaling Factor (s_s^i)	$\frac{2 \cdot \sum_{j=1}^{E} \mathrm{Tr}(\Sigma_j)}{E \cdot \mathrm{Tr}(\Sigma_i)}$
Hardware	8 $\times$ NVIDIA A100 GPUs
Parameter Efficiency	
Total Parameters	4,551.85 M
Frozen Parameters (Non-GSE)	4,437.82 M
Trainable Parameters (Total)	114.04 M (2.51%)
Trainable Parameters (GSE Modules)	48.41 M
Trainable Parameters (Action Head)	65.62 M

Dey et al. (2024) provide complementary evidence from the vision backbone perspective. Focusing on OpenVLA, they show that fine-tuning the full robotic model can lead to severe forgetting in the visual encoders, especially DINO-v2. In particular, they demonstrate that the DINO-v2 encoder inside OpenVLA fails on depth regression after VLA training, in sharp contrast to the original pretrained backbone. Their findings suggest that end-to-end robotic fine-tuning may overwrite core spatial representations required for visual generalization. Their remedy is *post hoc restoration*: they partially revert the visual encoders back to their pretrained states through backbone reversal and model merging.

Yang et al. (2025) further observe that existing VLA models often suffer catastrophic forgetting of pretrained vision-language capabilities when adapted to manipulation. Their analysis shows that directly fine-tuning OpenVLA on instruction-rich embodied data yields limited gains on reasoning-intensive tasks because the original vision-language competence is not well preserved. To address this, they propose a *training-centric* solution based on joint multimodal instruction tuning, where embodied data and general multimodal data are optimized together.

Most strikingly, Yu et al. (2026) provide explicit empirical analysis of the severity of forgetting during VLA training. They report that standard robot-only VLA fine-tuning can cause a near-complete collapse of general visual understanding; for example, the POPE score of Qwen3-VL drops from 88.87% to 0.04% after fine-tuning. They also show that simple co-training with general visual QA data only partially alleviates the issue. Their proposed solution is therefore *architectural decoupling*:

Algorithm 1 GSE: Generalized and Specialized Experts Initialization and Forward Pass

Require: Input x, pre-trained weight W_0, rank r_g for the generalized expert, rank d for specialized experts, number of specialized experts E, generalized scaling factor s_g, and specialized scaling factors $\{s_s^i\}_{i=1}^{E}$

Initialization Phase:
Compute SVD: $W_0 = U\Sigma V^\top$
▷ 1. Initialize generalized expert (largest singular values)
$B_g \leftarrow \sqrt{\frac{1}{s_g}} U_{1:r_g} \Sigma_{1:r_g}^{1/2}$
$A_g \leftarrow \sqrt{\frac{1}{s_g}} \Sigma_{1:r_g}^{1/2} (V_{1:r_g})^\top$
▷ 2. Initialize specialized experts (subsequent singular values)
for $i = 1$ to E **do**
    Extract the i-th SVD segment $(U_i, \Sigma_i, V_i^\top)$ of rank d, starting after index r_g
    $B_s^i \leftarrow \sqrt{\frac{1}{s_s^i}} U_i \Sigma_i^{1/2}$
    $A_s^i \leftarrow \sqrt{\frac{1}{s_s^i}} \Sigma_i^{1/2} V_i^\top$
end for
▷ 3. Backbone weight adjustment via residual subtraction
$W_{\text{res}} \leftarrow s_g B_g A_g + \frac{1}{E} \sum_{i=1}^{E} s_s^i B_s^i A_s^i$
$\tilde{W}_0 \leftarrow W_0 - W_{\text{res}}$
return $\tilde{W}_0, B_g, A_g, \{B_s^i, A_s^i\}_{i=1}^{E}$

Forward Pass Phase(x)**:**
Compute gating scores $w^i(x)$ for specialized experts
Select top-k specialized experts $\Omega_k(x)$
$y_{\text{gen}} \leftarrow s_g B_g A_g x$
$y_{\text{spec}} \leftarrow \sum_{i \in \Omega_k(x)} w^i(x)\, s_s^i\, B_s^i A_s^i x$
return $\tilde{W}_0 x + y_{\text{gen}} + y_{\text{spec}}$

a frozen "Left Brain" preserves general semantic understanding, while a trainable "Right Brain" specializes in embodied control.

Our proposed VLA-GSE differs from the above lines of work in an important way. Rather than addressing forgetting primarily through data relabeling, co-training, post-training reversal, or duplicating the backbone into separate frozen/trainable branches, we approach the problem from the perspective of *architectural improvement within the adaptation module itself*. Concretely, VLA-GSE is built on the hypothesis that the tension between general visual-semantic knowledge and task-specific embodied adaptation should be handled by a better parameterization of the trainable update, rather than by fully rewriting the backbone. The generalized expert is designed to preserve and reuse dominant pretrained directions that support broad transferable capability, while the specialized experts allocate dedicated capacity for task-dependent embodied adaptation. This design aims to improve specialization without forcing the backbone update to collapse into a single monolithic fine-tuning trajectory.

Therefore, compared with prior methods, VLA-GSE is not merely a forgetting *mitigation strategy* added on top of standard adaptation; it is an architectural reformulation of the adaptation process itself. In this sense, our method is conceptually closer to improving the *structure of transfer* than to modifying the training data distribution or repairing forgetting after it has already occurred. We believe this distinction is important: it suggests that preserving general VLM capability and acquiring embodied expertise need not be treated as a purely training-data trade-off, but can instead be addressed through a more suitable model architecture for VLM-to-VLA transfer.

G Detailed Explanation of the Load Balancing Auxiliary Loss

Each GSE block adopts sparse routing over the specialized experts, which enables input-dependent adaptation while preserving computational efficiency. However, sparse top-k routing is known to suffer from a degenerate optimization behavior in which only a small subset of experts is selected for most tokens, whereas the remaining experts receive little gradient signal. This *routing collapse* effect reduces the effective capacity of the expert pool and weakens the intended specialization mechanism. To mitigate this issue, we introduce an auxiliary load-balancing objective following prior sparse MoE literature (Dai et al., 2024; Fedus et al., 2022).

We emphasize that this regularizer is imposed only on the routed specialized experts. The *Generalized Expert* is always activated and therefore does not participate in the routing competition.

For the l-th GSE block, consider a batch of T tokens with token representations $\{x^{(t)}\}_{t=1}^{T}$. Let $\Omega_k(x^{(t)})$ denote the set of top-k selected specialized experts for token $x^{(t)}$, and let $p^i(x^{(t)})$ denote the corresponding soft routing probability of expert i. We define the empirical selection frequency of expert i as

$$f_i^{(l)} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{I}\big(i \in \Omega_k(x^{(t)})\big), \tag{44}$$

which measures how often expert i is actually activated under hard routing. In parallel, we define the average routing mass of expert i as

$$P_i^{(l)} = \frac{1}{T}\sum_{t=1}^{T} p^i(x^{(t)}), \tag{45}$$

which reflects the router's continuous preference for that expert before the top-k truncation.

The auxiliary balancing term for block l is then given by

$$\mathcal{L}_{bal}^{(l)} = E\sum_{i=1}^{E} f_i^{(l)} P_i^{(l)}, \tag{46}$$

where E is the number of specialized experts. This form couples the hard assignment statistics $f_i^{(l)}$ with the soft routing scores $P_i^{(l)}$, and therefore penalizes imbalance at both the discrete and probabilistic levels. In particular, if the router collapses onto a small number of experts, then both $f_i^{(l)}$ and $P_i^{(l)}$ concentrate on those experts, leading to a larger penalty. Conversely, when expert usage is more evenly distributed, both quantities become closer to uniform and the regularizer is correspondingly reduced.

The overall training objective is

$$\mathcal{L} = \|\hat{a} - a\|_1 + \alpha\sum_{l=1}^{L}\mathcal{L}_{bal}^{(l)}, \tag{47}$$

where $\|\hat{a} - a\|_1$ denotes the action regression loss and α controls the strength of the auxiliary regularization.

From an optimization perspective, this objective encourages agreement between the router's soft probability mass and its hard top-k allocation statistics while disincentivizing concentration on a small subset of experts. Under balanced routing, one expects both $f_i^{(l)}$ and $P_i^{(l)}$ to approach a near-uniform distribution across experts, in which case the loss is minimized up to the top-k sparsity constraint. Consequently, the regularizer improves training stability, prevents persistent expert under-utilization, and promotes more effective specialization across the expert pool. Empirically, this effect is important for maintaining the intended division of labor among specialized experts and for fully exploiting the parameter efficiency of the GSE design.

H Additional Controlled Experiments and Diagnostics

This section supplements the main comparisons with measured training costs, training-seed uncertainty, frozen-backbone controls, cross-backbone and action evaluations, optimization diagnostics,

and real-world failure analysis. We distinguish independently trained policies from fixed-checkpoint diagnostic replays and distinguish training-seed uncertainty from uncertainty conditional on a fixed set of evaluation tasks.

H.1 Matched Training Cost and Model Selection

For the comparison in Table 2, all methods use Qwen3-VL-4B-Instruct, the OFT action head, mixed precision, the same dataloader, a total batch size of 128, and eight NVIDIA A100 GPUs. The target training budget is approximately 41.2 hours. Table 10 reports the measured per-iteration cost and corresponding schedules. Thus, wall-clock time and hardware occupancy are approximately matched, whereas training steps, active parameters, and FLOPs are not. The PEFT methods have approximately matched trainable-parameter budgets; FFT updates all parameters. In particular, the reported FFT score of 74.9% corresponds to its 77,000-step, 41.1-hour trajectory, rather than an 80,000-step run.

Table 10: Measured training cost for the LIBERO-Plus fine-tuning comparison on eight A100 GPUs. SVD initialization is a separate, one-time cost and is not included in the training wall-clock or GPU-hour columns.

Method	s/iteration	Steps	Wall h	GPU-h	SVD (min)
FFT	1.920	77,000	41.1	328.5	–
LoRA	1.610	92,000	41.1	329.2	–
rsLoRA	1.620	92,000	41.4	331.2	–
DoRA	1.710	87,000	41.3	330.6	–
PiSSA	1.630	91,000	41.2	329.6	10.2
MoLoRA	1.800	82,000	41.0	328.0	–
AdaMoLE	1.840	81,000	41.4	331.2	–
HydraLoRA	1.780	83,000	41.0	328.3	–
MiLoRA	1.650	90,000	41.3	330.0	7.1
GOAT	1.830	81,000	41.2	329.4	10.2
VLA-GSE	1.854	80,000	41.2	329.6	10.2

VLA-GSE is approximately 1.3% slower per iteration than GOAT. Its SVD initialization takes 10.2 minutes on one A100, or 0.17 GPU-hours, approximately 0.05% of its training GPU-hours. These measurements quantify the implemented training pipelines, not an active-FLOP-matched comparison.

The baseline implementations retain their defining adaptation rules. The reported baseline learning-rate search grids are $\{5 \times 10^{-6}, 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}\}$ for FFT and $\{5 \times 10^{-5}, 10^{-4}, 2 \times 10^{-4}, 5 \times 10^{-4}\}$ for the PEFT baselines, with other method-specific ranges following the original implementations. The selected, module-specific VLA-GSE learning rates remain those in Table 9. Checkpoints are selected using held-out validation action loss, not test success.

H.2 Uncertainty across Training Seeds

We repeat FFT, GOAT, and VLA-GSE using training seeds 42, 123, and 456 under the same backbone, training data, OFT head, budget, and LIBERO-Plus evaluation protocol. Table 11 reports the individual outcomes, sample standard deviations, and seed-level confidence intervals. The seed-42 column contains the point estimates used for these three methods in Table 2.

Table 11: LIBERO-Plus overall success rate (%) across three independent training seeds. SD is the sample standard deviation across training seeds. The intervals describe training-seed uncertainty, not episode-level uncertainty.

Method	Seed 42	Seed 123	Seed 456	Mean $\pm$ SD	95% CI
FFT	74.9	73.7	75.8	74.8 ± 1.1	[72.2, 77.4]
GOAT	76.8	77.4	75.9	76.7 ± 0.8	[74.8, 78.6]
VLA-GSE	81.2	80.2	81.9	81.1 ± 0.9	[79.0, 83.2]

The intervals use $\bar{x} \pm t_{0.975,2} s/\sqrt{3}$, with $t_{0.975,2} \approx 4.303$. VLA-GSE's mean exceeds GOAT and FFT by 4.4 and 6.3 percentage points, respectively. With only three training seeds, these are preliminary robustness estimates. They are our primary training-seed uncertainty evidence and do not establish statistical superiority over ABot-M0, for which repeated training runs are unavailable.

The comparison with ABot-M0 in Table 1 is across systems with different robot-pretraining histories. ABot-M0 (Yang et al., 2026) uses UniACT pretraining with more than six million trajectories and 9,500 hours of data, whereas our configuration has no additional VLA/robot-pretraining stage and adapts its VLM backbone on LIBERO. The 81.2% versus 80.5% point comparison should not be interpreted as a matched-pretraining or statistically established advantage.

H.3 Backbone Attribution on a Different Benchmark

The frozen-backbone control on standard LIBERO is included in Table 7. We additionally evaluate SimplerEnv WidowX Visual Matching (Li et al., 2024c), using BridgeData V2 (Walke et al., 2024). This changes the training distribution, embodiment, scenes, objects, and real-to-simulation evaluation setup relative to LIBERO-Plus. We compare Qwen3-VL-4B-Instruct with PaliGemma 2-3B (Steiner et al., 2024), a smaller non-Qwen backbone based on SigLIP and Gemma 2.

Within each backbone, the adapted methods share training data, the OFT head, wall-clock budget, evaluator, and checkpoint-selection protocol. Method-specific hyperparameters are tuned independently, and PEFT parameter budgets are approximately matched. Frozen + OFT retains the pretrained VLM and trains only the action head. Each method is evaluated on 24 matched episodes for each of four tasks, giving 96 episodes per backbone and method. These evaluations use one policy-training seed per method and backbone.

Table 12: Success rates (%) on SimplerEnv WidowX Visual Matching. The frozen controls have only aggregate results; their unreported taskwise breakdowns are marked by dashes. Averages are computed from integer episode counts before rounding.

Method	Spoon	Carrot	Stack	Eggplant	Average
Qwen3-VL-4B					
Frozen + OFT	–	–	–	–	7.3
FFT	91.7	37.5	20.8	100.0	62.5
LoRA	79.2	45.8	12.5	87.5	56.3
GOAT	87.5	54.2	29.2	100.0	67.7
VLA-GSE	95.8	62.5	37.5	100.0	74.0
PaliGemma 2-3B					
Frozen + OFT	–	–	–	–	2.1
FFT	70.8	41.7	8.3	83.3	51.0
LoRA	58.3	45.8	4.2	70.8	44.8
GOAT	70.8	54.2	20.8	79.2	56.3
VLA-GSE	79.2	58.3	25.0	87.5	62.5

The frozen controls achieve 7/96 and 2/96 successes for Qwen3-VL and PaliGemma, respectively. They establish the need to adapt the backbone in these settings, rather than demonstrating GSE's relative effectiveness by themselves. Relative effectiveness is assessed using the adapted-method comparisons: Qwen3-VL yields 71/96 successes for GSE, 65/96 for GOAT, and 60/96 for FFT; PaliGemma yields 60/96, 54/96, and 49/96, respectively.

Confidence intervals in Table 13 use a task-stratified paired percentile bootstrap: 24 episode pairs are resampled within each task, the four tasks receive equal weight, and 100,000 resamples are generated with seed 42. Differences are calculated from integer counts before rounding. The six additional successes over GOAT correspond to $6/96 = 6.25$ percentage points on either backbone, and both intervals include zero. The intervals condition on the four evaluated tasks and the trained policies; they do not include training-seed uncertainty. These results support observed transfer across two backbone/scale configurations, not universal or statistically established superiority over GOAT.

Table 13: Paired SimplerEnv comparisons. Counts are ordered as both succeed / GSE only / comparator only / both fail. Δ and confidence intervals are in percentage points. The two-sided exact McNemar p-values are unadjusted for multiple comparisons.

Backbone	Comparator	Paired counts	Δ	p	95% CI
Qwen3-VL-4B	GOAT	58/13/7/18	6.3	0.263	[-3.1, 15.6]
PaliGemma 2-3B	GOAT	47/13/7/29	6.3	0.263	[-3.1, 15.6]
Qwen3-VL-4B	FFT	54/17/6/19	11.5	0.0347	[2.1, 20.8]
PaliGemma 2-3B	FFT	42/18/7/29	11.5	0.0433	[2.1, 21.9]

H.4 Action Representations and Decoders

Action coordinates. On LIBERO-Plus Long, we compare Delta EEF actions, which specify local pose increments, with Absolute EEF actions, which specify global pose targets. Both use the same seven-dimensional action space, with unchanged gripper commands. For Absolute EEF, demonstration trajectories are converted to absolute targets and the evaluator uses absolute-control mode. Qwen3-VL-4B, the OFT decoder, action horizon, data, training budget, and evaluation interface are matched within each FFT–GSE comparison.

Table 14: Action-representation comparison on LIBERO-Plus Long, in success rate (%). Totals follow the benchmark weighting used in Table 4, rather than an unweighted average of the displayed categories.

	Delta EEF		Absolute EEF	
Perturbation	FFT	VLA-GSE	FFT	VLA-GSE
Camera	30.6	35.8	18.6	24.7
Robot	53.8	61.0	38.9	45.6
Language	86.4	91.0	80.6	85.4
Light	92.2	94.9	87.3	90.8
Background	95.7	98.3	90.7	94.1
Noise	57.6	65.2	46.8	53.7
Layout	78.4	87.3	64.2	72.6
Total	68.3	74.1	58.2	64.1

The observed GSE-over-FFT differences are 5.8 percentage points for Delta EEF and 5.9 for Absolute EEF. These point estimates support compatibility with both tested representations; repeated-seed, paired evaluations are needed to characterize representation-level training uncertainty.

Action decoder. Keeping Qwen3-VL-4B and GSE fixed, we replace the OFT MLP regression head with a $\pi_{0.5}$-style cross-DiT flow-matching decoder (Physical Intelligence et al., 2025). OFT regresses action chunks with an L1 objective, whereas the flow decoder iteratively denoises continuous actions. Both use the same LIBERO-Plus Long data and benchmark aggregation.

Flow matching gives higher Camera and Noise point estimates, whereas OFT gives a higher overall result, 74.1% versus 72.3%. This establishes compatibility with both tested decoders. Without FFT + flow and repeated training seeds, it does not establish persistence of the GSE-over-FFT advantage across decoders.

H.5 Empirical Audit of Initialization Assumptions

We audit the assumptions underlying Theorem 1 using $\rho_i = \mathbb{E}_x[(w^i(x))^2]$ and $\beta_i^{L/R} = \alpha_i^{L/R}/\mathrm{Tr}(\Sigma_i)$. For each dataset and joint initialization/data seed, 32 samples are used to estimate these quantities. Each coefficient of variation (CV) and correlation is computed over the seven specialists within a layer and then averaged over 356 layers. Table 16 reports the mean and sample

Table 15: Decoder comparison with VLA-GSE on LIBERO-Plus Long, in success rate (%). The backbone and adaptation method are fixed; an FFT + flow control is not included.

Perturbation	OFT regression	$\pi_{0.5}$-style flow
Camera	35.8	37.1
Robot	61.0	58.6
Language	91.0	87.8
Light	94.9	92.7
Background	98.3	96.4
Noise	65.2	66.1
Layout	87.3	81.9
Total	74.1	72.3

SD across three such seeds. Projected moments are obtained directly from layer inputs and output gradients, rather than by inverting the theorem.

Table 16: Routing second-moment and projected-gradient diagnostics: mean $\pm$ sample SD across three joint initialization/data seeds. Entropy is the reported statistic for the normalized routing second-moment vector. The lower panel gives Pearson correlations with the assigned singular-value trace.

Dataset	$\mathrm{CV}(\rho)$	Entropy	$\mathrm{CV}(\beta^L)$	$\mathrm{CV}(\beta^R)$
Spatial	0.701 ± 0.006	0.859 ± 0.002	0.390 ± 0.017	0.647 ± 0.011
Object	0.699 ± 0.008	0.861 ± 0.002	0.370 ± 0.023	0.641 ± 0.007
Goal	0.705 ± 0.007	0.858 ± 0.002	0.374 ± 0.016	0.640 ± 0.005
Long	0.694 ± 0.010	0.862 ± 0.003	0.378 ± 0.038	0.645 ± 0.007
BridgeData	0.691 ± 0.004	0.862 ± 0.001	0.422 ± 0.023	0.645 ± 0.004

Dataset	$r(\alpha^L, \mathrm{Tr}\Sigma)$	$r(\alpha^R, \mathrm{Tr}\Sigma)$
Spatial	0.393 ± 0.009	0.515 ± 0.015
Object	0.415 ± 0.011	0.526 ± 0.009
Goal	0.406 ± 0.012	0.522 ± 0.004
Long	0.401 ± 0.063	0.525 ± 0.004
BridgeData	0.384 ± 0.026	0.515 ± 0.005

The dataset means span 0.691–0.705 for $\mathrm{CV}(\rho)$, 0.370–0.422 for $\mathrm{CV}(\beta^L)$, and 0.640–0.647 for $\mathrm{CV}(\beta^R)$. These finite-sample estimates show unequal routing second moments and only approximate proportionality between projected gradient moments and spectral trace. The theorem's conditions are analytical idealizations, not exact empirical equalities established by this audit.

H.6 Gradient-Scale Diagnostics beyond Initialization

To test the optimization consequence separately from the assumptions, we perform counterfactual checkpoint replays. Let $u_i^{(\ell)}$ be specialist i's effective one-step update norm in GSE layer ℓ. We first average its norm across all $L = 356$ layers and then compute CV across the seven specialists:

$$\widetilde{u}_i = \frac{1}{L}\sum_{\ell=1}^{L} u_i^{(\ell)}, \qquad \mathrm{CV}_{\mathrm{update}} = \frac{\mathrm{Std}_{i=1,\dots,7}(\widetilde{u}_i)}{\mathrm{Mean}_{i=1,\dots,7}(\widetilde{u}_i)}. \tag{48}$$

For each frozen checkpoint, the identical minibatch is replayed while changing only specialist scales. Proposed uses the block-matched trace-inverse scales, Uniform uses their mean, and Shuffled reassigns the same seven values. The early checkpoints at steps 0, 2k, and 5k use a controlled diagnostic batch; steps 20k–80k use fixed minibatches from the actual VLA pipeline. All three conditions share the batch within each checkpoint.

Table 17: Aggregate specialist update-CV in fixed-checkpoint, identical-batch replays; lower is more balanced. Each checkpoint is a single-batch diagnostic, not an independently trained scaling-control trajectory.

Checkpoint	Proposed	Uniform	Shuffled
0	0.175	0.329	0.394
2k	0.231	0.418	0.506
5k	0.132	0.240	0.351
20k	0.164	0.302	0.430
40k	0.147	0.281	0.398
60k	0.166	0.326	0.451
80k	0.159	0.294	0.467
Mean, 20k–80k	0.159	0.301	0.437

Proposed has the lowest CV at every audited replay. Across steps 20k–80k, Uniform and Shuffled have 1.89 and 2.75 times its mean CV, respectively. This supports an aggregate update-balancing effect on the tested batches despite inexact initialization assumptions. It does not demonstrate uniformity within every layer, robustness over multiple replay batches, or the behavior of three independently trained scaling variants.

The Uniform replay control also differs from the constant-2 scaling used in the training ablation in Table 4. That ablation reduces Long success from 74.1% to 72.1%, indicating a modest downstream benefit. The larger drops after removing SVD initialization or either expert type support their utility, but those structural ablations do not hold active paths or FLOPs constant. Scaling should therefore be interpreted as an optimization-balancing component, not the sole explanation of the total performance gain.

H.7 Spectral Ordering and Routing Symmetry

We compare three assignments of the same residual spectral blocks: Descending, a per-layer random permutation, and Reversed. For each mapping, policies are independently trained using seeds 42, 123, and 456. The router receives no explicit spectral-energy or rank metadata. At each audited checkpoint, all 356×7 routing/energy entries are finite.

For specialist i, let $\mathcal{E}_i = \sum_{j \in \mathcal{I}_i} \sigma_j^2$ be the assigned spectral energy and f_i its hard top-2 selection fraction. We compute the Spearman correlation $r_s(\mathcal{E}, f)$ within each layer and then average across layers and training seeds.

Table 18: Mean per-layer Spearman correlation between assigned spectral energy and hard routing frequency. Window ends are given in training steps; each mapping uses three training seeds.

Window end	Descending	Random/layer	Reversed
1k	0.025	0.004	0.003
2k	0.017	-0.011	0.014
3k	0.028	0.006	-0.009

All nine absolute correlations are below 0.03. Because energy decreases with expert index in Descending, its mean index–frequency correlations are -0.025, -0.017, and -0.028 at 1k, 2k, and 3k. At 1k, the reported pooled hard-routing entropy is 0.9998, and a joint permutation of specialists and corresponding router indices preserves the tested outputs within 10^{-6}.

These observations do not show a systematic decrease in activation with expert index in the audited windows. However, weak energy–frequency correlation and high pooled hard-routing entropy do not establish equal per-layer routing second moments; the distinct second-moment statistics are reported in Table 16.

H.8 Additional Real-World Baseline and Paired Evaluation

We add GOAT (Fan et al., 2025) to the same-backbone real-world comparison. All four methods in Table 19 use Qwen3-VL-4B and the OFT head, common data and preprocessing, independently tuned hyperparameters, one training seed, and paired initialization IDs. The four tasks each contain 60 evaluation episodes under the settings in Appendix D.6, giving 240 paired episodes per comparison.

Table 19: Same-backbone real-world comparison, in success rate (%). Water, Spoon, Carrot, and Pen denote the four tasks in Table 5. Averages are calculated from episode counts before rounding.

Method	Water	Spoon	Carrot	Pen	Average
LoRA	58.3	61.7	66.7	60.0	61.7
FFT	63.3	66.7	70.0	63.3	65.8
GOAT	66.7	75.0	80.0	71.7	73.3
VLA-GSE	81.7	85.0	83.3	80.0	82.5

GOAT and GSE achieve 176/240 and 198/240 successes, respectively. Their paired discordances are 32 GSE-only successes and 10 GOAT-only successes. Table 20 reports the paired comparisons against GOAT and FFT. Confidence intervals use a task-stratified paired bootstrap with 60 episode pairs per task and 100,000 resamples, using seed 42.

Table 20: Paired real-world differences. Δ and intervals are in percentage points; p is the unadjusted two-sided exact McNemar value. These intervals condition on the four tasks and the trained policies and exclude training-seed uncertainty.

Comparison	Δ	p	95% CI
GSE versus GOAT	9.2	9.41×10^{-4}	[4.2, 14.6]
GSE versus FFT	16.7	1.62×10^{-7}	[10.8, 22.5]

H.9 Failure Analysis and Limits of Attribution

Table 21 reports descriptive failure categories for FFT and VLA-GSE on the real-world evaluation. The counts sum to 82 and 42 failures, respectively. Semantic-grounding failures decrease from 18 to 7, while approach/localization, collision, and post-grasp execution each decrease by seven. Post-grasp execution and grasp/slippage remain the largest categories for GSE, with ten and eight failures. These labels characterize observed failures; they do not establish causal mechanisms.

Table 21: Descriptive real-world failure counts over 240 episodes per method. The comparison uses the same Qwen3-VL backbone and OFT head.

Failure category	FFT	VLA-GSE
Semantic grounding	18	7
Approach/localization	13	6
Grasp/slippage	11	8
Collision	12	5
Post-grasp execution	17	10
Occlusion/recovery	11	6
Total failures	82	42

Sensor-noise failures. VLA-GSE's average Noise score on LIBERO-Plus is 79.4%, below ABot-M0's 86.4% in Table 1. Preliminary inspection suggests that strong observation corruption makes small-object boundaries and graspable regions temporally inconsistent, perturbing localization and approach/grasp predictions; early errors can then propagate during long-horizon execution. We do not observe routing collapse or highly unstable expert selection in the inspected cases. Removing the

auxiliary loss or trace-inverse scaling lowers Long-Noise success from 65.2% to 62.9% and 63.1%, respectively. These observations are qualitative and do not causally separate observation corruption from routing behavior.

Simulation–real-world discrepancy. A simulated cup grasp can succeed while real pouring fails because the grasp is insufficiently stable for lifting and pouring. Embodiment, cup geometry, and contact dynamics are plausible contributors. However, the LIBERO and PiPER evaluations do not match tasks, objects, initial configurations, embodiments, or success criteria. This is a qualitative simulation-success/real-failure observation, not a quantitative estimate of a matched sim-to-real gap.

H.10 Scope of the Additional Evidence

The added evaluations cover two backbone/scale configurations, two simulation benchmark families and embodiments, one real-robot platform, two action representations, and two decoders. The three-seed study in Table 11 characterizes training uncertainty for the core LIBERO-Plus comparison; the single-seed SimplerEnv and real-world paired analyses do not. Same-scale cross-family tests, repeated-seed action and SimplerEnv evaluations, FFT + flow, active-FLOP-matched role ablations, multi-batch/per-layer replays, independently trained scaling controls, and matched simulation–real-world task comparisons remain outside the completed evidence reported here.