Revisiting Parameter Redundancy in Vision-Language-Action Models: Insights from VLM-to-VLA Adaptation

Fengnian Zhang[1,2*], Tao Huang[3*], Siyu Xu[4], Zhong Jin[1†], and Chang Xu[4]

[1] Computer Network Information Center, Chinese Academy of Sciences
fnzhang@cnic.cn, zjin@sccas.cn
[2] University of Chinese Academy of Sciences
[3] School of Computer Science, Shanghai Jiao Tong University
t.huang@sjtu.edu.cn
[4] School of Computer Science, The University of Sydney
{s.xu,c.xu}@sydney.edu.au

Abstract. Vision-Language-Action (VLA) models have made significant strides in embodied intelligence by integrating the powerful representations of pre-trained Vision-Language Models (VLMs). However, the massive parameter scale of VLAs imposes a heavy computational burden, and these models exhibit extreme sensitivity to parameter pruning. Current paradigms often treat the resulting performance degradation as inevitable, relying on fine-tuning or low-rank corrections to recover efficacy. We challenge this convention by questioning whether the removed parameters are truly redundant if VLA pruning necessitates performance recovery to be effective, or if this paradigm masks the indiscriminate pruning of critical parameters. We revisit parameter redundancy through the lens of VLM-to-VLA adaptation, first quantifying the spatial distribution of parameter divergence during adaptation to reveal structured patterns across different modules. Subsequently, we introduce controlled pruning as a diagnostic probe: by comparing the direct impact of removing different parameter subsets on VLA performance without any fine-tuning, we establish a causal link between adaptation-induced divergence signals and functional contributions. Based on the discovered modular heterogeneities, we design a multi-module joint pruning scheme. Evaluations on the LIBERO benchmark demonstrate that our approach reduces the parameters of OpenVLA and $\pi_{0.5}$ by 12%–30% while maintaining approximately 90% of the original performance without any post-pruning recovery. In contrast, existing parameter pruning criteria result in total performance collapse when evaluated under the same recovery-free constraints. Our study reveals the parameter evolution mechanism in VLA adaptation and provides a new path for deploying efficient, robust robotic policies in resource-constrained environments. Code is available at https://github.com/Niannnnnn/VLA_Parameter_Redundancy_VLM2VLA.

Keywords: VLA Models · Parameter Redundancy · Parameter Pruning · VLM-to-VLA Adaptation

* The authors contributed equally. † Corresponding author.

1 Introduction

Vision-Language-Action (VLA) models have rapidly become the core paradigm in embodied intelligence. By adapting large-scale pre-trained Vision-Language Models (VLMs) to downstream action and control tasks, VLAs inherit rich cross-modal representations, showcasing significant generalization advantages in environment variations, object diversity, and task compositions [12, 20, 24, 32]. However, this "inheritance" introduces a long-overlooked yet critical question: **during the VLM-to-VLA adaptation, which parameters truly participate in action generation, and which are merely structural residues from pre-training?**

This issue is amplified by the sheer scale of modern VLA architectures, which typically retain the vast majority of VLM parameters, leading to substantial computational and storage overhead. Various efficiency optimization methods have been proposed [7, 35], including parameter pruning, token pruning, layer pruning, quantization, and lightweight architectural design. However, a recurring phenomenon is that VLA models are extremely sensitive to parameter removal; even moderate pruning can lead to a "performance cliff" where task success rates plummet and model behavior becomes nearly non-functional.

Facing this sensitivity, current mainstream strategies are pragmatic and focused on engineering fixes [4, 10]: they treat post-pruning performance degradation as an unavoidable side effect and compensate for it through post-hoc recovery means such as fine-tuning or low-rank adaptation. A pruning method is deemed "successful" once the model performance reaches an acceptable level during the recovery phase. We fundamentally question this premise. We argue that the widespread reliance on performance recovery masks a more critical issue: **if a pruned VLA model must re-learn to recover its functionality, were the removed parameters truly redundant?**

This observation prompts us to shift the focus from traditional inquiries like "how to prune more" or "how to achieve better recovery" to a more foundational question: **when pruning necessitates explicit performance recovery, does it reflect true redundancy or the indiscriminate pruning of vital parameters?** This question remains under-discussed because existing methods often conflate two distinct problems: the identification of redundant parameters and the repair of pruning-induced damage. Our position is that parameter redundancy in VLAs is not equivalent to that in traditional CNNs or LLMs. Unlike models trained end-to-end on a single task distribution, VLAs are constructed through a structured adaptation process from general vision-language understanding to robotic control [13, 14, 21, 39]. During this process, parameters are selectively reused, re-weighted, or stabilized to support action-centric computation paths.

Based on these reflections, we revisit VLA parameter redundancy from the perspective of the VLM-to-VLA adaptation process. We first quantify the parameter divergence during this transition to reveal structured distribution patterns across different components. However, the relationship between parameter divergence and importance is not necessarily monotonic. Different modules may

follow distinct adjustment mechanisms: in some modules, significantly changed parameters may directly participate in action-centered decisions; in others, stable parameters might provide crucial cross-modal alignment.

To systematically investigate this, we introduce controlled pruning as a diagnostic probe. By evaluating the direct impact of removing parameters with different divergence characteristics (e.g., pruning parameters with the largest vs. smallest divergence) on policy performance, we establish a causal link between divergence signals and functional importance. This allows us to verify if VLM-to-VLA adaptation signals can effectively identify VLA parameter redundancy. Specifically, we systematically test the following hypotheses:

- **Hypothesis I**: If pruning in VLA models requires post-pruning performance recovery, the removed parameters are unlikely to be truly redundant.
- **Hypothesis II**: The parameter differences introduced during VLM-to-VLA adaptation contain useful signals for identifying redundant parameters.
- **Hypothesis III**: The usefulness of VLM-VLA parameter differences for redundancy varies across modules.
- **Hypothesis IV**: When properly utilized, VLM-VLA parameter differences allow structured pruning without requiring post-pruning performance recovery.

Through systematic verification of these hypotheses, we establish a causal chain from adaptation signals to functional importance, and translate the resulting insights into a principled, recovery-free pruning scheme. Our contributions are summarized as follows:

1. We revisit VLA parameter redundancy through the lens of VLM-to-VLA adaptation, introducing a novel diagnostic framework that treats VLA pruning as a controlled intervention to causally link adaptation-induced divergence with parameter functional importance.
2. We reveal that the prevailing recovery-dependent paradigm conceals the indiscriminate removal of vital parameters, and show that the ΔW signals exhibit strong module heterogeneity, necessitating differentiated pruning strategies across vision encoders, language backbones, and cross-modal projectors in VLA models.
3. We propose a multi-module joint pruning scheme that reduces VLA parameter scale and memory usage by 12%–30% while maintaining approximately 90% of the original performance of OpenVLA and $\pi_{0.5}$ *without any post-pruning recovery*, validating the high utility of VLM-to-VLA adaptation signals.

2 Related Work

2.1 Vision-Language-Action (VLA) Models

Recent VLA models have substantially improved embodied agents by adapting large-scale Vision-Language Models (VLMs) to robotic action generation. Pioneering works such as RT-2 [39] and PaLM-E [5] have demonstrated zero-shot

transfer potential across various tasks and environments. OpenVLA [13] established a high-performance baseline for the open-source community by instruction-tuning the Prismatic VLM [11], while $\pi_{0.5}$ [9] adopted the modular and lightweight PaLI-Gemma [2] architecture to balance inference efficiency and performance. Furthermore, works like CogACT [14] and Gr00t [3] explored the deep coupling between multimodal representations and action decision-making. Despite these advances, most VLAs inherit the massive parameter scale of their VLM backbones, while the functional roles of these parameters remain largely unexplored.

2.2 Efficiency and Pruning in VLA Models

To alleviate the computational burden of large-scale VLAs, researchers have proposed various optimization strategies, including parameter pruning [4,10,33], token pruning [16,23,27,31,34], layer pruning [25,34,36,38], quantization [6,22,29], and lightweight architectural design [18,30]. Orthogonally, adaptive test-time compute allocation has also been explored to improve VLA efficiency [15]. Among these, parameter pruning has most directly exposed the extreme sensitivity of VLA models.

RLRC [4] introduced structured pruning based on LLM-Pruner and Taylor importance criteria, utilizing fine-tuning and reinforcement learning (RL) to recover performance. However, empirical findings showed that without Supervised Fine-Tuning (SFT), performance could hardly be recovered even after 2 million RL steps, revealing the indispensability of post-hoc recovery in existing workflows. To address the VLA performance collapse caused by parameter pruning, GLUESTICK [10] utilized SVD to extract principal directions of weight differences and added lightweight corrective terms during inference. While avoiding fine-tuning, it essentially remains a post-hoc remedy.

Unlike studies focused on “how to recover”, we challenge the premise that performance loss is inevitable. We argue that the heavy reliance on recovery reflects a failure to identify redundancy criteria. We explore a precision identification method based on adaptation features to enable direct pruning without any post-hoc compensation.

2.3 Mechanistic Connections between VLM and VLA

The performance of a VLA model is deeply coupled with the pre-training quality of its VLM backbone, yet the relationship is not a simple linear mapping.

VLM4VLA [37] systematically compared various VLM backbones and found that VLM performance on standard benchmarks does not necessarily correlate with downstream VLA task performance. This suggests complex structural reorganization during the transfer from vision-language understanding to robotic control. Actions as Language [8] highlighted the "catastrophic forgetting" of VLM knowledge during VLA fine-tuning. By re-labeling robotic actions as image-text pairs, they maintained VLM capabilities while ensuring VLA performance.

Existing studies primarily focus on final performance comparisons. In contrast, we treat the VLM-to-VLA adaptation process itself as an observable structured signal. By investigating the link between adaptation-induced parameter divergence and policy performance, we seek to understand adaptation from a parametric dimension and utilize it as the core criterion for identifying redundancy.

3 Analysis Framework

This section presents an analysis framework designed to systematically investigate parameter redundancy in Vision-Language-Action (VLA) models. Unlike existing studies that treat pruning merely as a performance optimization technique, we reframe it as a *diagnostic probe* to explore the functional roles of parameters adapted from pre-trained Vision-Language Models (VLMs) in robotic manipulation and control tasks. This shift in perspective allows us to clearly decouple two issues often conflated in prior research: the identification of redundant parameters and the post-pruning recovery of performance.

3.1 Reframing VLA Parameter Pruning as an Analysis Problem

Parameter pruning, a classical method for reducing computational overhead in large-scale neural networks, has been extensively studied. However, in the context of VLA models, a recurring phenomenon has been observed: even the removal of a moderate proportion of parameters can lead to a drastic decline in task success rates. Such performance losses typically necessitate large-scale fine-tuning or low-rank adaptation to recover.

While these recovery strategies are effective in engineering practice, they implicitly accept a fundamental premise: that performance degradation is an inevitable cost of parameter pruning. We argue that this assumption masks a more fundamental question about **what pruning actually removes**: vital parameters undergoing functional reorganization, or genuine structural residues. Let $f(\cdot; W)$ denote a VLA model with parameters W. Given a parameter pruning operator $\mathcal{P}(\cdot; M)$ defined by a binary mask $M \in \{0,1\}^{|W|}$, most pruning methods implicitly optimize the following objective:

$$\max_{M} \ \mathcal{S}\big(f(\cdot; \mathcal{P}(W; M))\big) \quad \text{subject to} \quad \|M\|_0 \leq k, \tag{1}$$

where $\mathcal{S}(\cdot)$ represents the task success rate and k is the sparsity budget.

However, once a fine-tuning process $\mathcal{T}(\cdot)$ is introduced post-pruning, the final evaluated model becomes $f\big(\cdot; \mathcal{T}(\mathcal{P}(W; M))\big)$. In this case, the model's performance no longer reflects the intrinsic functional capability supported solely by the retained parameters; instead, it reflects the model's capacity for re-learning and self-reconstruction under structural perturbation. From an analytical standpoint, this raises a fundamental concern: **if a pruning intervention must**

rely on explicit performance recovery to be effective, were the removed parameters truly redundant? Consequently, we reframe VLA parameter pruning as an analytical problem rather than an optimization goal. In this framework, pruning (without subsequent performance recovery) is treated as a *controlled intervention* used to reveal the direct impact of parameter removal on policy behavior. Under this paradigm, the necessity of performance recovery is no longer evidence of successful redundancy identification, but rather a signal of its failure.

3.2 VLM–VLA Parameter Difference as a Structural Signal

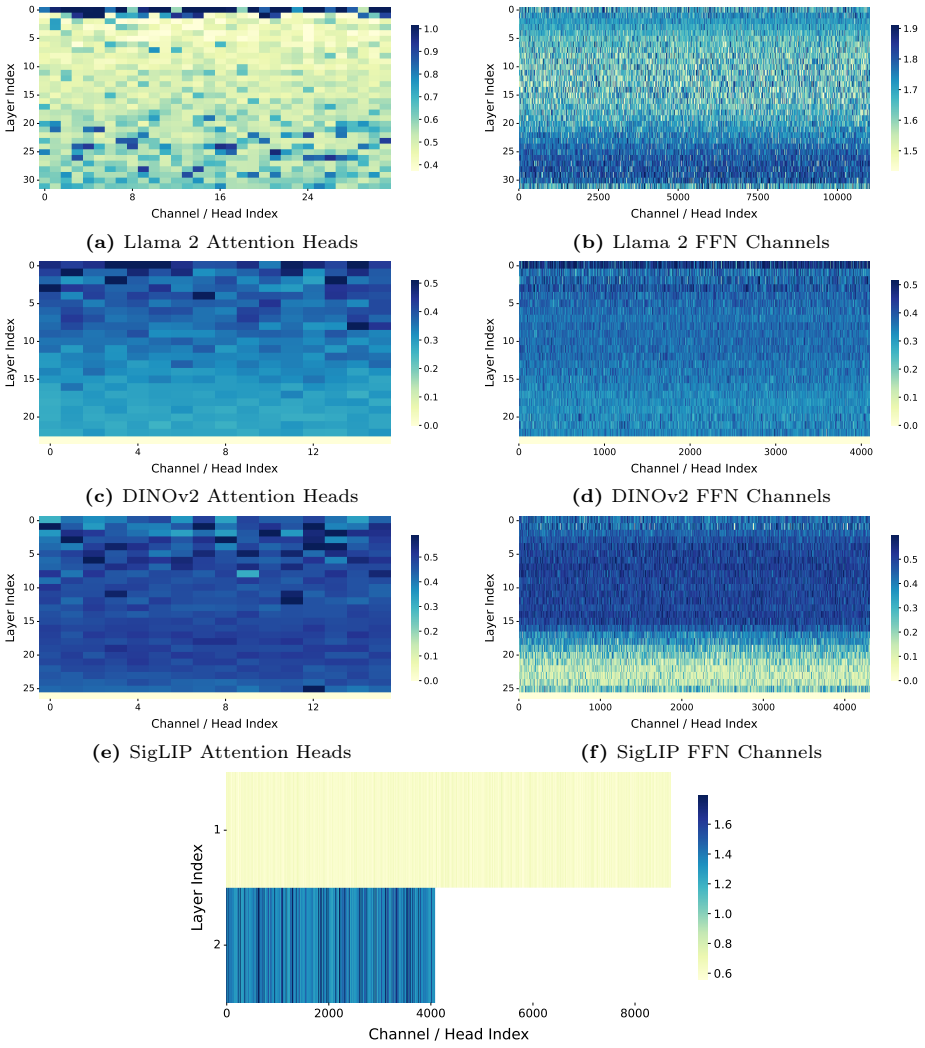
VLA models are typically not trained from scratch but are initialized from pre-trained VLMs and adapted using robotic manipulation data. Let W^{VLM} denote the parameters of a pre-trained VLM, and W^{VLA} denote the corresponding parameters after adaptation. In the shared VLM backbone subspace, we define the relative parameter divergence as:

$$\Delta W_{\mathrm{rel}} = \frac{\|W^{\mathrm{VLA}} - W^{\mathrm{VLM}}\|_2}{\|W^{\mathrm{VLM}}\|_2}. \tag{2}$$

We hypothesize that ΔW is not mere random noise but contains structured information regarding the functional reorganization of weights during adaptation. To verify this, we visualized the weight divergence distributions for two representative pairs: ⟨Prismatic, OpenVLA⟩ and ⟨PaLI-Gemma, $\pi_{0.5}$⟩.

In the **OpenVLA** (Prismatic-based) model, parameter divergence exhibits a highly non-uniform distribution across components (see Fig. 1):

- **Language Model (Llama 2):** Parameter updates follow a distinct "three-stage" vertical distribution. The initial layers (L_0) undergo dense calibration to handle multimodal fusion after token injection; the middle layers (L_1–L_{23}) remain relatively stable; and the terminal layers (L_{24}–L_{31}) fluctuate again, reflecting the refined mapping to the action semantic space. In the Feed-Forward Network (FFN) modules, we observe inter-layer fluctuations and distinct "strip-like" sparsity in the channel dimension, suggesting that embodied knowledge is encoded in specific sub-channels rather than uniformly distributed.
- **Vision Backbone (DINOv2&SigLIP):** For DINOv2, updates in Attention and FFN are concentrated in shallow layers, focusing on low-level visual cues like grasping and obstacle avoidance. Conversely, SigLIP shows stronger responses in deeper layers, with its FFN channel updates generally higher than those of DINOv2, serving as a semantic supplement to align visual features with the language model.
- **Projector:** Comprising FFN structures, its parameter changes show a monotonic increase. The late-stage mapping layers (fc_2, fc_3) fluctuate far more than the initial up-sampling layer (fc_1), indicating that semantic transformation layers near the language model entrance have higher update priority.



(a) Llama 2 Attention Heads **(b)** Llama 2 FFN Channels

(c) DINOv2 Attention Heads **(d)** DINOv2 FFN Channels

(e) SigLIP Attention Heads **(f)** SigLIP FFN Channels

(g) Projector FFN Channels

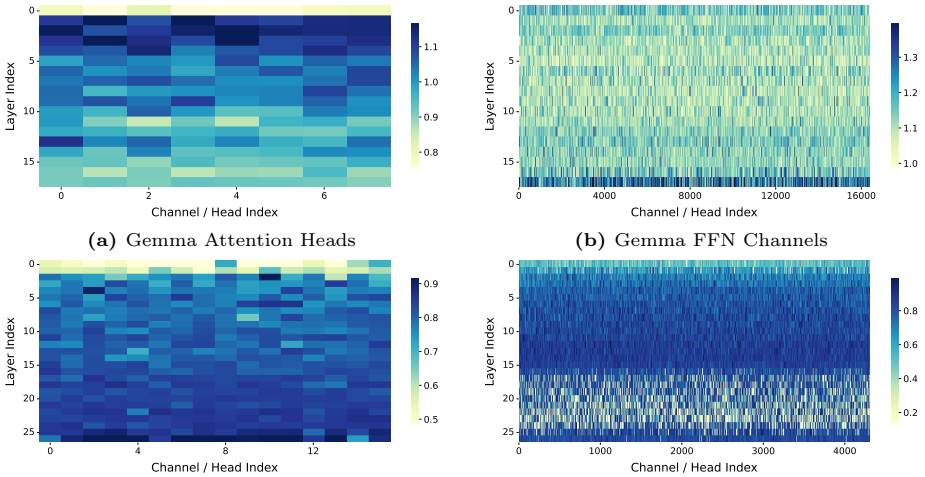
Fig. 1: Visualizing the relative parameter divergence ΔW_{rel} between the **Prismatic** (VLM) and **OpenVLA** (VLA) model pair. The color intensity indicates the magnitude of divergence: **darker blue** denotes significant parameter shifts, while **brighter yellow** represents minimal change. Subfigures (a)–(g) display the divergence across different modules, calculated at the granularity of individual attention heads or FFN channels.

In the $\pi_{0.5}$ (PaLI-Gemma-based) model, the modular design leads to more regular evolution patterns (see Fig. 2):

– **Language Model (Gemma):** The Multi-Query Attention (MQA) mechanism shows high discriminative signals in head dimensions. Low layers (L_0)

show minimal divergence, retaining general text-parsing priors; middle layers (L_1–L_9) exhibit large divergence, reflecting intense functional reorganization for redirection to embodied task attention; high layers (L_{10}–L_{17}) show a slight decrease, indicating stability after achieving the semantic-to-action mapping.

– **Vision Tower:** Attention modules show mild changes, suggesting well-preserved visual priors. FFN modules in the middle (L_3–L_{15}) and high (L_{25}–L_{26}) layers show strong divergence signals, providing a key basis for identifying redundancy in vision modules.



(a) Gemma Attention Heads **(b)** Gemma FFN Channels

(c) Vision Attention Heads **(d)** Vision FFN Channels

Fig. 2: Visualizing the relative parameter divergence ΔW_{rel} for the **PaLI-Gemma** (VLM) and $\pi_{0.5}$ (VLA) model pair. The color intensity indicates the magnitude of divergence: **darker blue** denotes significant parameter shifts, while **brighter yellow** represents minimal change. Subfigures (a)–(d) display the divergence across different modules, calculated at the granularity of individual attention heads or FFN channels.

These systematic observations lead to a crucial conclusion: parameter updates during VLM-to-VLA adaptation are not random noise but exhibit strong **structured heterogeneity** in both selective Attention-head reorganization and localized FFN-channel activation. This non-random distribution provides the empirical foundation for establishing precise VLA redundancy criteria and breaking the "prune-then-collapse" bottleneck.

3.3 Controlled Pruning as a Diagnostic Instrument

To link parameter divergence with model behavior, we introduce **controlled pruning** as a diagnostic tool. Given a target module (g, h) and a pruning ratio r, we construct two complementary strategies based on $|\Delta W|$:

– **Highest-difference pruning:**

$$M^{\text{high}}(r) = \text{Top-}r\% \text{ parameters ranked by } |\Delta W| \tag{3}$$

– **Lowest-difference pruning:**

$$M^{\text{low}}(r) = \text{Bottom-}r\% \text{ parameters ranked by } |\Delta W| \tag{4}$$

The pruned parameters are defined as $W^{\text{pruned}} = \mathcal{P}(W^{\text{VLA}}; M)$, where the pruning is applied directly *without any fine-tuning*. To ensure the reproducibility of our diagnostic probe and clarify how ΔW is aggregated across multi-matrix modules (e.g., LLM FFNs), we provide the systematic mask construction process in Algorithm 1.

Algorithm 1 Module-Aware VLA Mask Construction

Require: $W^{\text{VLM}}, W^{\text{VLA}}$, Pruning ratio r, Strategy $\in$ {Lowest, Highest}
Ensure: Binary masks M for VLA parameters
Initialize: Importance score pool $S \leftarrow \emptyset$
for each module $\mathcal{L} \in$ {LLM, Vision, Projector} **do**
  for each targeted sub-module $W \in \mathcal{L}$ **do**
    $\Delta W \leftarrow W^{\text{VLA}} - W^{\text{VLM}}$
    Dimension Alignment: Identify aggregation dim d
    {e.g., $d = 1$ for input-side projection; $d = 0$ for output-side projection}
    Metric Calculation: $v \leftarrow \|\Delta W\|_{L2,\text{dim}=d} \;/\; (\|W^{\text{VLM}}\|_{L2,\text{dim}=d} + \epsilon)$
    if Target is LLM FFN **then**
      Accumulate v across {*gate, up, down*} projections to align FFN channels
    else if Target is Attention Head **then**
      $v \leftarrow$ Aggregate v to head-level importance via mean operator
    end if
    Add aggregated scores v to pool S
  end for
end for
Global Ranking: Find threshold τ for the **Lowest** or **Highest** $r\%$ elements in S
Mask Assignment: Generate unit mask M_{unit} based on threshold τ
Intra-module Broadcast: Expand M_{unit} to parameter-wise masks M.
{For FFNs, M_{unit} masks the *output* of fc1/up/gate and the *input* of fc2/down simultaneously.}
return M

3.4 Unified Pipeline for Causal Analysis

We employ a unified analysis pipeline: (1) Select correlated ⟨VLM, VLA⟩ pairs; (2) Calculate ΔW across sub-modules; (3) Analyze spatial distribution patterns; (4) Apply controlled pruning interventions; (5) Evaluate direct inference behavior without recovery; (6) Perform cross-module and cross-model comparisons.

This methodology allows us to systematically identify which parameters are redundant and provides the theoretical support for exploration of recovery-free pruning strategies in Sec. 4.

4 Experiments

4.1 Experimental Settings

Dataset. We conduct experiments on the **LIBERO** benchmark [17], the mainstream evaluation suite in embodied manipulation. LIBERO is designed to test skills inspired by human activities, requiring agents equipped with a Franka Panda arm to complete tasks based on natural language instructions and visual observations. The benchmark includes four sub-datasets: **LIBERO-Spatial** (same objects, different spatial layouts), **LIBERO-Object** (same layouts, different object categories), **LIBERO-Goal** (diverse task goals), and **LIBERO-Long** (long-horizon tasks). We use the Success Rate (SR) of each task set as the core performance metric.

Baselines. Our study is based on two representative VLM-to-VLA model pairs: ⟨Prismatic, OpenVLA⟩ and ⟨PaLI-Gemma, $\pi_{0.5}$⟩. The former represents the classic architecture adapted from large-scale LLMs (Llama-2 [28]), while the latter represents emerging modular and lightweight VLA models. During the discovery and verification phases (Sec. 4.2–Sec. 4.4), we use these original models as Backbone VLA baselines. In the final algorithm application experiments (Sec. 4.5), we introduce widely-adopted pruning criteria as comparative baselines to evaluate their efficacy in a recovery-free setting, including structured pruning methods **LLM-Pruner** [19], **FLAP** [1], and the importance-based sparsification method **Wanda** [26].

Implementation Details. All experiments are executed on NVIDIA A100 GPU platforms. In diagnostic experiments without performance recovery, models are evaluated via direct inference after pruning. In control experiments involving recovery (Sec. 4.2), we utilize **LoRA** fine-tuning under the FSDP distributed strategy.

4.2 Hypothesis I: If Pruning in VLA Models Requires Post-Pruning Performance Recovery, the Removed Parameters Are Unlikely to Be Truly Redundant

This subsection examines the "fine-tuning compensation paradigm" in VLA pruning. We question whether post-pruning performance recovery stems from true redundancy or the repair of "falsely killed" vital parameters. Using the Prismatic-OpenVLA pair, we compare three strategies—*Highest-diff*, *Lowest-diff*, and *Random*—targeting the LLM's FFN intermediate layers. We evaluate performance pre- and post-LoRA fine-tuning (FSDP, 10k steps, LR 1e-4).

Table 1 reveals a critical paradox: in immediate evaluations of the LLM FFN, removing low-divergence channels (*Lowest-diff*) causes total performance

Table 1: Model performance comparison before and after fine-tuning. LIBERO-Spatial, Baseline SR = 84.7%.

Pruning Strategy	Ratio (%)	SR (Pre-FT) (%)	SR (Post-FT) (%)
Lowest-diff	20	1.5	86.5
Lowest-diff	50	0.0	81.0
Lowest-diff	80	0.0	76.4
Highest-diff	20	76.3	85.8
Highest-diff	50	20.5	84.1
Highest-diff	80	0.0	80.7
Random	20	12.2	86.0
Random	50	0.0	82.2
Random	80	0.0	77.6

collapse. However, LoRA fine-tuning enables all configurations to recover to or exceed baseline levels, regardless of initial damage (even from 0.0% SR). This "strong compensation" masks the inherent quality of pruning decisions.

Further causal analysis (Fig. 3) shows that convergence steps required for recovery scale with the pruning ratio. This increasing difficulty suggests that higher pruning ratios introduce deeper structural perturbations. These findings support **Hypothesis I**: reliance on recovery essentially repairs structural damage from "parameter mis-deletion." Thus, effective redundancy identification must maintain core functionality without requiring recovery.

4.3 Hypothesis II: The Parameter Differences Introduced During VLM-to-VLA Adaptation Contain Useful Signals for Identifying Redundant Parameters

We systematically evaluate the effectiveness of ΔW in identifying redundancy by comparing *Highest-diff* and *Lowest-diff* pruning across OpenVLA and $\pi_{0.5}$. The results reveal that parameter divergence is not a global importance metric but exhibits profound **Module Heterogeneity**.

In OpenVLA's DINOv2 (Table 2), we observe a striking "sensitivity reversal": for attention heads, pruning the highest-diff heads causes collapse (1.6% SR), whereas for FFN channels, pruning the lowest-diff channels leads to collapse (0.0%). This indicates that functional importance is highly path-dependent even within the same backbone. Similar patterns emerge in the language: removing the lowest-diff attention heads or FFN channels leads to total degradation (0.0% and 2.7% SR, respectively), while pruning the results in total degradation (0.0% and 2.7% SR, respectively), whereas the highest-diff components maintain high performance. In contrast, SigLIP shows minimal sensitivity to ΔW, consistent with its role as an auxiliary semantic supplement [13].

The modular $\pi_{0.5}$ model (Table 3) exhibits even clearer trends. Pruning highest-diff FFN channels in both vision and language barely affects performance

Table 2: Impact of VLM-VLA parameter differences across modules (OpenVLA). LIBERO-Spatial, Baseline SR = 84.7%.

Module	Sub-module	Ratio	SR (%)	
			High-diff (H)	Low-diff (L)
Vision	DINOv2 Attn (head)	0.125	1.6	76.7
Vision	DINOv2 FFN (channel)	0.20	82.0	0.0
Vision	SigLIP Attn (head)	0.125	83.4	80.7
Vision	SigLIP FFN (channel)	0.20	75.1	81.7
Language	Llama2 Attn (head)	0.125	84.3	0.0
Language	Llama2 FFN (channel)	0.20	72.0	2.7
Projector	FFN (channel)	0.30	0.0	54.0

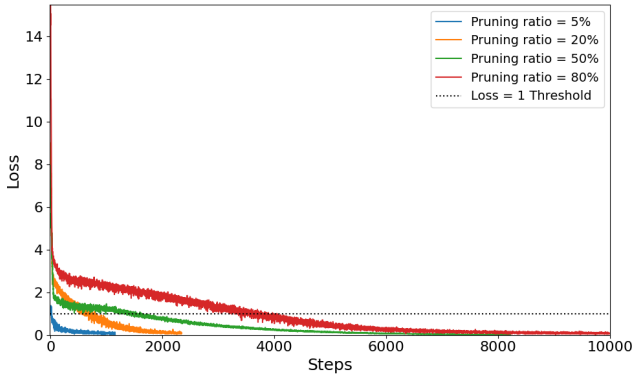


Fig. 3: Causal analysis of recovery difficulty: Convergence steps vs. Pruning ratio.

(∼95.0% SR), while removing lowest-diff channels causes significant drops. Language model attention also collapses (0.0%) only when the lowest-diff heads are removed. These cross-model observations confirm **Hypothesis II**: VLM-to-VLA parameter divergence contains structured signals that effectively distinguish vital from redundant parameters across different computational roles.

4.4 Hypothesis III: The Usefulness of VLM-VLA Parameter Differences for Redundancy Varies Across Modules

We further explore the boundaries of the ΔW signal utility. We find that the contribution of VLA components to the final performance is unequal, determining the strategic focus of global compression.

The Projector module is "fragile and non-selective." As shown in Table 4, across pruning ratios of 0.2 to 0.5, both pruning strategies lead to collapse. This suggests the Projector functions primarily as a cross-modal alignment interface; once this bottleneck is damaged, performance degrades regardless of which parameters are removed. It must be strictly protected. In contrast, SigLIP exhibits

Table 3: Impact of VLM-VLA parameter differences across modules ($\pi_{0.5}$). LIBERO-Spatial, Baseline SR = 98.8%.

Module	Sub-module	Ratio	SR (%)	
			High-diff (H)	**Low-diff (L)**
Language	Gemma Attn (head)	0.20	85.0	0.0
Language	Gemma FFN (channel)	0.50	95.0	5.0
Vision	SigLIP Attn (head)	0.20	90.0	55.0
Vision	SigLIP FFN (channel)	0.50	95.0	15.0

Table 4: Pruning robustness of weakly-coupled modules (OpenVLA). LIBERO-Spatial, Baseline SR = 84.7%.

Module	Sub-module	Ratio	SR (%)	
			High-diff (H)	**Low-diff (L)**
Projector	FFN (channel)	0.20	84.5	78.5
Projector	FFN (channel)	0.30	0.0	54.0
Projector	FFN (channel)	0.50	0.0	0.0
Vision	SigLIP Attn (head)	1.00	47.0	–
Vision	SigLIP FFN (channel)	1.00	70.0	–

extreme robustness. Even with a pruning ratio of 1.0 (total removal), OpenVLA maintains moderate success. This contrasts with DINOv2, where even slight pruning causes failure. DINOv2 serves as the primary structural representation module, while SigLIP only provides supplementary semantic information. These results validate **Hypothesis III**: the signal's usability depends on the module's functional role, requiring differentiated allocation logic in joint pruning.

4.5 Hypothesis IV: When Properly Utilized, VLM-VLA Parameter Differences Allow Structured Pruning Without Requiring Post-Pruning Performance Recovery

We verify if the revealed module heterogeneity enables a **recovery-free** pruning scheme. We design a multi-module joint pruning algorithm with three configurations: *Pruned-Light, Moderate,* and *Aggressive*. Following [4, 10], we compare our approach against representative pruning criteria evaluated under identical recovery-free conditions: **LLM-Pruner** [19], **FLAP** [1], and **Wanda** [26].

Table 5 shows results on OpenVLA. Traditional methods (LLM-Pruner, FLAP, Wanda) suffer catastrophic failure in the absence of performance recovery. In a **matched-capacity comparison** at 12.4GB, our *Moderate* config maintains 62.3% SR, while LLM-Pruner achieves only 1.0%. This 60% gap demonstrates that conventional pruning metrics fail in VLA because it removes vital parameters undergoing functional reorganization. In the *Aggressive* config (5.7B), the model still retains 60% efficacy in core tasks like Spatial and Object.

Table 5: Task success rates (%) on LIBERO benchmark for pruned OpenVLA variants.

Model	Params (B)	Mem (GB)	Spatial (%)	Object (%)	Goal (%)	Long (%)	Average (%)
OpenVLA (Baseline)	7.5	14.9	84.7	88.4	79.2	53.7	76.5 (+0.0)
LLM-Pruner [4]	6.2	12.4	23.4	-	-	1.0	-
FLAP [4]	6.3	12.5	0.2	-	-	0.0	-
Wanda(Full Sparse) [10]	-	10.2	0.0	13.4	0.8	0.0	7.1 (-69.4)
Wanda(Sparse Lang. BB) [10]	-	10.6	31.2	50.8	20.0	12.4	28.6 (-47.9)
Wanda(75% Sparse Lang. BB) [10]	-	12.0	25.4	49.0	20.8	11.4	26.7 (-49.8)
Ours-Light	6.6	13.0	78.3	82.5	74.0	46.8	**70.4 (-6.1)**
Ours-Moderate	6.2	12.4	70.5	74.9	64.7	39.0	**62.3 (-14.2)**
Ours-Aggressive	5.7	11.3	59.0	65.7	56.0	29.5	**52.5 (-24.0)**

Table 6: Task success rates (%) on LIBERO benchmark for pruned $\pi_{0.5}$ variants.

Model	Params (B)	Mem (GB)	Spatial (%)	Object (%)	Goal (%)	Long (%)	Average (%)
$\pi_{0.5}$ (Baseline)	3.6	7.3	98.8	98.2	98.0	92.4	96.9 (+0.0)
Ours-Light	3.0	6.1	95.5	94.0	95.0	88.5	**93.3 (-3.6)**
Ours-Moderate	2.8	5.6	90.7	90.1	91.3	84.0	**89.0 (-7.9)**
Ours-Aggressive	2.5	5.0	83.6	83.0	84.5	77.1	**82.1 (-14.8)**

The pattern holds for $\pi_{0.5}$ (Table 6). Our *Moderate* scheme reduces memory from 7.3GB to 5.6GB, with SR only shifting from 96.9% to 89.0%. This supports **Hypothesis IV**: VLA pruning sensitivity is not insurmountable. By correctly interpreting structural importance from adaptation, we can achieve efficient deployment on edge devices without expensive compensation.

5 Conclusion

This study systematically investigates parameter redundancy and parameter pruning sensitivity in VLA models by focusing on the structured signals inherent in VLM-to-VLA adaptation. Our analysis reveals that the prevailing reliance on post-hoc recovery often masks the "indiscriminate killing" of vital parameters, whereas adaptation-induced parameter divergence (ΔW) serves as a high-fidelity criterion for parameter redundancy identification. We uncover significant structural heterogeneity across modules: primary structural vision encoders and language backbones exhibit high pruning sensitivity, cross-modal projectors act as fragile alignment bottlenecks, while semantic-supplementary vision towers demonstrate remarkable redundancy resilience. By leveraging these insights

through differentiated pruning logic, we demonstrate across multiple representative VLA baselines that substantial model compression is achievable without any fine-tuning or compensation mechanisms. Ultimately, understanding parameter evolution from general perception to embodied action is crucial for efficient VLA deployment, providing a foundation for future research in resource-constrained robotic intelligence.

Acknowledgements

This work was supported by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA0460301 and the National Natural Science Foundation of China under Grant No. 62506235.

References

1. An, Y., Zhao, X., Yu, T., Tang, M., Wang, J.: Fluctuation-based adaptive structured pruning for large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 10865–10873 (2024)
2. Beyer, L., Steiner, A., Pinto, A.S., Kolesnikov, A., Wang, X., Salz, D., Neumann, M., Alabdulmohsin, I., Tschannen, M., Bugliarello, E., et al.: Paligemma: A versatile 3b vlm for transfer. arXiv preprint arXiv:2407.07726 (2024)
3. Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: Gr00t n1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734 (2025)
4. Chen, Y., Li, X.: Rlrc: Reinforcement learning-based recovery for compressed vision-language-action models. arXiv preprint arXiv:2506.17639 (2025)
5. Driess, D., Xia, F., Sajjadi, M.S., Lynch, C., Chowdhery, A., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., et al.: Palm-e: An embodied multimodal language model (2023)
6. Fang, H., Liu, Y., Du, Y., Du, L., Yang, H.: Sqap-vla: A synergistic quantization-aware pruning framework for high-performance vision-language-action models. arXiv preprint arXiv:2509.09090 (2025)
7. Guan, W., Hu, Q., Li, A., Cheng, J.: Efficient vision-language-action models for embodied manipulation: A systematic survey. arXiv preprint arXiv:2510.17111 (2025)
8. Hancock, A.J., Wu, X., Zha, L., Russakovsky, O., Majumdar, A.: Actions as language: Fine-tuning vlms into vlas without catastrophic forgetting. arXiv preprint arXiv:2509.22195 (2025)
9. Intelligence, P., Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Galliker, M.Y., Ghosh, D., Groom, L., Hausman, K., Ichter, B., Jakubczak, S., Jones, T., Ke, L., LeBlanc, D., Levine, S., Li-Bell, A., Mothukuri, M., Nair, S., Pertsch, K., Ren, A.Z., Shi, L.X., Smith, L., Springenberg, J.T., Stachowicz, K., Tanner, J., Vuong, Q., Walke, H., Walling, A., Wang, H., Yu, L., Zhilinsky, U.: $\pi_{0.5}$: a vision-language-action model with open-world generalization (2025), https://arxiv.org/abs/2504.16054
10. Jabbour, J., Kim, D.K., Smith, M., Patrikar, J., Ghosal, R., Wang, Y., Agha, A., Reddi, V.J., Omidshafiei, S.: Don't run with scissors: Pruning breaks vla models but they can be recovered. arXiv preprint arXiv:2510.08464 (2025)

11. Karamcheti, S., Nair, S., Balakrishna, A., Liang, P., Kollar, T., Sadigh, D.: Prismatic vlms: Investigating the design space of visually-conditioned language models. In: Forty-first International Conference on Machine Learning (2024)
12. Kawaharazuka, K., Oh, J., Yamada, J., Posner, I., Zhu, Y.: Vision-language-action models for robotics: A review towards real-world applications. IEEE Access (2025)
13. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
14. Li, Q., Liang, Y., Wang, Z., Luo, L., Chen, X., Liao, M., Wei, F., Deng, Y., Xu, S., Zhang, Y., et al.: Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. arXiv preprint arXiv:2411.19650 (2024)
15. Li, W., Su, X., Cao, Y., Xu, H., Xia, X., You, S., Chen, Y., Xu, C.: Vla-attc: Adaptive test-time compute for vla models with relative action critic model. arXiv preprint arXiv:2605.01194 (2026)
16. Li, Y., Meng, Y., Sun, Z., Ji, K., Tang, C., Fan, J., Ma, X., Xia, S., Wang, Z., Zhu, W.: Sp-vla: A joint model scheduling and token pruning approach for vla model acceleration. arXiv preprint arXiv:2506.12723 (2025)
17. Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., Stone, P.: Libero: Benchmarking knowledge transfer for lifelong robot learning. Advances in Neural Information Processing Systems **36**, 44776–44791 (2023)
18. Liu, J., Liu, M., Wang, Z., An, P., Li, X., Zhou, K., Yang, S., Zhang, R., Guo, Y., Zhang, S.: Robomamba: Efficient vision-language-action model for robotic reasoning and manipulation. Advances in Neural Information Processing Systems **37**, 40085–40110 (2024)
19. Ma, X., Fang, G., Wang, X.: Llm-pruner: On the structural pruning of large language models. Advances in neural information processing systems **36**, 21702–21720 (2023)
20. Ma, Y., Song, Z., Zhuang, Y., Hao, J., King, I.: A survey on vision-language-action models for embodied ai. arXiv preprint arXiv:2405.14093 (2024)
21. Mitra, C., Luo, Y., Saravanan, R., Niu, D., Pai, A., Thomason, J., Darrell, T., Anwar, A., Ramanan, D., Herzig, R.: Mechanistic finetuning of vision-language-action models via few-shot demonstrations. arXiv preprint arXiv:2511.22697 (2025)
22. Park, S., Kim, H., Kim, S., Jeon, W., Yang, J., Jeon, B., Oh, Y., Choi, J.: Saliency-aware quantized imitation learning for efficient robotic control. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13140–13150 (2025)
23. Pei, X., Chen, Y., Xu, S., Wang, Y., Shi, Y., Xu, C.: Action-aware dynamic pruning for efficient vision-language-action manipulation. arXiv preprint arXiv:2509.22093 (2025)
24. Sapkota, R., Cao, Y., Roumeliotis, K.I., Karkee, M.: Vision-language-action (vla) models: Concepts, progress, applications and challenges. arXiv preprint arXiv:2505.04769 (2025)
25. Shukor, M., Aubakirova, D., Capuano, F., Kooijmans, P., Palma, S., Zouitine, A., Aractingi, M., Pascal, C., Russi, M., Marafioti, A., et al.: Smolvla: A vision-language-action model for affordable and efficient robotics. arXiv preprint arXiv:2506.01844 (2025)
26. Sun, M., Liu, Z., Bair, A., Kolter, J.Z.: A simple and effective pruning approach for large language models. arXiv preprint arXiv:2306.11695 (2023)

27. Tan, X., Yang, Y., Ye, P., Zheng, J., Bai, B., Wang, X., Hao, J., Chen, T.: Think twice, act once: Token-aware compression and action reuse for efficient inference in vision-language-action models. arXiv preprint arXiv:2505.21200 (2025)
28. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
29. Wang, H., Xiong, C., Wang, R., Chen, X.: Bitvla: 1-bit vision-language-action models for robotics manipulation. arXiv preprint arXiv:2506.07530 (2025)
30. Wen, J., Zhu, Y., Li, J., Zhu, M., Tang, Z., Wu, K., Xu, Z., Liu, N., Cheng, R., Shen, C., et al.: Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. IEEE Robotics and Automation Letters (2025)
31. Xu, S., Wang, Y., Xia, C., Zhu, D., Huang, T., Xu, C.: Vla-cache: Efficient vision-language-action manipulation via adaptive token caching. arXiv preprint arXiv:2502.02175 (2025)
32. Xu, S., Wang, Z., Wang, Y., Xia, C., Huang, T., Xu, C.: Affordance field intervention: Enabling vlas to escape memory traps in robotic manipulation. arXiv preprint arXiv:2512.07472 (2025)
33. Xu, Y., Yang, Y., Fan, Z., Liu, Y., Li, Y., Li, B., Zhang, Z.: Qvla: Not all channels are equal in vision-language-action model's quantization. arXiv preprint arXiv:2602.03782 (2026)
34. Yang, Y., Wang, Y., Wen, Z., Zhongwei, L., Zou, C., Zhang, Z., Wen, C., Zhang, L.: Efficientvla: Training-free acceleration and compression for vision-language-action models. arXiv preprint arXiv:2506.10100 (2025)
35. Yu, Z., Wang, B., Zeng, P., Zhang, H., Zhang, J., Gao, L., Song, J., Sebe, N., Shen, H.T.: A survey on efficient vision-language-action models. arXiv preprint arXiv:2510.24795 (2025)
36. Yue, Y., Wang, Y., Kang, B., Han, Y., Wang, S., Song, S., Feng, J., Huang, G.: Deer-vla: Dynamic inference of multimodal large language models for efficient robot execution. Advances in Neural Information Processing Systems **37**, 56619–56643 (2024)
37. Zhang, J., Chen, X., Wang, Q., Li, M., Guo, Y., Hu, Y., Zhang, J., Bai, S., Lin, J., Chen, J.: Vlm4vla: Revisiting vision-language-models in vision-language-action models. arXiv preprint arXiv:2601.03309 (2026)
38. Zhang, R., Dong, M., Zhang, Y., Heng, L., Chi, X., Dai, G., Du, L., Du, Y., Zhang, S.: Mole-vla: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. arXiv preprint arXiv:2503.20384 (2025)
39. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al.: Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: Conference on Robot Learning. pp. 2165–2183. PMLR (2023)

A Detailed Model Composition and Parameter Dimensions

In this section, we provide a comprehensive breakdown of the parameter composition for the two primary VLA models utilized in our study: OpenVLA (based on the Prismatic VLM) and $\pi_{0.5}$ (based on PaliGemma). Our pruning analysis specifically targets the weight parameters within the Vision Tower, Projector, and Language Model modules.

A.1 Parameter Module Distribution

To illustrate the structural complexity of the VLA baselines, we first count the number of individual weight tensors (including biases and normalization parameters) across the functional modules. As shown in Table 7, the majority of the parameter objects are concentrated in the Vision Backbone and Language Model.

Table 7: Model parameter specifications and module distribution.

Model	Total	Vision	Projector	Language	Expert	Other
OpenVLA	982	685	6	291	–	–
$\pi_{0.5}$	812	437	2	164	201	8

A.2 Detailed Weight Dimensions for Pruning

Our pruning methodology focuses on the fundamental channel units within the Attention and Feed-Forward Network (FFN) blocks. Table 8 details the dimensions for the key weight-bearing layers. Note that for the Language Model in $\pi_{0.5}$, Multi-Query Attention (MQA) is employed, resulting in fewer K/V heads relative to Q heads.

A.3 Projector Configurations

The cross-modal projectors serve as the bridge between vision and language.

- **OpenVLA:** Uses a 2-layer MLP (3 weight tensors) with dimensions mapping from the fused vision tokens ($1024 + 1152 = 2176$) to the Llama hidden size (4096).
- $\pi_{0.5}$**:** Employs a single linear projection (2 weight tensors including bias) mapping from 1152 to 2048.

Table 8: Architectural dimensions for the primary weight-bearing modules. d_{model}: hidden size; L: number of layers; N_{head}: attention heads; d_{int}: intermediate size.

Model Family	Component	d_{model}	L	N_{head}	d_{int}
OpenVLA	Vision (DINOv2)	1024	24	16	4096
	Vision (SigLIP)	1152	27	16	4304
	Language (Llama-2)	4096	32	32	11008
$\pi_{0.5}$	Vision (SigLIP)	1152	27	16	4304
	Language (Gemma)	2048	18	8	16384

* In $\pi_{0.5}$, Q uses 8 heads while K/V use 1 head (MQA) to reduce memory overhead during inference.

B Extended Experimental Results

This section provides a complete empirical verification of our hypotheses across all LIBERO sub-datasets. We provide granular data for both OpenVLA and $\pi_{0.5}$ to demonstrate the universality of our observations.

B.1 Generalizability of Hypothesis I: The Recovery Paradox

Table 9 illustrates the "strong compensation" effect. Across all task suites and both model families, fine-tuning consistently bridges the performance gap caused by sub-optimal pruning decisions, reinforcing the need for recovery-free evaluation.

Table 9: Pre- and Post-finetuning Success Rate (SR %) across all LIBERO sub-datasets. Pruning targets: Llama-2 FFN (20%) for OpenVLA. *Pre.* and *Post.* denote results before and after fine-tuning, respectively.

Model	Strategy	Spatial		Object		Goal		Long	
		Pre.	*Post.*	*Pre.*	*Post.*	*Pre.*	*Post.*	*Pre.*	*Post.*
OpenVLA	Baseline	84.7		88.4		79.2		53.7	
	Lowest-diff	1.5	86.5	3.2	86.8	0.0	79.5	0.0	49.5
	Highest-diff	76.3	85.8	78.4	89.5	71.2	81.6	44.8	53.9
	Random	12.2	86.0	14.8	87.2	7.6	76.8	4.2	51.2

B.2 Validation of Hypothesis II: Signal Validity and Module Heterogeneity

We evaluate the contrastive impact of *High-diff* (H) and *Low-diff* (L) strategies. As shown in Table 10, the "sensitivity reversal" (e.g., in DINOv2 vs. Llama-2)

is a persistent structural property. We have included SigLIP and Projector data for OpenVLA to provide a full-spectrum analysis of the adaptation signal.

Table 10: Contrastive analysis of **High-diff (H)** vs. **Low-diff (L)** strategies (SR %) across multiple modules. Note the performance fluctuations reflecting inherent task difficulty in the Long-horizon suite.

Model	Module	Ratio	Spatial		Object		Goal		Long	
			H	**L**	**H**	**L**	**H**	**L**	**H**	**L**
OpenVLA	DINOv2 Attn	0.125	1.6	**76.7**	5.4	**79.8**	2.1	**73.5**	0.0	**44.2**
	DINOv2 FFN	0.20	**82.0**	0.0	**83.7**	3.2	**75.4**	1.2	**50.6**	0.0
	SigLIP Attn	0.125	83.4	79.7	85.2	81.5	77.1	75.2	48.6	42.0
	SigLIP FFN	0.20	75.1	81.0	76.0	82.2	65.5	70.1	43.5	48.0
	Llama2 Attn	0.125	**84.3**	0.0	**87.1**	0.0	**78.4**	0.0	**52.5**	0.0
	Llama2 FFN	0.20	**72.0**	2.7	**79.2**	4.8	**66.5**	2.4	**41.8**	0.8
$\pi_{0.5}$	SigLIP Attn	0.20	**90.0**	55.0	**90.0**	60.0	**85.0**	50.0	**80.0**	45.0
	SigLIP FFN	0.50	**95.0**	15.0	**90.0**	20.0	**90.0**	15.0	**85.0**	10.0
	Gemma Attn	0.20	**85.0**	0.0	**85.0**	5.0	**80.0**	5.0	**75.0**	0.0
	Gemma FFN	0.50	**95.0**	5.0	**95.0**	10.0	**90.0**	5.0	**85.0**	5.0

B.3 Validation of Hypothesis III: Signal Boundaries and Robustness

This subsection explores the limits of pruning robustness. Table 11 demonstrates the extreme robustness of SigLIP and the severe fragility of the Projector across all task suites. The Projector's collapse under both strategies at Ratio 0.3 identifies it as a critical non-selective bottleneck.

Table 11: Robustness limits of weakly-coupled and bottleneck modules (SR %). Ratio 1.0 represents the complete removal of the component.

Model	Module	Ratio	Spatial		Object		Goal		Long	
			H	**L**	**H**	**L**	**H**	**L**	**H**	**L**
OpenVLA	SigLIP Attn	1.00	47.0	–	51.5	–	40.8	–	27.2	–
	SigLIP FFN	1.00	70.0	–	73.8	–	64.2	–	38.5	–
	Projector FFN	0.30	0.0	**54.0**	0.0	**57.4**	0.0	**50.8**	0.0	**26.5**

C Algorithmic Specifications for Joint Pruning

To ensure reproducibility, we provide the detailed configurations for the *Light, Moderate,* and *Aggressive* settings used in our multi-module joint pruning scheme.

As shown in Table 12, different modules are assigned specific pruning ratios and selection criteria based on the divergence signals ΔW observed during VLM-to-VLA adaptation.

Table 12: Pruning configurations for OpenVLA and $\pi_{0.5}$. Ratios indicate the fraction of parameters removed. Selection criteria: **H** (High-diff), **L** (Low-dif).

	OpenVLA			$\pi_{0.5}$		
Module	**Light**	**Moderate**	**Aggressive**	**Light**	**Moderate**	**Aggressive**
LLM-Attn	0.125 (H)	0.125 (H)	0.125 (H)	0.2 (H)	0.2 (H)	0.2 (H)
LLM-FFN	0.1 (H)	0.2 (H)	0.3 (H)	0.2 (H)	0.3 (H)	0.5 (H)
SigLIP-Attn	0.125 (H)	0.125 (H)	0.125 (H)	0.2 (H)	0.2 (H)	0.2 (H)
SigLIP-FFN	0.1 (L)	0.1 (L)	0.1 (L)	0.4 (H)	0.2 (H)	0.2 (H)
DINOv2-Attn	0.0625 (L)	0.0625 (L)	0.0625 (L)	-	-	-
DINOv2-FFN	0.1 (H)	0.1 (H)	0.1 (H)	-	-	-

D Generalization Across Models and Benchmarks

To further validate the generality of our adaptation-based pruning criterion, we evaluate π_0 on five RoboTwin2.0 tasks by pruning the highest- and lowest-ΔW 10% channels in the LLM FFN layers.

As shown in Table 13, pruning high-ΔW channels consistently outperforms pruning low-ΔW channels across all tasks, demonstrating that adaptation-induced parameter divergence remains an effective indicator of parameter importance across different VLA models and benchmarks.

Table 13: Generalization evaluation on RoboTwin2.0 using π_0. We compare pruning the highest- and lowest-ΔW 10% channels in the LLM FFN layers.

Simulation Task	**Original**	**High-ΔW**	**Low-ΔW**
Beat Block Hammer	41	35	8
Move Can Pot	55	43	11
Shake Bottle	90	80	18
Place Phone Stand	32	27	6
Rotate QRcode	62	52	10
Average	**56.0**	**47.4**	**10.6**