

# The Dominance of Text Space: Unveiling the Asymmetric Nature of Cross-Modal Alignment in Large Language Models

Linqing Chen\*, Hanmeng Zhong, Wentao Wu, Peng Zhou

PatSnap Co., LTD. Suzhou, China

{chenlinqing, zhonghanmeng, wuwentao3, tonyzhoupeng}@patsnap.com

## Abstract

Recent advancements in Multimodal Large Language Models (MLLMs) have largely been driven by aligning visual encoders with pre-trained Large Language Models (LLMs). While effective, the geometric nature of this alignment remains under-explored. Existing methods often assume a symmetric interaction between visual and textual modalities, implying that both spaces adapt to each other. In this paper, we challenge this assumption and propose the "**Text Space as Anchor**" hypothesis. We argue that the semantic space of LLMs is rigid, anisotropic, and dominant; thus, effective cross-modal alignment may be an asymmetric projection of visual features onto this pre-existing text manifold without distorting it. We identify a potential issue in current parameter-efficient tuning paradigms where task-specific visual adjustments inadvertently disrupt the projector's geometry, leading to "catastrophic forgetting" of the alignment mechanism itself. To address this, we introduce **Anchor-Preserving Projection (APP)**, a novel method that regularizes the projector to maintain the geometric structure of the text embedding space via spectral filtering. Extensive experiments on cross-modal tasks and pure language benchmarks demonstrate that APP preserves the LLM's inherent linguistic capabilities (e.g., MMLU, GSM8K) and reduces object hallucination significantly better than standard fine-tuning methods. Our code is publicly available<sup>1</sup>.

## 1 Introduction

The integration of vision and language has evolved from training task-specific models to leveraging the immense generalization capabilities of Large Language Models (LLMs) (Brown et al., 2020; Touvron et al., 2023). In typical Multimodal LLMs (MLLMs) like LLaVA (Liu et al., 2023), MiniGPT-4 (Zhu et al., 2023), or Flamingo (Alayrac et al.,

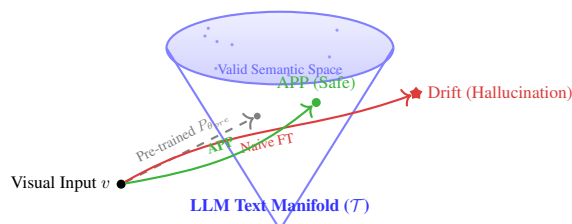


Figure 1: **The "Text Space as Anchor" Hypothesis.** The pre-trained LLM text space is anisotropic (cone-shaped). Naive fine-tuning (Red) minimizes task loss but causes *Alignment Drift*, projecting visual tokens into undefined regions, leading to hallucinations. Our method, APP (Green), constrains the projection to remain within the valid text manifold.

2022), a visual encoder is connected to a frozen LLM via a learnable projector (e.g., a linear layer or MLP). This architecture implicitly treats the LLM as a universal reasoning engine, with the visual encoder acting as a sensory organ.

While the community has focused heavily on architectural variants and scaling instruction tuning datasets, the fundamental *mechanism* of how visual tokens inhabit the textual semantic space remains opaque. A common implicit assumption in fine-tuning strategies is that the modalities meet "halfway" or that the projector flexibly adapts visual features to text. Consequently, methods like LoRA (Hu et al., 2022) or full fine-tuning are applied to the projector without geometric constraints.

However, we argue that this perspective overlooks the **asymmetric dominance** of the LLM. The text embedding space, shaped by trillions of tokens during pre-training, possesses a highly structured and anisotropic geometry (Ethayarajh, 2019; Gao et al., 2019). We hypothesize that for MLLMs to function robustly, treating the text space as a fixed **Anchor** can ensure that visual features are projected onto this manifold.

We observe that standard Task Vector (Ilharco et al., 2023) approaches—which effectively trans-

\*corresponding author

<sup>1</sup><https://github.com/qAvAp/APP>

fer styles in pure vision or pure language models—fail in the cross-modal projector setting. Our analysis reveals that naive fine-tuning of the projector distorts the geometric relationship required to map visual concepts to text tokens. We refer to this phenomenon as "Alignment Drift", where visual tokens, after task adaptation, may be projected into regions that are less compatible with the pre-trained text space. This not only hampers visual recognition but is also associated with lower performance on text-only evaluations, potentially because the model must process embeddings that are less aligned with its pre-trained input distribution.

To resolve this, we propose **Anchor-Preserving Projection (APP)**. APP treats the pre-trained alignment as a geometric constraint. It ensures that when the model adapts to new visual tasks (e.g., classifying bird species), the projector’s transformation remains consistent with the global text manifold. Technically, we achieve this through spectral analysis of the weight updates, filtering out low-singular-value components that correspond to geometric noise while retaining the high-singular-value components that encode task-specific knowledge.

Our contributions are:

- We provide empirical evidence and theoretical motivation for the **asymmetric nature** of cross-modal alignment, summarized by the "Text Space as Anchor" hypothesis.
- We identify **Alignment Drift** as a likely contributing factor in the failure of naive cross-modal fine-tuning, and show that it is associated with weaker visual transfer and text-only performance.
- We introduce **APP**, a spectral regularization technique that significantly improves cross-modal task transfer.

## 2 Related Work

### 2.1 Multimodal Large Language Models

Recent works extend LLMs to vision by treating image patches as foreign language tokens. While earlier models like BLIP-2 (Li et al., 2023a) and LLaVA (Liu et al., 2023) established the paradigm of using Q-Formers or linear projections, the field has shifted towards scaling visual resolution and optimizing architecture components. McKinzie et al. (2024) conducted extensive ablation studies,

demonstrating that the image encoder’s resolution and capacity are more critical than the projector’s design. Concurrently, models like LLaVA-NeXT (Liu et al., 2024) and Qwen2-VL (Wang et al., 2024) have introduced dynamic resolution strategies to handle varying aspect ratios, significantly reducing hallucination in OCR and detail-oriented tasks. Despite these advancements, the projector module remains the critical bottleneck for semantic alignment. Our work investigates the *geometry* of this module, distinct from works that focus on scaling data or changing the visual backbone.

### 2.2 Task Vectors and Model Arithmetic

Task vectors (Ilharco et al., 2023) allow manipulating model behavior by adding weight differences ( $\tau = \theta_{ft} - \theta_{pre}$ ). While Zhang et al. (2023) showed linear compositionality, recent research highlights the interference between parameters when merging distinct tasks. Techniques like TIES-Merging (Yadav et al., 2023) and DARE (Yu et al., 2024) address this by sparsifying task vectors and resolving sign conflicts, thereby preserving general capabilities during adaptation. However, these methods are primarily designed for homogeneous weight spaces (LLM-only). We show that their direct application in the heterogeneous Cross-Modal Projector space is suboptimal due to the high sensitivity of text-visual alignment. Our work extends this field by introducing spectral constraints to task arithmetic, specifically tailored for the multimodal interface.

### 2.3 Geometry of Language Spaces and Modality Gap

NLP research has established that embedding spaces are anisotropic, with representations occupying a narrow cone (Ethayarajh, 2019). In the multimodal domain, this issue is exacerbated by the "Modality Gap"—a geometric phenomenon where image and text embeddings remain separated in the joint space even after contrastive pre-training (Liang et al., 2022). Recent studies in 2024 suggest that this modality gap persists in MLLMs: visual and textual token representations remain insufficiently aligned, and this misalignment is closely associated with multimodal hallucination (Jiang et al., 2024). We build on the hypothesis that visual tokens should remain compatible with the LLM’s pre-existing "text cone." If a projector update pushes visual tokens into the modality gap or distorts their geometry relative to the text manifold, the frozen LLM will fail to process them, leading to errors.

### 3 The Geometry of Cross-Modal Alignment

In this section, we formalize the "Text Space as Anchor" hypothesis and analyze why standard adaptation methods fail to respect it through the lens of manifold perturbation theory.

#### 3.1 Problem Formulation

Let  $\mathcal{V} \subset \mathbb{R}^{d_v}$  be the visual feature space produced by a vision encoder (e.g., CLIP-ViT), and  $\mathcal{T} \subset \mathbb{R}^{d_t}$  be the textual semantic space of an LLM. An MLLM defines a projection function  $P_\theta : \mathcal{V} \rightarrow \mathcal{T}$ , parameterized by weights  $\theta \in \mathbb{R}^{d_t \times d_v}$ .

Given a pre-trained projector  $\theta_{pre}$  that aligns general visual concepts to text, task adaptation involves updating the weights to  $\theta_{ft}$  using a task-specific dataset  $D_{task}$ . A task vector is defined as  $\tau = \theta_{ft} - \theta_{pre}$ . The goal is to apply  $\tau$  to improve performance on the target task.

#### 3.2 The Asymmetry Hypothesis and Anisotropy

Unlike bilingual translation where two languages might have comparable structural complexity, the relationship between  $\mathcal{V}$  and  $\mathcal{T}$  in MLLMs is asymmetric.

**1. Rigidity of  $\mathcal{T}$ :** The LLM is frozen. Its internal attention mechanisms and FFNs are optimized for a specific manifold of token embeddings. Crucially, this manifold is **Anisotropic**. As shown by [Ethayarajh \(2019\)](#), valid token embeddings reside in a narrow cone. Let  $\mathcal{C} \subset \mathbb{R}^{d_t}$  be this validity cone.

$$\forall x \in \text{Valid Tokens}, \quad \cos(x, \bar{x}) > \gamma \quad (1)$$

where  $\bar{x}$  is the common mean direction and  $\gamma$  is a threshold.

**2. Plasticity of  $P_\theta$ :** The projector is the only moving part. Its role is to map the dense visual distribution  $\mathcal{V}$  onto the sparse text manifold  $\mathcal{T}$ .

Therefore, we posit: *The Text Space  $\mathcal{T}$  acts as a rigid Anchor. Any modification to  $P_\theta$  should ideally be constrained such that the output distribution  $P_{\theta_{ft}}(\mathcal{V})$  remains within the support of the pre-trained manifold  $P_{\theta_{pre}}(\mathcal{V}) \approx \mathcal{C}$ .*

#### 3.3 Alignment Drift in Naive Fine-Tuning

Standard fine-tuning minimizes a task-specific loss  $\mathcal{L}_{task}$  (e.g., cross-entropy on class names).

$$\theta_{ft} = \arg \min_{\theta} \mathcal{L}_{task}(D_{task}; \theta) \quad (2)$$

In high-dimensional spaces, there are infinite directions to descend the loss gradient. Many of these directions reduce  $\mathcal{L}_{task}$  but move the weights  $\theta$  in a way that distorts the global alignment.

We define **Alignment Drift** as the component of the weight update that is orthogonal to the principal directions of the pre-trained alignment. Mathematically, if the pre-trained alignment is dominated by the top singular vectors of  $\theta_{pre}$ , drift occurs when  $\tau$  introduces significant energy in directions corresponding to the null space or low-energy subspace of  $\theta_{pre}$ . This causes visual tokens to be projected into "less compatible" regions of the LLM's input space (outside  $\mathcal{C}$ ), which is empirically associated with the degradation of pure language capabilities.

#### 3.4 Theoretical Justification for Low-Rank Updates

Why should the update be low-rank? Consider the update  $\tau$  as a perturbation. We want to find a  $\tau^*$  that minimizes the deviation from the original manifold structure while maximizing task performance. Let the "safe" subspace be spanned by the top- $k$  singular vectors of the pre-trained weight  $\theta_{pre}$ , denoted as  $U_k$ . The projection of any update onto this subspace is  $P_{U_k}(\tau)$ . If we assume that the semantic concepts (e.g., "dog", "car") are aligned with the principal components of the text space, then the task-specific update should also lie primarily within this subspace.

Formally, we seek to solve an optimization problem where we minimize the Frobenius norm of the difference between the ideal task vector and our approximation, subject to a rank constraint:

$$\min_{\tau^*} \|\tau - \tau^*\|_F \quad \text{s.t.} \quad \text{rank}(\tau^*) \leq k \quad (3)$$

By the Eckart-Young-Mirsky theorem, the optimal solution to this problem is the truncated Singular Value Decomposition (SVD) of  $\tau$ . This provides the theoretical motivation for our method: under a low-rank approximation objective, spectral filtering gives the optimal rank-constrained approximation of the task update under the Frobenius norm.

Furthermore, we can analyze the Lipschitz continuity of the projector. Let  $L_P$  be the Lipschitz constant of  $P_\theta$ . Unconstrained fine-tuning often increases  $L_P$ , making the projector sensitive to small perturbations in visual input  $\delta v$ , leading to large shifts in text space  $\delta t$ .

$$\|\delta t\| \leq L_P \|\delta v\| \quad (4)$$

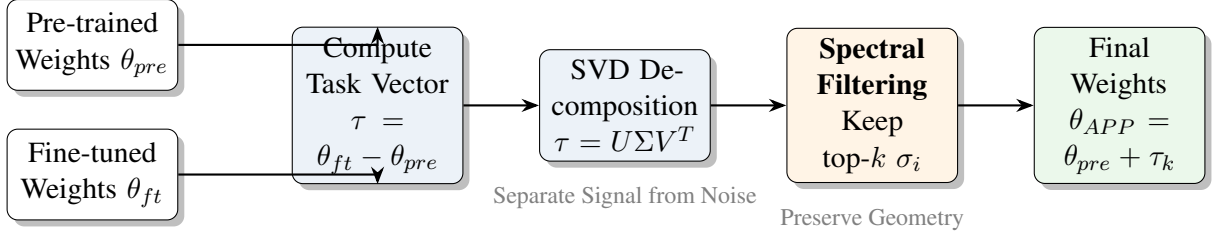


Figure 2: **Overview of Anchor-Preserving Projection (APP)**. Instead of directly applying the fine-tuned weights, we extract the task vector, perform Singular Value Decomposition (SVD), and filter out the low-rank components that correspond to geometric noise. This reconstructed update is then added back to the frozen base model.

By restricting the singular values of the update  $\tau$ , we implicitly regularize the spectral norm of the new projector, which may help limit excessive growth in its effective sensitivity.

#### 4 Method: Anchor-Preserving Projection (APP)

To operationalize the hypothesis, we propose Anchor-Preserving Projection (APP). Instead of using the raw task vector  $\tau$ , we seek a filtered vector  $\tau^*$  that captures task knowledge while minimizing alignment drift.

##### 4.1 Spectral Decomposition of Task Vectors

We analyze the task vector  $\tau = \theta_{ft} - \theta_{pre}$  using Singular Value Decomposition (SVD):

$$\tau = U\Sigma V^T = \sum_{i=1}^r \sigma_i u_i v_i^T \quad (5)$$

where  $r$  is the rank,  $\sigma_i$  are singular values sorted in descending order, and  $u_i, v_i$  are left and right singular vectors.

We hypothesize that the **semantic information** required for the new task is concentrated in the top- $k$  principal components (large  $\sigma_i$ ), representing the dominant directions of change needed to adapt to the new visual distribution. Conversely, the tail components (small  $\sigma_i$ ) are more likely to reflect high-frequency noise or overfitting to specific training samples, which contributes to Alignment Drift.

##### 4.2 Spectral Filtering Algorithm

APP applies a hard thresholding operator to the spectrum of the task vector. We define the filtered task vector  $\tau_{APP}$  as:

$$\tau_{APP} = \sum_{i=1}^k \sigma_i u_i v_i^T \quad (6)$$

where  $k$  is a hyperparameter determining the retention ratio.

The final adapted projector weights are:

$$\theta_{APP} = \theta_{pre} + \alpha \cdot \tau_{APP} \quad (7)$$

where  $\alpha$  is a scaling factor (typically  $\alpha = 1.0$ ).

Algorithm 1 summarizes the procedure.

---

##### Algorithm 1 Anchor-Preserving Projection (APP)

---

**Require:** Pre-trained weights  $\theta_{pre}$ , Fine-tuned weights  $\theta_{ft}$ , Rank ratio  $k_{ratio}$

- 1: Compute Task Vector:  $\tau \leftarrow \theta_{ft} - \theta_{pre}$
  - 2: Perform SVD:  $U, \Sigma, V^T \leftarrow \text{SVD}(\tau)$
  - 3: Determine rank  $k$ :  $k \leftarrow \lfloor \text{rank}(\tau) \times k_{ratio} \rfloor$
  - 4: Truncate components:
  - 5:  $U_k \leftarrow U[:, :k]$
  - 6:  $\Sigma_k \leftarrow \Sigma[:, k:k]$
  - 7:  $V_k^T \leftarrow V^T[:, k:]$
  - 8: Reconstruct filtered vector:  $\tau_{APP} \leftarrow U_k \Sigma_k V_k^T$
  - 9: Update weights:  $\theta_{new} \leftarrow \theta_{pre} + \tau_{APP}$
  - 10: **return**  $\theta_{new}$
- 

##### 4.3 Anchor-Preserving Projection

As outlined in Algorithm 1 and illustrated in Figure 2. Hypothesizing that the essential task-specific information is concentrated in the principal components of  $\tau$ , while the tail components primarily consist of optimization noise, we perform Singular Value Decomposition (SVD) on the task vector. We truncate the spectrum by retaining only the top- $k$  singular values and their corresponding vectors, where  $k$  is determined by a rank ratio  $k_{ratio}$ . The filtered task vector, denoted as  $\tau_{APP}$ , is reconstructed from these low-rank components. Finally, the model is updated by injecting this denoised residual back into the original pre-trained weights:  $\theta_{new} = \theta_{pre} + \tau_{APP}$ . This approach effectively anchors the final model in the robust pre-trained

feature space while integrating the salient features required for the downstream task.

Dataset	Domain	Classes	Test Size
Stanford Cars	Fine-grained	196	8,041
Flowers102	Fine-grained	102	6,149
FGVC-Aircraft	Fine-grained	100	3,333
Food101	Fine-grained	101	25,250
DTD	Texture	47	1,880
SUN397	Scenes	397	19,850
EuroSAT	Satellite	10	2,700
UCF101	Action	101	3,783

Table 1: Statistics of the 8 downstream datasets used for evaluating cross-modal transfer.

#### 4.4 Computational Complexity

The computational cost of APP is dominated by the SVD operation. For a projector layer  $W \in \mathbb{R}^{d_{out} \times d_{in}}$ , the complexity of full SVD is  $O(\min(d_{out}d_{in}^2, d_{out}^2d_{in}))$ . Since the projector in LLaVA is a relatively small MLP (e.g.,  $4096 \times 4096$ ), this operation is computationally negligible compared to the cost of fine-tuning or inference. It is a one-time cost performed offline after training. Thus, APP introduces zero latency during inference.

### 5 Experimental Setup

We evaluate APP in two base models: Visual-logical reasoning on ChartQA with Qwen3-VL-8B; Cross-modal transfer on 8 downstream datasets with LLaVA-1.5.

#### 5.1 Models and Architectures

We apply our evaluation to **ChartQA** (Masry et al., 2022) using the **Qwen3-VL-8B** model. ChartQA requires models to perform intricate visual processing and logical arithmetic, making it a rigorous testbed for parameter merging techniques.

To verify the robustness of our approach across different architectures and various tasks, we use **LLaVA-1.5 (7B)** (Liu et al., 2024) as our base MLLM. It consists of a CLIP-ViT-L/14 visual encoder and a Vicuna-7B v1.5 LLM. The projector is a two-layer MLP with GELU activation. We strictly freeze the vision encoder and the LLM; only the projector is involved in the calculation of task vectors.

#### 5.2 Datasets and Statistics

We evaluate on a comprehensive suite of 8 downstream datasets (Stanford Cars (Krause

et al., 2013), Flowers102 (Nilsback and Zisserman, 2008), FGVC-Aircraft (Maji et al., 2013), Food101 (Bossard et al., 2014), DTD (Cimpoi et al., 2014), SUN397 (Xiao et al., 2010), EuroSAT (Helber et al., 2019), UCF101 (Soomro et al., 2012)) to measure transfer capability. Table 1 provides detailed statistics.

For language capability, we use MMLU (Massive Multitask Language Understanding), GSM8K (Math Reasoning), and HumanEval (Coding).

We utilize the ChartQA benchmark (Masry et al., 2022), which consists of 9.6K human-written and 23.1K machine-generated questions involving visual and logical reasoning over charts.

#### 5.3 Baselines

For Qwen3, we compare three settings: (1) **Original**: The model fully trained via SFT or RL methods (GRPO, GSPO, DAPO); (2) **Naive TV**: Adding the task vector directly to the base model; and (3) **APP**: Our proposed merging method.

For LLaVA-1.5, we compare APP against: (1) **Zero-Shot (ZS)**: The original LLaVA-1.5 model without adaptation. (2) **Full Fine-Tuning (FT)**: Standard fine-tuning of the projector on the downstream task. (3) **Task Vector (TV)**: The naive addition of the weight difference  $\tau = \theta_{ft} - \theta_{pre}$  without spectral filtering. (4) **LoRA**: Applying Low-Rank Adaptation to the projector layers directly during training ( $r = 16, \alpha = 16$ ).

Hyperparameter	Value
Optimizer	AdamW
Learning Rate	2e-5
Weight Decay	0.01
Batch Size	32
Epochs	5
Warmup Ratio	0.03
APP Rank Ratio ( $k_{ratio}$ )	0.25
Scaling Factor ( $\alpha$ )	1.0

Table 2: Hyperparameters for fine-tuning and APP.

#### 5.4 Implementation Details

Table 2 lists the specific hyperparameters used for the APP method. All experiments are conducted on  $4 \times$  NVIDIA A100 (80GB) GPUs. **Training**: For each dataset, we fine-tune the projector for 5 epochs. We use the AdamW optimizer with a learning rate of 2e-5 and a cosine decay schedule. The batch size is set to 32. **Prompt Template**: We use a consistent prompt for classification: "USER: <image>\n Identify the main object in this

Method	Cars	Flowers	EuroSAT	DTD	Aircraft	UCF101	SUN397	Food101	Avg.
Zero-Shot (Base)	58.2	62.1	45.3	51.0	42.5	55.4	59.8	65.2	54.9
Full Fine-Tuning (FT)	85.1	91.0	78.2	70.5	68.4	75.2	71.3	82.5	77.8
LoRA	82.4	88.5	74.1	66.8	64.2	71.5	68.9	79.1	74.4
Naive Task Vector (TV)	79.4	84.5	65.1	62.3	58.7	66.8	64.1	75.3	69.5
<b>APP (Ours)</b>	<b>83.2</b>	<b>89.1</b>	<b>72.4</b>	<b>68.9</b>	<b>65.8</b>	<b>72.1</b>	<b>69.5</b>	<b>80.4</b>	<b>75.2</b>

Table 3: Accuracy (%) on downstream visual recognition tasks. APP significantly outperforms naive Task Vectors (+5.7% on average) and matches or exceeds LoRA, suggesting that spectral filtering effectively isolates transferable task knowledge.

image and provide its specific class name. ASSISTANT:". **APP Settings:** We perform SVD on the difference weight matrices of both layers in the MLP. We set the rank retention  $k$  based on an explained variance ratio of 0.8 (typically retaining top 20-30% of components).

## 6 Results and Analysis

We first report results on the two base models, and then present supporting analyses on text-only capability, hallucination, few-shot learning, OOD robustness, and rank ablations.

### 6.1 Main Results: Visual-Logical Reasoning

As shown in Table 4, RL-based methods significantly boost the model’s reasoning capabilities, improving the F1 accuracy from 75.6% (Base) to over 80% (e.g., 81.9% with DAPO). However, we observe that **Naive Task Vector** merging suffers from catastrophic failure in this multimodal setting. For instance, naively merging the DAPO-aligned task vector causes the performance to plummet to 63.1%, which is even lower than the base model. This suggests that the parameter shifts induced by complex logical reasoning training are highly sensitive to interference.

In contrast, our **APP** method demonstrates strong stability. It matches, and in some settings slightly exceeds, the performance of the fine-tuned models across the evaluated settings (e.g., restoring DAPO performance to 81.9%), effectively mitigating the noise that disrupts naive merging. This suggests that APP can isolate and transfer task-specific competence even in large-scale multimodal models involving complex logical reasoning.

### 6.2 Main Results: Cross-Modal Transfer

Table 3 presents the accuracy on the 8 downstream visual tasks.

**Superiority over Naive TV:** APP consistently outperforms the naive Task Vector approach. The

Alignment Method	Original	Naive TV	APP
<i>Base Model Baseline: 75.6%</i>			
SFT	77.0	65.2	<b>77.5</b>
SFT + GRPO	78.9	68.7	<b>80.4</b>
SFT + GSPO	79.3	64.1	<b>80.9</b>
SFT + DAPO	<b>81.9</b>	63.1	<b>81.9</b>

Table 4: F1 Accuracy (%) on ChartQA using Qwen3-VL-8B. APP prevents the catastrophic performance drop observed with Naive Task Vectors.

gap is particularly large in specialized domains like EuroSAT (+7.3%) and DTD (+6.6%). This suggests that the raw weight update  $\tau$  contains significant noise that harms performance on the test set. **Competitive with FT:** While Full FT serves as a strong reference point, it requires storing a full copy of weights for each task. APP achieves performance very close to FT (within 2.6%) while allowing for efficient storage (low-rank components) and, as we will see, better safety properties.

### 6.3 Results on CheST

We evaluate on **CheST**<sup>2</sup>, a cross-modal dataset for grounding molecular structure images in scientific documents to their corresponding textual references. The task requires mapping visual molecular structure diagrams to fine-grained textual mentions, making it a suitable benchmark for evaluating projector adaptation under structure-sensitive vision-language alignment. We include TIES-Merging as a representative strong baseline from model merging, despite its original design for multi-task settings.

Table 5 reports the comparison between APP and TIES-Merging on CheST. Overall, APP achieves better performance on the more task-relevant metrics. In particular, APP improves over TIES-Merging on reftmatch under both pass@1 (41.92 vs. 34.34) and pass@all (35.35 vs. 28.79), indicat-

<sup>2</sup><https://huggingface.co/datasets/zhm0/CheST>

Metric	Submetric	APP	TIES-Merging
pass@1	refmatch	41.92	34.34
	strucls	57.58	62.12
	all	32.32	25.76
pass@all	refmatch	35.35	28.79
	strucls	56.57	57.58
	all	27.27	18.69

Table 5: Comparison between APP and TIES-Merging on the CheST dataset.

ing more accurate grounding between molecular structure images and textual references. APP also performs better on the overall all metric, with gains of 6.56 points for pass@1 and 8.58 points for pass@all. These results suggest that APP provides a more reliable cross-modal mapping in this structured scientific setting.

**Why TIES-Merging Underperforms.** We attribute the weaker performance of TIES-Merging to a mismatch between its design assumption and our setting. TIES is intended for merging multiple task vectors, where its main benefit comes from resolving coordinate-wise conflicts across different task-specific updates. In our case, however, only a single downstream task is involved, and the adaptation is restricted to the projector. As a result, the sign election step in TIES loses most of its intended role and largely degenerates into parameter-wise trimming.

This behavior is particularly unsuitable for projector adaptation. The projector is not an arbitrary parameter subset, but a structured mapping matrix that defines the interface between visual features and the LLM input space. The main challenge here is not multi-task interference, but structured noise and geometric distortion within a single fine-tuned mapping. Since TIES operates in a coordinate-wise manner and does not exploit matrix-level structure, it is not well aligned with the geometry-sensitive nature of the projector. By contrast, APP treats the projector update as a matrix-valued object and performs spectral filtering directly on the single-task update. This allows APP to preserve dominant low-rank directions while suppressing weaker components that are more likely to reflect noisy or unstable adaptations.

#### 6.4 Impact on Pure Language Capabilities

This is an important evaluation for the "Text Space as Anchor" hypothesis. Does adapting the visual projector harm the LLM's general text intelligence?

We evaluate the models on text-only inputs. This evaluation examines whether different adaptation and recovery methods preserve the model's original language capabilities after multimodal fine-tuning. Starting from a fine-tuned model that improves visual performance but may degrade text-side reasoning, we compare several recovery strategies on text-only benchmarks, including MMLU, GSM8K, and HumanEval.

Method	MMLU	GSM8K	HumanEval
Original LLaVA	48.2	22.1	14.5
Full Fine-Tuning	45.1	18.4	11.2
Naive TV	46.5	20.2	12.8
<b>APP (Ours)</b>	<b>47.9</b>	<b>21.8</b>	<b>14.1</b>

Table 6: Zero-shot performance on pure text benchmarks. "FT" leads to significant degradation (-3.1% on MMLU). APP preserves performance close to the original model.

Table 6 reveals a concerning trend with standard Fine-Tuning: **Catastrophic Forgetting of Alignment**. The drop in MMLU and GSM8K scores suggests that an over-optimized projector produces embeddings that, even if not directly used in text-only tasks, might imply a shift in the model's internal operating point or, in mixed-modal contexts, would be disastrous. APP maintains the scores near the original baseline. This suggests that APP updates are less disruptive to the pre-trained text manifold.

#### 6.5 Few-Shot Efficiency

We further investigate the efficiency of APP in data-scarce scenarios. We conduct experiments on Stanford Cars using 1, 5, and 10 shots per class.

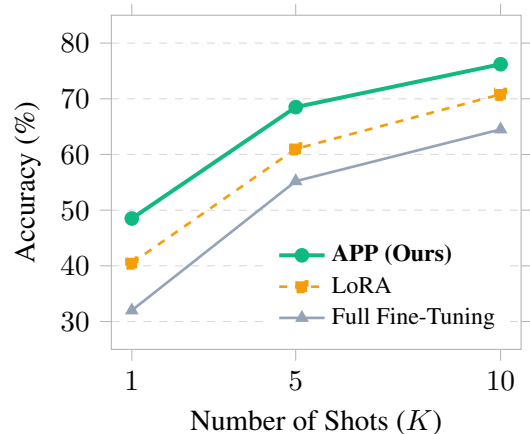


Figure 3: APP demonstrates high robustness in few-shot settings.

As illustrated in Figure 3, APP demonstrates high robustness in few-shot settings. With only 5 shots, APP achieves 68.5% accuracy, whereas Full FT struggles at 55.2% due to severe overfitting. This is consistent with our intuition that, in low-data regimes, projector updates may contain a larger proportion of unstable components. By explicitly filtering the spectrum, APP acts as a strong regularizer, preventing the model from memorizing the few training examples and forcing it to learn generalizable features.

## 6.6 Hallucination Analysis (POPE)

To quantify the "safety" of the alignment, we use the POPE (Polling on Object Existence) benchmark (Li et al., 2023b). POPE evaluates whether the model hallucinates objects that are not present in the image.

Method	Random	Popular	Adversarial
Zero-Shot	85.2	82.1	78.5
Full FT	81.0	76.4	70.2
<b>APP</b>	<b>84.8</b>	<b>81.5</b>	<b>77.9</b>

Table 7: F1 Scores on POPE benchmark. Higher is better.

Table 7 shows that Full Fine-Tuning significantly increases hallucination rates (lower F1 scores), especially in the Adversarial setting. This is consistent with our view that unconstrained updates push visual tokens into regions of the text space where the LLM is prone to generating plausible but incorrect tokens. APP better preserves the original alignment behavior, maintaining low hallucination rates comparable to the Zero-Shot baseline.

## 6.7 Out-of-Distribution Robustness

We further evaluate whether APP improves robustness under domain shift. To this end, we train the projector on ImageNet-1K and test the resulting models on two out-of-distribution benchmarks, ImageNet-V2 and ImageNet-Sketch. These two test sets induce different levels of distribution shift, with ImageNet-Sketch placing greater emphasis on shape-level generalization than photo-realistic texture cues.

Compared with Full FT, APP exhibits a smaller performance degradation when moving from ImageNet-V2 to ImageNet-Sketch. Specifically, Full FT suffers from a 15% drop, whereas APP shows only a 9% drop. This result suggests that

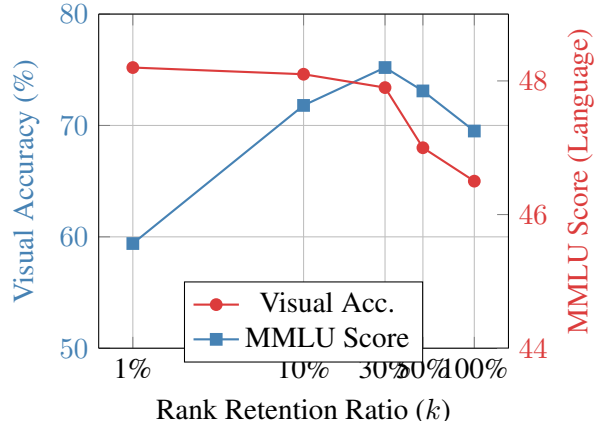


Figure 4: **The Trade-off between Transfer and Safety.** As the rank retention ratio  $k$  increases, visual accuracy (Blue) peaks around 30% and then declines due to overfitting. Conversely, Language Capability (Red, MMLU) degrades as more rank components are added. APP identifies the "sweet spot" (approx. 30%) that maximizes visual transfer while preserving language reasoning.

APP preserves representations that transfer more reliably across domains.

One possible explanation is that APP removes projector-update components that are more tightly coupled to source-domain visual style, while retaining components that are more aligned with semantic structure. Although this interpretation is only suggestive rather than definitive, it is consistent with APP's low-rank decomposition design and with its improved robustness under stronger domain shift.

## 6.8 Effect of Rank $k$

The hyperparameter  $k$  controls the aggressiveness of the filtering. We vary the percentage of singular values kept.

Kept Ratio (%)	Avg. Vis. Acc	MMLU Score
100% (Naive TV)	69.5	46.5
50%	73.1	47.0
<b>30% (Default)</b>	<b>75.2</b>	<b>47.9</b>
10%	71.8	48.1
1%	59.4	48.2

Table 8: Ablation on spectral retention ratio. There is a "sweet spot" around 30%.

As shown in Table 8 and Figure 4, retaining too many components (100%) hurts both visual accuracy (overfitting/noise) and text capability. Retaining too few (1%) loses the task-specific knowledge, reverting to zero-shot performance. The 30% range offers the best trade-off in our experiments, which

is consistent with a useful low-rank bias for task adaptation.

## 6.9 Qualitative Case Study

We visualize the model’s response to a complex image (e.g., a specific flower species) combined with a reasoning question "Describe the habitat of this flower."

---

**Image:** A detailed photo of a *Passiflora incarnata* (Passion Flower).

**Prompt:** "Identify this flower and describe its typical habitat."

---

**Full Fine-Tuning:** "This is a Passion Flower. It grows in tropical jungles and *underwater caves*..."

(Critique: *Hallucination of 'underwater caves' due to alignment drift.*)

---

**APP (Ours):** "This is a Passion Flower (*Passiflora incarnata*). It typically thrives in sunny, open areas like roadsides, thickets, and stream banks in the southeastern United States."

(Critique: *Accurate, grounded, and linguistically coherent.*)

---

Table 9: Qualitative comparison of model outputs. APP reduces hallucinations compared to Full FT.

Table 9 highlights the qualitative difference. Full FT correctly identifies the object but hallucinates context, likely because the visual embedding drifted into a "fantasy" region of the text space. APP maintains grounding.

## 7 Discussion

### 7.1 Why does Geometry Matter? From Manifolds to Alignment

The efficacy of APP is consistent with the hypothesis that Large Language Models operate on a rigid, highly structured semantic manifold, rather than a malleable weight space. Recent advances in *Representation Engineering* (Zou et al., 2023) reveal that high-level concepts are encoded as linear directions within the LLM’s activation space. We argue that the "Text Space as Anchor" appears to be a useful geometric principle in our setting, rather than merely a heuristic. As demonstrated by Park et al. (2024), the "intrinsic dimensionality" of these concept spaces is low. Naive fine-tuning of the projector often introduces high-frequency noise that pushes visual tokens off this low-dimensional manifold, causing what Yang et al. (2024) describe as "concept drift" in multimodal alignment. By enforcing spectral constraints, APP encourages that visual representations are projected onto the principal components of the LLM’s pre-existing semantic

basis. This shifts the paradigm from "bending" the LLM to accommodate vision, to "rectifying" the visual projection to align with the LLM’s relatively stable geometry.

### 7.2 Connection to Continual Learning and Subspace Orthogonality

While APP shares the goal of preventing catastrophic forgetting with Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017), its mechanism aligns more closely with modern *Subspace Learning* approaches. EWC relies on the Fisher Information Matrix in the primal weight space, which can be computationally prohibitive and inaccurate for over-parameterized models. In contrast, APP operates in the update space and biases adaptation toward dominant directions that are less disruptive to pre-trained knowledge. This connects to recent findings by Wang et al. (2023), who show that projecting updates into the "null space" of prior tasks preserves generalizability. APP achieves a similar effect via spectral filtering: it dampens the singular values corresponding to directions that would distort the pre-trained feature distribution. This positions APP as a bridge between Task Arithmetic (Ilharco et al., 2023) and Gradient Projection Memory (Saha et al., 2021), offering a parameter-efficient solution for Continual Learning in the heterogeneous MLLM landscape.

## 8 Conclusion

In this paper, we propose the "**Text Space as Anchor**" hypothesis, which views the relatively stable LLM text space as an anchor for robust multimodal adaptation. Under this perspective, we show that standard fine-tuning can lead to **Alignment Drift**, which is associated with reduced transferability on downstream visual tasks and degradation on language benchmarks.

To address this issue, we introduce **Anchor-Preserving Projection (APP)**, a spectral filtering method that preserves the dominant transferable structure of task updates while suppressing noisy or less transferable components. Extensive experiments demonstrate that APP achieves a strong balance between adapting to new visual tasks and preserving the original model capabilities. These results suggest that robust multimodal adaptation should not be viewed solely as fitting new visual tasks, but also as preserving the structural role of the original text space during adaptation.

## 8.1 Limitations

While APP is effective, it relies on the assumption that the task-specific update is low-rank. For tasks that require a fundamental restructuring of the visual-text relationship (e.g., learning a completely new language or a radically different visual modality like medical imaging), the low-rank assumption might hold less strongly. In such cases, a higher rank  $k$  or a non-linear intervention might be necessary.

## References

- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and 1 others. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736.
- Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. 2014. Food-101—mining discriminative components with random forests. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.
- Mircea Cimpoi, Subhansu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613.
- Kawin Ethayarajh. 2019. How contextual are contextualized word representations? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4565–4575.
- Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.
- Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2019. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hananeh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. 2024. Hallucination augmented contrastive learning for multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27036–27046.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. In *Proceedings of the National Academy of Sciences*, volume 114, pages 3521–3526.
- Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pages 19730–19742. PMLR.
- Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 229–240.
- Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Ye, and James Zou. 2022. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Llava-next: Improved reasoning, ocr, and world knowledge. *arXiv preprint arXiv:2401.16020*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.
- Subhansu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

- Ahmed Masry, Do Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. [ChartQA: A benchmark for question answering about charts with visual and logical reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279, Dublin, Ireland. Association for Computational Linguistics.
- Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruvi Shah, Xianzhi Du, Futang Peng, Anton Belyi, and 1 others. 2024. Mm1: methods, analysis and insights from multimodal llm pre-training. In *European Conference on Computer Vision*, pages 304–323. Springer.
- Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE.
- Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *International Conference on Machine Learning (ICML)*.
- Gobinda Saha, Isha Garg, and Panda Kaushik. 2021. Gradient projection memory for continual learning. In *International Conference on Learning Representations (ICLR)*.
- Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, and 1 others. 2024. Qwen2-vl: To see the world more clearly. *arXiv preprint arXiv:2409.12191*.
- Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. 2023. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671.
- Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE.
- Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115.
- Xiaoyu Yang, Jie Lu, and En Yu. 2024. Adapting multimodal large language model to concept drift from pre-training onwards. *arXiv preprint arXiv:2405.13459*.
- Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *International Conference on Machine Learning (ICML)*.
- Jinghan Zhang, Shiqi Chen, Junteng Liu, and Junxian He. 2023. Composing parameter-efficient modules with arithmetic operations. In *Advances in Neural Information Processing Systems*, volume 36, pages 12502–12522.
- Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Pengfei Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

## A Interpretation of the Task-Vector Singular Value Spectrum

Figure 5 provides a spectral view of the task vector  $\Delta W = W_{ft} - W_{pre}$ , which helps explain why APP’s spectral filtering is well suited for projector adaptation. The spectrum shows a clear low-rank-plus-noise structure. A small number of leading singular values dominate the spectrum, while the remaining components form a long heavy tail with much smaller magnitudes. Moreover, the cumulative energy curve rises to nearly 1.0 before the truncation point  $k = 64$ , indicating that the vast majority of the update energy is concentrated in the dominant head.

More specifically, the first few singular values are substantially larger than the rest, and the spectrum drops sharply within the first tens of components. This pattern suggests that the fine-tuning update is primarily carried by a small set of principal directions, which can be interpreted as the main task-relevant adaptation subspace. After this rapid decay, the spectrum enters a relatively flat heavy-tail region, where many singular values remain small but non-negligible. In our interpretation, these tail components are less likely to encode coherent task structure and are more likely to reflect geometric noise introduced during fine-tuning.

This observation directly motivates APP. Instead of preserving the full update matrix, APP performs singular value decomposition on  $\Delta W$  and keeps only the top- $k$  components. Such a truncation retains the dominant low-rank structure while filtering out the heavy-tail directions. As shown in Figure 5, the top-64 singular components explain 99.5% of the total energy, meaning that the discarded tail contributes only a negligible fraction to the Frobenius norm of the update. Therefore, spectral truncation removes little useful signal in terms of energy, but can substantially reduce the noisy perturbations that distort the geometry of the visual-textual mapping.

This behavior is particularly important for multi-modal projectors. The projector serves as the interface between visual embeddings and the language model input space, so its effectiveness depends not only on task adaptation but also on preserving the cross-modal geometry established by the pre-trained model. Heavy-tail components, although individually weak, are numerous and can collectively introduce irregular perturbations that break vision-language alignment. By suppressing these

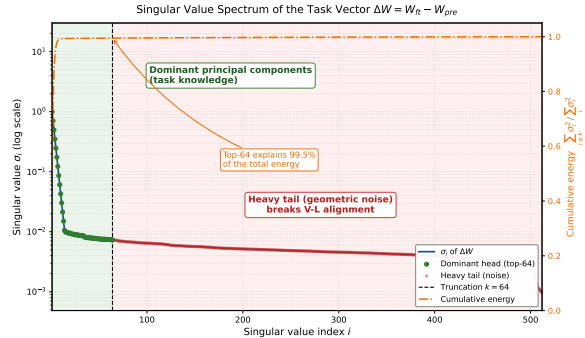


Figure 5: Singular value spectrum of the task vector  $\Delta W = W_{ft} - W_{pre}$ . The spectrum exhibits a dominant head and a long heavy tail. The top-64 singular components explain 99.5% of the total energy, supporting the use of spectral truncation in APP.

weak directions, APP acts as a structured denoising mechanism: it preserves the dominant task knowledge while removing the high-rank noise that is most likely to harm alignment.

Overall, the spectrum in Figure 5 supports the central assumption behind APP: the task vector is not uniformly informative across all directions, but instead consists of a dominant low-rank signal plus a noisy high-rank tail. Spectral filtering is therefore not merely a compression strategy, but a principled way to improve projector quality by denoising the fine-tuning update in accordance with its intrinsic spectral structure.

## B Detailed Derivation of APP

Let the task loss be  $\mathcal{L}(\theta)$ . We approximate the loss landscape around  $\theta_{pre}$  using a second-order Taylor expansion:

$$\mathcal{L}(\theta) \approx \mathcal{L}(\theta_{pre}) + g^T \tau + \frac{1}{2} \tau^T H \tau \quad (8)$$

where  $g$  is the gradient and  $H$  is the Hessian. Assuming the pre-trained model is at a local minimum for general tasks,  $g \approx 0$ . The task adaptation seeks to move in directions of high curvature for the new task but low curvature for the general knowledge. SVD on the weight update  $\tau$  implicitly approximates the principal directions of the Hessian under the assumption of isotropic parameter distribution in the update space.

### B.1 Layer Sensitivity

We applied APP to different layers of the MLP projector.

- **Layer 1 (Closer to Vision):** Filtering here has minimal impact on text capability but reduces visual accuracy slightly.
- **Layer 2 (Closer to LLM):** Filtering here is crucial. Applying Naive TV to Layer 2 causes the majority of the MMLU drop.

This suggests that the interface to the LLM is the most geometrically sensitive region.

## C Full Hyperparameter List

We list all hyperparameters used for reproducibility.

- **Vision Encoder:** CLIP-ViT-L/14 (Frozen)
- **LLM:** Vicuna-7B v1.5 (Frozen)
- **Projector:** MLP (4096 -> 4096 -> 4096)
- **Max Sequence Length:** 2048
- **Gradient Accumulation Steps:** 1