EDITING VISUAL CAPABILITIES OF LARGE VISION-LANGUAGE MODELS THROUGH TASK ARITHMETIC

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision-language models (LVLMs) are typically adapted through end-to-end multimodal training, even when the desired modification primarily concerns visual capabilities. In this work, we study whether such capabilities can be upgraded independently from the rest of the model, as is common in human-engineered modular systems. We show that LVLMs support localized interventions on the vision tower: task updates can be constructed from independently specialized CLIP vision encoders and applied directly to the LVLM vision tower, while keeping the multimodal adapter and language model fixed. This provides an efficient means of transferring visual expertise into LVLMs without additional multimodal training and, when available, enables the reuse of fine-tuned or domain-specialized CLIP checkpoints. We demonstrate that this intervention supports multiple editing goals, including improving a single downstream task, composing improvements across multiple tasks, and removing targeted visual capabilities while preserving model behavior. Our results show that vision-encoder editing improves target-task performance while maintaining the original multimodal capabilities of the LVLM, at a fraction of the cost of end-to-end training.

1 INTRODUCTION

Large vision-language models (LVLMs) such as LLaVA (Liu et al., 2023) combine a pretrained CLIP (Radford et al., 2021) vision encoder with a large language model through a multimodal adapter, enabling visual question answering, captioning, and multimodal instruction following. Despite this modular design, LVLM adaptation is typically performed through end-to-end post-training, most commonly via visual instruction tuning in either full fine-tuning or parameter-efficient settings (Liu et al., 2023; 2024a; Zhai et al., 2024; Zhang et al., 2024a; Wang et al., 2024). Such adaptation is computationally expensive, requiring optimization through large models and dependence on multimodal instruction data. Differently, in real-world systems, domain-specific behavior is often modified by replacing or improving only the relevant subsystem rather than redesigning the entire pipeline. A vehicle intended for rough terrain, for example, may benefit more from specialized tires than from redesigning its drivetrain. Motivated by this analogy, we ask whether LVLMs admit a similarly modular form of adaptation, in which specialized visual capabilities can be transferred by modifying only the vision encoder.

Vision-encoder-focused interventions offer a practical route to adapt the visual capabilities of an LVLM without additional multimodal training. As a matter of fact, visual expertise learned independently by the vision encoder could be transferred to the full model while keeping the multimodal adapter and language model fixed, substantially reducing the computational requirements of end-to-end adaptation. As an additional benefit, this approach could also enable the reuse of domain-specialized or fine-tuned CLIP (Radford et al., 2021) checkpoints available through platforms such as Hugging Face (Hugging Face, 2026), including models tailored to image-safety filtering, character recognition and other visual domains, even when no corresponding LVLM variant is available.

To exploit this opportunity, task arithmetic (Ilharco et al., 2023) provides a simple mechanism for editing models directly in weight space. Rather than retraining an entire model, task arithmetic represents the effect of fine-tuning as a displacement in parameter space named task vector. Such task-specific updates can then be added to insert a capability, summed to compose multiple capabilities,

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

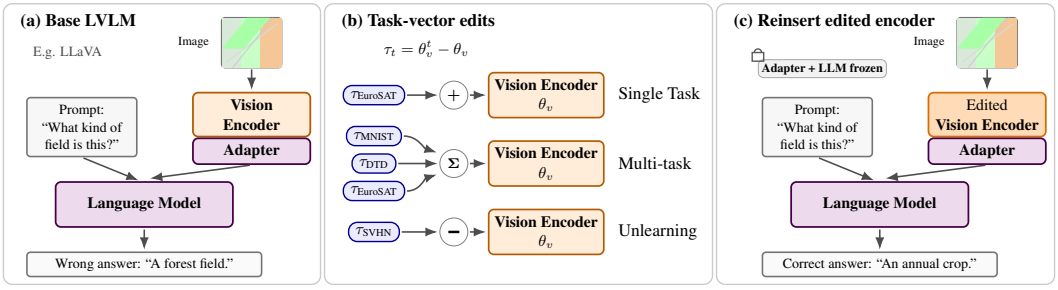

Figure 1: Graphical abstract of our modular LVLM editing framework. (a) A large vision-language model with its different modules: a vision encoder, an adapter, and a language model. (b) Task vectors $\tau_t = \theta_v^t - \theta_v$ learned from task-specialized vision encoders can be added individually, combined across several tasks, or subtracted to suppress a capability. Here, θ_v denotes the parameters of the base vision encoder and θ_v^t denotes the parameters specialized for task t. (c) The edited vision encoder is plugged back into the full LVLM while the adapter and language model remain frozen.

or subtracted to repress a task-specific behavior. This makes task arithmetic a natural candidate for lightweight and modular visual editing.

In this work, we investigate whether task vectors learned from standalone CLIP vision encoders can be applied directly to the matching vision tower of an LVLM while freezing the multimodal adapter and language model. We study this setting on LLaVA, covering full and LoRA (Hu et al., 2021) fine-tuned visual task vectors, as well as single-task insertion, multi-task vector composition, and vector unlearning.

Our experiments show that editing only the vision encoder can improve the visual capabilities of the full LVLM while preserving general multimodal behavior across multiple datasets. These results suggest that visual expertise can be transferred modularly through the vision stack, without requiring additional expensive multimodal training. Our main contributions are as follows:

- We introduce a modular editing framework for LVLMs that reuses task-specialized vision-encoder checkpoints to improve visual capabilities without retraining either the multimodal adapter or the language model.
- We demonstrate the framework on single-task editing, multi-task model merging, and targeted visual unlearning, through experiments spanning multiple domain-specific visual datasets and multimodal control benchmarks, using both full fine-tuning and LoRA-derived task vectors.
- We provide additional analyses of vision-encoder editing in LVLMs, including the relationship between standalone vision-encoder improvements and downstream LVLM gains, comparisons with adapter fine-tuning, and qualitative changes in multimodal image understanding.

Overall, our work enables direct reuse of the growing ecosystem of pretrained and domain-specialized vision-encoder checkpoints to edit the visual capabilities of LVLMs in a lightweight and modular manner.

2 Related Work

Large Vision-Language Models. A common class of Large Vision-Language Models (LVLMs) reuses pretrained visual encoders as perception modules inside generative multimodal systems. Contrastive two-tower models such as CLIP train separate image and text encoders on image–text pairs, producing transferable visual representations that support zero-shot recognition and downstream adaptation (Radford et al., 2021; Ilharco et al., 2021; Cherti et al., 2023). For example, LLaVA-1.5-7B couples a pretrained CLIP vision tower to a language model through a learned multimodal adapter and then applies visual instruction tuning (Liu et al., 2023; 2024a). Recent analyses further show that, despite strong captioning and VQA abilities, instruction-tuned LVLMs can remain limited on

fine-grained visual categorization tasks (Kim & Ji, 2024; Yu et al., 2026). This setup makes the visual stack a natural target for modular adaptation, yet in practice visual changes are still usually introduced through further end-to-end post-training, most commonly via visual instruction tuning, in either full fine-tuning or parameter-efficient variants (Liu et al., 2023; 2024a; Zhai et al., 2024; Zhang et al., 2024a).

Task arithmetic. Task arithmetic provides a weight-space mechanism for reusing the effect of fine-tuning without performing additional training. Given a base model and a task-fine-tuned model, the task vector is defined as their difference in parameter space; this vector can then be added to insert a capability, summed with other task vectors to compose capabilities, or negated to suppress task-specific behavior (Ilharco et al., 2023). Recent works generalize task arithmetic to transfer across different models by first aligning models up to permutation symmetries (Ainsworth et al., 2023), or by transporting task vectors through re-basin-based alignment (Rinaldi et al., 2025). In contrast to these works, which primarily study task arithmetic at the level of complete models, we investigate whether task vectors learned by standalone vision encoders can be reused to directly edit a modular component of a LVLM, namely its vision tower, while leaving the multimodal adapter and language model unchanged.

Vision encoder update of a LVLMs. Few works study the independent update of the visual encoder of LVLMs, and they do it in different problem settings. LoRSU (Panos et al., 2025) introduces LoRA updates for critical parameters of LVLMs to improve visual behavior in a few-shot continual learning setting. Robust-CLIP (Schlarmann et al., 2024) adversarially fine-tunes the CLIP vision encoder and replaces the original encoder in downstream LVLMs such as LLaVA, improving robustness under visual attacks without retraining the downstream model. Our aim, instead, is to show that task vectors can effectively edit the vision encoder to improve the visual capabilities of LVLMs, enabling direct reuse of pretrained and task-specialized vision-encoder checkpoints without additional multimodal training. We support this claim through a systematic and broad empirical study spanning multiple domains, adaptation regimes, and evaluation benchmarks.

3 Background

We first describe the modular parts of the studied LVLM, which is composed of a vision encoder, a multimodal adapter, and a language model. We then formalize how task vectors are obtained and how they are used for visual capability editing. We report the pipeline in Fig. 1.

3.1 LVLM Architecture and Parameter Groups

We denote the parameters of the pretrained LVLM by

$$\Theta = \{\theta_v, \theta_a, \theta_l\}, \tag{1}$$

where θ_v, θ_a, and θ_l are respectively the parameters of the vision encoder, multimodal adapter, and language model. Given an image x and a text prompt q, the vision encoder first extracts visual representations

$$h_v = f_v(x; \theta_v). \tag{2}$$

The multimodal adapter maps these features into the token space expected by the language model,

$$z_v = g(h_v; \theta_a), \tag{3}$$

after which the language model generates the output conditioned on the adapter-transformed visual tokens and the text prompt,

$$y \sim p_{\theta_l}(y \mid z_v, q). \tag{4}$$

Standard LVLM adaptation typically relies on visual instruction tuning, where the entire model is optimized on image-text instruction-response triples (Liu et al., 2023; 2024a; Zhai et al., 2024; Zhang et al., 2024a). Even in parameter-efficient settings, this process still requires optimization through the complete multimodal pipeline. In contrast, we study a modular editing setting in which only the vision encoder is modified, while the multimodal adapter and language model remain frozen. This allows visual capabilities learned independently by standalone vision encoders to be transferred into the LVLM without any additional multimodal training.

3.2 VISION-ENCODER SPECIALIZATION

For each target task t and dataset $\mathcal{D}_t$, we start from the base vision encoder θ_v and specialize it through supervised classification. We denote by g_T the frozen CLIP text encoder associated with the same pretrained model. For each class $c \in \mathcal{C}_t$, the text encoder produces a fixed class embedding $g_T(c)$.

Given an image x and vision-encoder parameters $\tilde{\theta}_v$, the visual representation is

$$f_v(x; \tilde{\theta}_v), \tag{5}$$

and the prediction logits are defined as

$$\hat{y}_c(x; \tilde{\theta}_v) = \left\langle f_v(x; \tilde{\theta}_v), g_T(c) \right\rangle, \quad c \in \mathcal{C}_t. \tag{6}$$

Optimizing the standard cross-entropy loss on $\mathcal{D}_t$ produces a task-specialized vision encoder with parameters θ_v^t.

3.3 REPRESENTATION ANCHORING THROUGH FROZEN TEXT EMBEDDINGS

During specialization, the CLIP text encoder remains frozen, so the class prototypes stay fixed in the original pretrained embedding space and only the vision encoder parameters are updated. Rather than optimizing against arbitrary task-specific classifier weights, the specialized vision encoder is therefore constrained to remain aligned with the original vision-language representation geometry.

This property is particularly important in our setting because the LLaVA multimodal adapter is trained to process visual features originating from the same pretrained CLIP encoder. By preserving the semantic structure of the original embedding space during specialization (Kumar et al., 2022), the resulting task vectors remain more compatible with the frozen multimodal adapter when reinserted into the LVLM. In this sense, the frozen text encoder acts as a semantic anchor that stabilizes the interface between the edited vision tower and the downstream multimodal components. We provide an empirical analysis of the proposed anchoring mechanism in Sec. F.

3.4 TASK VECTORS

For each task t, we define the corresponding task vector as

$$\tau_t = \theta_v^t - \theta_v, \tag{7}$$

which represents the parameter displacement induced by specialization on task t. We will study three task-arithmetic editing operations:

Single-task editing. A single capability is inserted through

$$\theta_v' = \theta_v + \alpha\tau_t, \tag{8}$$

where α controls the strength of the edit.

Multi-task editing. Multiple capabilities are composed through

$$\theta_v' = \theta_v + \alpha \sum_{t \in \mathcal{S}} \tau_t, \tag{9}$$

where $\mathcal{S}$ is a set of target tasks.

Unlearning. A capability is suppressed through

$$\theta_v' = \theta_v - \alpha\tau_t. \tag{10}$$

The edited vision encoder is then reinserted into the frozen LVLM while keeping the multimodal adapter and language model unchanged.

4 SINGLE-TASK ARITHMETIC

We first evaluate whether task vectors learned from standalone vision encoders can transfer task-specific visual capabilities to the full LVLM through single-task editing of the vision tower.

All experiments in this section use the LLaVA-1.5-7B checkpoint (Liu et al., 2024a) with its original OpenAI CLIP vision encoder, *i.e.*, ViT-L/14-336 (Radford et al., 2021). For each target task t, we independently specialize the standalone CLIP vision encoder on the corresponding dataset $\mathcal{D}_t$ using the supervised classification setup described in Sec. 3. The resulting task vector τ_t is then inserted into the frozen LLaVA vision tower through

$$\theta_v' = \theta_v + \alpha\tau_t, \tag{11}$$

where α controls the strength of the edit.

We evaluate the edited LVLM on both target tasks and control benchmarks. As target tasks, we use the eight domain-specialized classification datasets from the Vision-8 benchmark (Ilharco et al., 2023): Stanford Cars (Krause et al., 2013), DTD (Cimpoi et al., 2014), EuroSAT (Helber et al., 2019), GTSRB (Stallkamp et al., 2011), MNIST (Lecun et al., 1998), RESISC45 (Cheng et al., 2017), SUN397 (Xiao et al., 2016), and SVHN (Netzer et al., 2011). As a control benchmark, we use VQAv2 (Goyal et al., 2017), which was specifically designed to reduce language-prior shortcuts through balanced image-question pairs. This design makes VQAv2 well suited for assessing whether edits to the vision encoder preserve the general visual grounding capabilities of the LVLM beyond the target classification tasks. Throughout the main body, we use VQAv2 as our primary control benchmark, complemented by a broader evaluation on standard LLaVA benchmarks in Sec. D.

For each target classification dataset, we cast the task as visual question answering by prompting the LVLM to choose among the dataset labels. We use deterministic decoding and answer normalization before exact-match scoring, so that evaluation measures classification accuracy rather than sensitivity to free-form wording. Full prompt templates and parsing details are provided in Sec. E.

Table 1: Single-task task-vector insertion on the LLaVA vision encoder. The upper block reports standalone CLIP classification performance before and after specialization. The lower block reports the corresponding LLaVA performance obtained after replacing the frozen vision tower with the edited encoder. The final column reports accuracy on the VQAv2 control benchmark, our primary measure of multimodal capability retention.

	Target Accuracy								Average		VQAv2
Model	**Cars**	**DTD**	**EuroSAT**	**GTSRB**	**MNIST**	**RES45**	**SUN**	**SVHN**	**Avg.**	**Δ**	**Ctrl.**
Standalone CLIP											
Zero-shot CLIP	78.0	55.3	61.0	54.4	59.9	71.8	70.3	24.1	59.4	–	–
FT CLIP	93.4	84.4	98.7	99.3	98.9	97.1	82.7	97.8	94.0	+ **34.6**	–
LoRA CLIP	90.3	82.9	98.7	99.0	99.6	97.0	80.4	97.3	93.2	+ **33.8**	–
LVLM Transfer											
Zero-shot LLaVA	37.6	36.3	49.6	24.0	84.9	51.0	85.5	57.5	53.3	–	76.6
LLaVA + FT CLIP	36.8	44.9	89.6	35.4	98.8	74.4	87.0	95.9	70.4	+ **17.1**	75.5
LLaVA + LoRA CLIP	36.6	37.7	89.4	31.8	98.9	66.0	86.1	93.8	67.5	+ **14.2**	76.0

Single-task edits improve target accuracy while preserving control. As shown in Tab. 1, single task-vector insertion enables specialized CLIP encoders to improve LVLM performance on the corresponding target tasks. Across the Vision-8 datasets, replacing the frozen vision tower with an edited encoder improves target accuracy for both full fine-tuning and LoRA-derived task vectors. Figure 2 further shows that LVLM gains are positively correlated with the corresponding standalone CLIP gains, indicating that improvements learned in the vision encoder often transfer to the complete LVLM after the edited encoder is plugged back into the frozen multimodal adapter. Although LVLM gains are typically smaller than standalone CLIP gains, their consistent positive relationship demonstrates that task-vector arithmetic remains effective after reinsertion into the full multimodal model. This reduction in gain is expected, since the edited visual representation must still pass through the pretrained multimodal adapter and be used by the language model under a prompted

classification interface (Zhang et al., 2024b). Importantly, these target-task gains are accompanied by only minor changes in VQAv2 control accuracy, indicating that the inserted task vectors largely preserve the LVLM's general visual grounding capabilities.

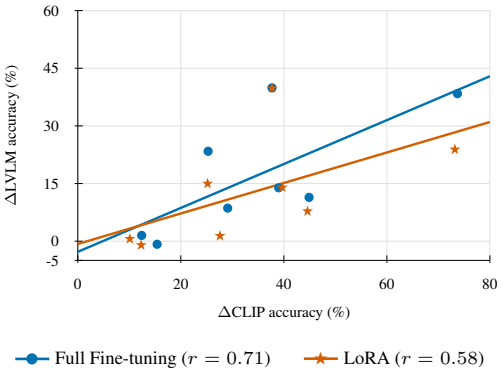

Figure 2: CLIP–LVLM gains transfer correlation for single-task vision-encoder edits. Each point compares CLIP and LVLM accuracy gains for one dataset after inserting the same task vector. LVLM gains are usually smaller than the corresponding CLIP gains.

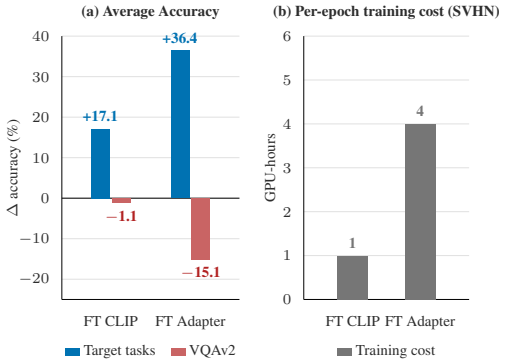

Figure 3: Cost and performance comparison between vision-encoder editing and adapter fine-tuning on the Vision-8 benchmark. **(a)** Average Δ Accuracy in target-task and VQAv2 w.r.t. zero-shot LLaVA. **(b)** Per-epoch training cost, measured on SVHN.

Table 2: Adapter fine-tuning versus vision-encoder editing on the Vision-8 benchmark. Entries report the change in accuracy with respect to zero-shot LLaVA.

Model	Cars		DTD		EuroSAT		GTSRB		MNIST		RES45		SUN		SVHN	
	Target	VQAv2	Target	VQAv2	Target	VQAv2	Target	VQAv2	Target	VQAv2	Target	VQAv2	Target	VQAv2	Target	VQAv2
ZS LLaVA	37.6	76.6	36.3	76.6	49.6	76.6	24.0	76.6	84.9	76.6	51.0	76.6	85.5	76.6	57.5	76.6
+ FT CLIP	-0.8	-0.8	+8.6	-1.0	+40.0	-0.9	+11.4	-1.2	+13.9	-0.4	+23.4	-1.3	+1.5	-1.7	+38.4	-1.1
+ FT Adapter	+49.3	-20.1	+39.0	-28.1	+47.7	-0.4	+64.1	-26.7	+13.9	-0.4	+41.9	-28.8	+4.0	-15.7	+31.0	-0.8

Efficiency of vision-encoder-only editing. To assess whether task arithmetic on the vision encoder can serve as a more efficient alternative to direct adapter fine-tuning, we compare vision-encoder-only task vector insertion against fine-tuning the LVLM-side adapter (Zhang et al., 2024b). Table 2 reports this comparison on the Vision-8 benchmark.. From a **cost perspective**, vision-encoder-only editing avoids repeated fine-tuning of the LVLM-side adapter and allows task-specific vision-encoder updates to be inserted modularly and swapped across target domains. This makes the edit **cheaper** to apply than adapter fine-tuning, which requires updating the LVLM-side interface for each task.

Figure 3 summarizes the resulting trade-off reporting average target-task gains and VQAv2 preservation across the Vision-8 datasets, together with the per-epoch training cost on SVHN. We report SVHN as a representative case, since it is among the most expensive Vision-8 tasks; however, we observe a comparable gap in GPU-hours between the two approaches on every Vision-8 dataset. CLIP fine-tuning is measured on a single NVIDIA RTX A6000, while adapter fine-tuning is performed on a single NVIDIA A100, which is required to ensure that the LVLM adapter fine-tuning pipeline fits with the available GPU and system memory, accounting for the model, intermediate activations, optimizer states and data batches during training. Despite this hardware advantage, adapter fine-tuning remains substantially more computationally expensive, making the reported cost gap a conservative estimate. In terms of performance, adapter fine-tuning achieves the larger average target-task gain, but these gains come at a substantial cost to general visual grounding, as reflected by large drops in VQAv2 accuracy. In contrast, task-vector insertion requires much lower training cost while preserving VQAv2 much more closely, highlighting a favorable cost-preservation trade-off for vision-encoder-only editing.

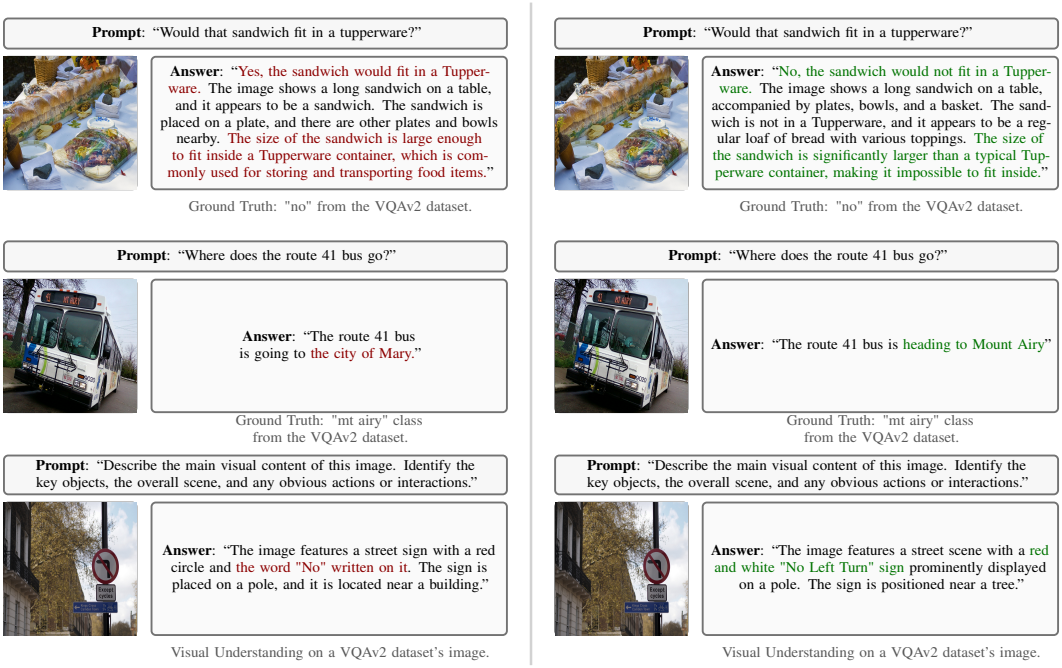

Figure 4: Qualitative illustration of LVLM visual editing. Green words highlight task-relevant content improved by editing the vision encoder, while red words indicate incorrect or less relevant base-model outputs. The examples show that adapting only the vision encoder can change the final multimodal response even though the adapter and language model remain frozen.

Qualitative effect of visual editing. Figure 4 complements the quantitative results with before-and-after examples. These examples illustrate the intended mechanism of the method: improving the visual side of the LVLM can change the final multimodal response even though the adapter and language model are fixed, beyond classification and into multimodal understanding.

5 Multi-Task Arithmetic

Here we study whether multiple visual capabilities can be composed within a single LVLM by merging task vectors in the vision encoder. Unlike the single-task setting, where one task vector is inserted at a time, the multi-task setting combines all task-specific updates into a single merged encoder, which is reinserted into the LLaVA architecture.

In addition to standard Task Arithmetic (TA) (Ilharco et al., 2023), we evaluate TSV-Merge (Gargiulo et al., 2025), a recent conflict-aware merging method that decomposes task updates into task singular vectors and merges them in a low-rank space designed to reduce inter-task interference. We further combine TSV with Core Space (Panariello et al., 2025), a LoRA merging framework that aligns updates in a shared low-rank basis before composition. We refer to this variant as TSV-Core. For each method, the scaling coefficient α is selected on validation sets in the standalone CLIP setting following standard practice (Ilharco et al., 2023; Gargiulo et al., 2025; Panariello et al., 2025), and then reused unchanged for LVLM transfer. Table 3 reports results on the Vision-8 benchmark together with preservation on the VQAv2 control dataset. In the standalone CLIP setting, merged encoders substantially improve over zero-shot CLIP across the target tasks, confirming that the merged task vectors retain strong multi-task visual specialization. Conflict-aware merging methods such as TSV further improve over naive task arithmetic, showing that reducing interference among task vectors is beneficial even before transfer to the multimodal model.

After reinsertion into LLaVA, the edited LVLM also consistently improves over the zero-shot baseline, demonstrating that multiple visual capabilities can transfer to the full multimodal model without retraining either the multimodal adapter or the language model. However, the downstream gains remain smaller than the corresponding improvements observed on the standalone merged CLIP encoders. This gap highlights an important constraint of modular LVLM editing: improving the vision encoder alone is not sufficient if the resulting visual representation drifts too far from the feature distribution expected by the frozen multimodal adapter. Consequently, successful multi-task transfer depends not only on the quality of the merged vision encoder itself, but also on preserving compatibility with the pretrained multimodal interface.

The comparison between merging strategies further supports this interpretation. Conflict-aware methods such as TSV not only produce the strongest standalone CLIP encoders, but also yield the best downstream LVLM transfer, suggesting that reducing task interference benefits both vision classification and multimodal reuse. At the same time, the ranking between methods is not perfectly preserved after reinsertion into LLaVA. In particular, LoRA-based merging can remain competitive in the standalone CLIP setting while transferring less effectively to the LVLM. One possible explanation is that the low-rank constraint limits the expressivity of the merged update, producing representations that remain effective for CLIP classification but less compatible with the frozen multimodal adapter and downstream language model. Overall, strong standalone vision performance alone does not guarantee equally strong multimodal transfer.

Finally, multi-task vision editing is computationally attractive, while enabling task-specific updates to reused, composed, or subtrcated without LVLM training.

Table 3: Multi-task task-vector merging on the LLaVA vision encoder. The upper block reports standalone CLIP performance after merging. The lower block reports the corresponding LLaVA performance obtained after replacing the frozen vision tower with the merged one. The final column reports accuracy on the VQAv2 control.

		Target Accuracy								Average		VQAv2
Method	**Model**	**Cars**	**DTD**	**EuroSAT**	**GTSRB**	**MNIST**	**RES45**	**SUN**	**SVHN**	**Avg.**	**Δ**	**Ctrl.**
Standalone CLIP												
–	Zero-shot CLIP	78.0	55.3	61.0	54.4	59.9	71.8	70.3	24.1	59.4	–	–
TA (FT)	Merged FT CLIP	85.9	65.4	89.0	85.9	97.7	87.8	75.0	85.8	84.1	+**24.7**	–
TA (LoRA)	Merged LoRA CLIP	85.3	72.1	93.4	92.9	99.0	90.1	74.8	95.7	87.9	+**28.5**	–
TSV (FT)	Merged FT CLIP	92.3	80.5	94.7	96.6	98.6	95.0	79.6	95.2	91.6	+**32.2**	–
TSV-Core (LoRA)	Merged LoRA CLIP	88.2	81.2	95.2	97.5	98.9	92.2	77.7	96.3	90.9	+**31.5**	–
LVLM Transfer												
–	Zero-shot LLaVA	37.6	36.3	49.6	24.0	84.9	51.0	85.5	57.5	53.3	–	76.6
TA (FT)	LLaVA + FT CLIP	37.7	37.7	59.0	32.4	94.8	58.3	85.6	75.8	60.2	+**6.9**	75.5
TA (LoRA)	LLaVA + LoRA CLIP	38.7	36.6	75.1	31.5	97.7	60.2	86.5	91.1	64.7	+**11.4**	75.8
TSV (FT)	LLaVA + FT CLIP	37.7	42.7	78.7	36.0	97.6	67.2	86.1	87.8	66.7	+**13.4**	75.5
TSV-Core (LoRA)	LLaVA + LoRA CLIP	37.0	37.0	76.1	33.8	97.5	60.5	86.2	92.2	65.0	+**11.7**	75.7

6 UNLEARNING

We finally evaluate whether negative task vectors can selectively suppress a targeted visual capability while preserving the broader behavior of the LVLM. In this setting, we negate one task vector at a time and apply the resulting update to the base vision encoder, as defined in Eq. (10). For each target task t, we negate the corresponding full fine-tuning task vector and tune its scaling coefficient independently. The selected coefficient maximizes suppression on the target task while preserving performance on all remaining Vision-8 tasks. VQAv2 is then used as a control benchmark to assess preservation of general multimodal capabilities.

Figure 5 shows that negative task vectors consistently reduce target-task performance while largely preserving VQAv2 accuracy, indicating selective suppression rather than broad model degradation.

Overall, these results show that task-vector arithmetic in the LVLM vision encoder supports both positive and negative composition. Subtracting a task vector can suppress a targeted visual capability while preserving general visual grounding, indicating that the task-specific structure underlying

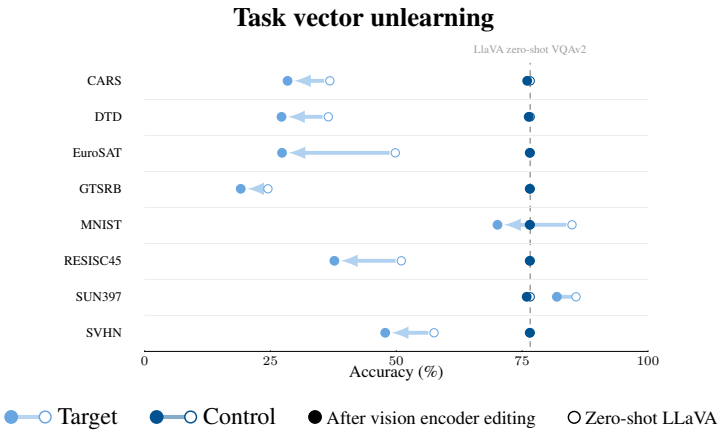

Figure 5: Full fine-tuning task-vector unlearning in LLaVA. Accuracy shifts in both target and VQAv2 control datasets are shown. Open markers denote the initial score and filled markers denote the score after the edit.

arithmetic in standalone CLIP encoders transfers to more complex multimodal architectures while remaining compatible with the LVLM adapter.

7 Discussion, Limitations, and Future Work

Our results suggest that vision-encoder task vectors provide a lightweight route for adapting LVLMs, since visual updates can be inserted without retraining either the multimodal adapter or the language model. The key open question is therefore not only whether a task vector improves a standalone CLIP encoder, but under which conditions that improvement transfers through the frozen multimodal interface. Future work should better characterize this transfer in terms of adapter compatibility, feature drift, task-vector scale, and layerwise structure.

Another important direction is extending vision-side editing to LVLMs whose visual backbones diverge from the original CLIP checkpoint during multimodal training. In such settings, checkpoint alignment or re-basin techniques may enable task-vector transfer across parameter spaces (Ainsworth et al., 2023; Rinaldi et al., 2025). Broader evaluations across LVLM families, control benchmarks, and computational settings are needed to establish when modular vision-side editing is preferable to targeted multimodal retraining.

Limitations. Our experiments are limited to a LLaVA-style architecture with a CLIP-based vision tower and a frozen multimodal adapter. This setting is important because it isolates the effect of editing the vision encoder, but the results may not directly extend to architectures with substantially updated visual backbones or different multimodal interfaces. Moreover, our evaluation focuses on benchmark accuracy and capability preservation, without comprehensively assessing open-ended behavior, distribution shifts, or multimodal safety.

8 Conclusions

We investigated whether task arithmetic can be applied to the vision tower of a large vision-language model as a modular mechanism for editing multimodal behavior. Across single-task insertion, multi-task merging, negative-vector unlearning, and adapter fine-tuning comparisons, our results show that vision-encoder edits can induce meaningful LVLM changes while leaving the language model and multimodal adapter untouched. The transfer from standalone CLIP gains to LVLM gains is not automatic, but it is structured enough to make the vision encoder a useful control surface for efficient multimodal adaptation. These findings suggest that vision-encoder editing offers a practical route to targeted adaptation whose usefulness depends on both visual specialization and compatibility with the multimodal interface, motivating further investigations into which encoder-side modifications preserve this compatibility.

ETHICS STATEMENT

This work uses existing public datasets and pretrained model checkpoints, and does not introduce new human-subject data collection. The proposed editing mechanism is intended to support efficient adaptation, capability composition, and selective suppression of visual skills in LVLMs without repeatedly retraining the full multimodal model.

Model editing methods can have dual-use implications. The same ability to add or remove capabilities could be used to improve specialization and reduce unwanted behavior, but it could also be used to hide undesirable model properties or selectively alter behavior without transparent documentation. For this reason, edited checkpoints should be accompanied by clear reporting of the base model, task vectors, scaling coefficients, datasets, and evaluation results. By reusing vision-side adaptations rather than retraining entire LVLMs, the approach may reduce compute in suitable settings, although the experiments themselves still require GPU resources and should be reported with their associated compute budget.

REPRODUCIBILITY STATEMENT

All code needed to reproduce our experiments is released anonymously (Sec. A). Section 3 formalizes the vision-encoder specialization and task-arithmetic operations, and Sec. B lists the checkpoints, hyperparameters, scaling coefficients, and seeds we report for each experiment. Dataset descriptions for the Vision8 target tasks and the VQAv2 control benchmark are provided in Secs. C and D, and the exact prompt templates, decoding settings, and answer-normalization procedure used for LVLM evaluation are given in Sec. E.

AI USE STATEMENT

In this work, we used generative AI tools only to polish the writing, improve readability of the manuscript and to assist in code refactoring and maintenance during the development process. We have not used generative AI tools to develop the research methodology, formulate hypotheses, design the experiments, implement the proposed method and interpret the results. We have manually reviewed all AI-assisted text and code modifications and we take responsibility for the final content of this work.

REFERENCES

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *International Conference on Learning Representations*, 2023.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote Sensing Image Scene Classification: Benchmark and State of the Art. *Proceedings of the IEEE*, 2017.

Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705, June 2025.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017.

Danna Gurari, Qing Li, Abigale Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3608–3617, 2018. URL https://api.semanticscholar.org/CorpusID:3831582.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Hugging Face. Hugging face hub. https://huggingface.co, 2026. Accessed: 2026-05-14.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations*, 2023.

Jeonghwan Kim and Heng Ji. Finer: Investigating and enhancing fine-grained visual concept recognition in large vision language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D Object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji rong Wen. Evaluating object hallucination in large vision-language models. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a.

Huan Liu, Lingyu Xiao, Jiangjiang Liu, Xiaofan Li, Ze Feng, Sen Yang, and Jingdong Wang. Revisiting mllms: An in-depth analysis of image classification abilities, 2024b. URL https://arxiv.org/abs/2412.16418.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024c.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Aniello Panariello, Daniel Marczak, Simone Magistri, Angelo Porrello, Bartłomiej Twardowski, Andrew D. Bagdanov, Simone Calderara, and Joost van de Weijer. Accurate and efficient low-rank model merging in core space. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=y1z7SAS8q8.

Aristeidis Panos, Rahaf Aljundi, Daniel Olmeda Reino, and Richard E. Turner. Efficient few-shot continual learning in vision-language models. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=sQ1w92WW0V.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Filippo Rinaldi, Giacomo Capitani, Lorenzo Bonicelli, Donato Crisostomi, Federico Bolelli, Elisa Ficarra, Emanuele Rodolà, Simone Calderara, and Angelo Porrello. Update your transformer to the latest release: Re-basin of task vectors. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 51688–51702, 2025. URL https://proceedings.mlr.press/v267/rinaldi25a.html.

Christian Schlarmann, Naman Deep Singh, Francesco Croce, and Matthias Hein. Robust CLIP: Unsupervised adversarial fine-tuning of vision embeddings for robust large vision-language models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 43685–43704, 2024.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, 2011.

Taowen Wang, Yiyang Liu, James Chenhao Liang, Junhan Zhao, Yiming Cui, Yuning Mao, Shaoliang Nie, Jiahao Liu, Fuli Feng, Zenglin Xu, Cheng Han, Lifu Huang, Qifan Wang, and Dongfang Liu. M2pt: Multimodal prompt tuning for zero-shot instruction learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3723–3740, 2024. doi: 10.18653/v1/2024.emnlp-main.218.

Jianxiong Xiao, Krista A Ehinger, James Hays, Antonio Torralba, and Aude Oliva. Sun database: Exploring a large collection of scene categories. *International Journal of Computer Vision*, 2016.

Hongtao Yu, Yuxin Peng, Serge Belongie, and Xiu-Shen Wei. Benchmarking large vision-language models on fine-grained image tasks: A comprehensive evaluation. In *International Conference on Learning Representations*, 2026.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, and Sergey Levine. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. In *Advances in Neural Information Processing Systems 37*, 2024.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5625–5644, 2024a. doi: 10.1109/TPAMI.2024.3369699.

Yuhui Zhang, Alyssa Unell, Xiaohan Wang, Dhruba Ghosh, Yuchang Su, Ludwig Schmidt, and Serena Yeung-Levy. Why are visually-grounded language models bad at image classification? *Conference on Neural Information Processing Systems (NeurIPS)*, 2024b.

A Code Availability

All the code has been uploaded as a .zip file during the submission and the link will be available after the review period.

B Reproducibility Checklist

For each experiment, we report or release:

- The exact base LVLM checkpoint;
- The exact CLIP vision encoder checkpoint used inside the LVLM;
- The eight vision datasets and label spaces;
- Full fine-tuning hyperparameters;
- LoRA rank, alpha, dropout, and target modules;
- Task-vector scaling coefficient α for single-task edits and unlearning;
- Naive multi-task scaling coefficient α, including $\alpha = 1.0$ and the validation-selected α;
- whether LoRA weights are evaluated as adapters or merged into the encoder;
- whether the edit is addition, naive multi-task summation, or full-fine-tuning negative-vector unlearning;
- classification prompt templates;
- LVLM preservation benchmarks and decoding settings;
- random seeds and validation splits used to select α.

C Vision8 Datasets

The eight datasets in Vision8 are:

- **Cars196** (Krause et al., 2013): The Cars196 dataset is available for non-commercial research purposes. Specific licensing details are not explicitly provided.
- **Describable Textures Dataset (DTD)** (Cimpoi et al., 2014): The DTD is made available to the computer vision community for research purposes. The dataset is licensed under the Creative Commons Attribution 4.0 License (CC BY 4.0).
- **EuroSAT** (Helber et al., 2019): The EuroSAT dataset is licensed under the MIT License.
- **German Traffic Sign Recognition Benchmark (GTSRB)** (Stallkamp et al., 2011): The GTSRB dataset is licensed under the Creative Commons Zero (CC0) Public Domain Dedication.
- **MNIST** (Lecun et al., 1998): The MNIST dataset is publicly available for research purposes. Specific licensing details are not explicitly provided; users are advised to consult the dataset's source for more information.
- **RESISC45** (Cheng et al., 2017): The RESISC45 dataset is licensed under the Creative Commons Attribution 4.0 License (CC BY 4.0).
- **SUN397** (Xiao et al., 2016): The SUN397 dataset is available for research purposes only. Specific licensing details are not explicitly provided; users are advised to consult the dataset's source for more information.
- **Street View House Numbers (SVHN)** (Netzer et al., 2011): The SVHN dataset is available for non-commercial use only.

D CONTROL BENCHMARK SUITE

The control benchmark suite consists of the following datasets:

- **VQAv2** (Goyal et al., 2017): Open-ended visual question answering designed to reduce language priors and emphasize image grounding.
- **MMBench** (Liu et al., 2024c): A bilingual multiple-choice benchmark that evaluates the fine-grained perception and reasoning capabilities of vision–language models using the CircularEval evaluation strategy. We report results for both its English (MMBench^{EN}) and Chinese (MMBench^{CN}) versions.
- **MME** (Fu et al., 2023): A comprehensive benchmark for multimodal large language models that evaluates both perception and cognition across fourteen subtasks. Its instruction–answer pairs are manually designed to reduce evaluation-data leakage and enable standardized quantitative evaluation.
- **POPE** (Li et al., 2023): A polling-based benchmark for evaluating object hallucination by asking binary questions about the presence of objects in an image.
- **TextVQA** (Singh et al., 2019): An open-ended visual question answering benchmark that requires models to read and reason about text embedded in natural images.
- **GQA** (Hudson & Manning, 2019): A visual question answering benchmark designed to evaluate compositional reasoning over real-world images. It uses image scene graphs and structured question representations to mitigate language priors and enable fine-grained analysis of models' reasoning capabilities.
- **VizWiz** (Gurari et al., 2018): A goal-oriented visual question answering benchmark constructed from images and spoken questions submitted by people who are blind. The benchmark evaluates both visual question answering and the ability to recognize questions that cannot be answered from the corresponding image.
- **ScienceQA-IMG** (Lu et al., 2022): The subset of ScienceQA containing an image context, comprising multimodal multiple-choice questions that evaluate scientific knowledge and reasoning with annotated lectures and explanations.

In Tab. 4, we report the complete set of scores for LLaVA-1.5-7B and its vision-edited variants across the aforementioned control benchmark suite.

Table 4: Control benchmark suite results of LLaVA-1.5-7B and its vision-edited variants. MME is reported as the raw score; all other benchmarks are accuracies (%).

Model	VQAv2	GQA	VizWiz	SQA^{IMG}	TextVQA	POPE	MME	MMBench^{EN}	MMBench^{CN}
LLaVA-1.5-7B	76.6	61.7	54.0	69.3	57.6	86.8	1481.8	64.7	58.4
+ TA (FT)	75.5	61.2	53.9	67.5	55.8	84.6	1443.0	64.6	57.3
+ TA (LoRA)	75.8	61.8	53.7	67.9	56.6	86.0	1437.3	65.1	57.8
+ TSV (FT)	75.5	61.5	53.9	67.2	56.0	85.1	1459.3	63.1	56.7
+ TSV-Core (LoRA)	75.7	61.8	54.5	68.0	56.7	85.8	1464.0	63.9	57.4

E PROMPT TEMPLATES AND LVLM EVALUATION PROTOCOL

For classification datasets, we follow the prompt formulation of (Liu et al., 2024b):

```
USER: <image> What is the class of this image? Answer
directly with the corresponding option letter from the
following choices: A. [class name] ... Z. [class name]
ASSISTANT:
```

For short-answer VQA benchmarks, we use:

```
USER: <image> [question] Answer the question using a single
word or phrase. ASSISTANT:
```

For multiple-choice benchmarks, we use:

```
USER: <image> [question] A. [option A] B. [option B] C.
[option C] D. [option D] Answer with the option's letter
from the given choices directly.  ASSISTANT:
```

For target-task evaluation, the label list in the classification prompt is instantiated with the dataset specific class names. However, we adopt a different formulation for MNIST and SVHN, using digits directly as candidate class labels rather than mapping them to letter-indexed answer options. Predictions are decoded deterministically, and both predicted and ground-truth labels are normalized before scoring by converting them to lowercase, stripping surrounding whitespace, and removing trailing punctuation. For edited LVLM evaluations, only the vision encoder is changed; the multimodal adapter and language model remain frozen.

For VQAv2 control evaluation, we use the short-answer prompt template above and report the resulting VQAv2 accuracy as the preservation benchmark. This control score is used to check whether the vision-side edit preserves general image-grounded question answering beyond the target classification task.

F Empirical analysis of Frozen-Text Representation Anchoring

In Section 3.3, we hypothesize that retaining the frozen CLIP text encoder during task specialization constraints the updated vision encoder to remain aligned with the original vision-language representation geometry. Therefore, this alignment preserves the compatibility of the edited visual features with the frozen LLaVA multimodal adapater.

To empirically evaluate this hypothesis, we replace the fixed CLIP text embeddings used during specialization with a learnable classification head. Afterwards, fine-tuning is applied both to the vision encoder and to the learnable classification head. The classification head is subsequently discarded and the task vector is derived exclusively from the parameter update of the vision encoder. The obtained task vector is then applied to the LLaVA vision tower using the same procedure as in the main experiments.

In addition to the end-to-end performance, we quantify the change in the features received by the multimodal adapter. In particular, we compute the mean squared error (MSE) between the adapater-input activations produced by the zero-shot vision encoder and those produced by the specialized vision encoder. We refer to this quantity as *Feature drift / adapter-input shift*; lower values indicate smaller deviations from the original vision tower's adapter-input features.

As reported in Table 5, the two specialization strategies achieve similar average CLIP performance. However, when the resulting vision-encoder task vectors are transferred to LLaVA, the frozen-text specialization yields higher average LLaVA performance and lower adapter-input feature drift. Thus, the learnable-head baseline remains effective on the specialized CLIP tasks but transfers less successfully to LLaVA.

These results are consistent with the representation-anchoring hypothesis: specialization against fixed text prototypes is associated with smaller deviations at the adapter input and better transfer through the frozen multimodal adapter. However, this ablation does not establish that reduced feature drift causes the transfer improvement or identify it as the sole contributing mechanism.

Table 5: Comparison of specialization with frozen CLIP text embeddings and a learnable classification head. Frozen-text specialization achieves similar average CLIP performance, higher average LLaVA performance, and lower adapter-input feature drift. Arrows indicate the preferred direction for each metric.

Specialization head	**Avg. CLIP** ↑	**Avg. LLaVA** ↑	**Feature drift** ↓
Frozen text embeddings	**94.0**	**53.3**	**14.0**
Learnable head	93.5	45.4	17.3