Label-Free Cross-Task LoRA Merging with Null-Space Compression

Supplementary Material

A. Experimental Settings

A.1. Datasets and Tasks

A.1.1. Heterogeneous Vision Tasks

We evaluate 20 dense prediction vision tasks with a ViT-based multi-task model across three datasets spanning indoor and outdoor scenes.

- **NYUD-v2 [68].** An indoor RGB-D dataset of scenes such as living rooms, offices, and kitchens. We use four dense prediction tasks: depth estimation, semantic segmentation, surface normal prediction, and edge detection. The official split provides 795 training images and 654 test images but no validation set. In our experiments, we keep the 795 training images and randomly divide the original test split into 327 validation and 327 test images.
- **PASCAL-Context [60].** An extension of PASCAL VOC 2010 with dense pixel-wise annotations for natural scenes containing everyday objects. We use five tasks: semantic segmentation, human parts estimation, saliency estimation, surface normal prediction, and edge detection. The dataset provides 10,103 images with an official split of 4,998 training and 5,105 validation images but no test set. We keep the 4,998 training images and split the original validation set into 2,607 validation and 2,498 test images following [49, 77].
- **Taskonomy [100].** A large-scale indoor dataset with diverse geometric and semantic tasks. We construct a tiny subset for training and validation by sampling 259,747 training and 35,774 validation images from the standard Taskonomy buildings, and use images from the Ihlen building as the test set with 9,007 images. We adopt eleven dense prediction tasks: Depth Euclidean (DE), Depth Z-buffer (DZ), Edge Texture (ET), Keypoints 2D (K2), Keypoints 3D (K3), Normal (N), Principal Curvature (C), Reshading (R), Segment Unsup2D (S2), and Segment Unsup2.5D (S2.5).

A.1.2. NLI Tasks

We evaluate six sentence-pair classification tasks with LLaMA-3 8B.

- **QNLI [78].** Question–answering reformulated as sentence-pair classification. Each example is a question and a candidate sentence, and the label indicates whether the sentence contains the answer (binary).
- **MNLI [86].** A multi-genre collection of premise–hypothesis pairs for natural language inference. The goal is to predict one of three labels: *entailment*, *contradiction*, or *neutral*.

Table 7. Vision–Language Tasks Setup.

Dataset	Provided Prompt
IconQA, **VizWiz**	Answer the question using a single word.
ChartQA, **DocVQA**	Answer the question using a single word or short phrase.
COCO, **Flickr30K**	Provide a one-sentence caption for the provided image.

- **SNLI [4].** A large corpus of human-written premise–hypothesis pairs supporting standard NLI with the same three labels as MNLI. The dataset contains roughly 570k examples.
- **RTE [2, 3, 11, 13, 24].** A suite of textual entailment benchmarks consolidated in GLUE as binary *entailment* versus *not-entailment*. Examples are drawn from sources such as news and Wikipedia.
- **SICK [56].** Sentence pairs constructed from image and video captions. Each pair has a three-way entailment label *entailment*, *contradiction*, or *neutral* and a semantic relatedness score on a 1–5 scale. The dataset includes about 10k pairs.
- **SciTail [38].** Entailment derived from science multiple-choice questions and retrieved web sentences. Each premise–hypothesis pair is labeled *entails* or *neutral* with 27,026 total examples 10,101 *entails* and 16,925 *neutral*.

A.1.3. VLM Tasks

We evaluate our vision–language setup using LLaVA-1.5-7B [46, 47]. Our evaluation primarily follows the protocol described in Liu et al. [47]. For datasets not included in that work, we adopt the evaluation procedure from Zhang et al. [105].

- **IconQA [52].** A diagram-based question–answering benchmark that evaluates visual reasoning over abstract and textbook-style images, emphasizing symbolic understanding and multi-step reasoning.
- **VizWiz [25].** A visual question–answering dataset consisting of images captured by visually impaired users. The questions are often open-ended and exhibit real-world visual and linguistic noise. We randomly split the validation and test sets in a 2:8 ratio.
- **ChartQA [57].** A visual question–answering dataset requiring numerical and semantic reasoning over data presented in scientific charts and plots.
- **DocVQA [59].** A document-based visual question–answering dataset designed for understanding scanned documents, forms, and invoices. We randomly split the validation and test sets in a 2:8 ratio.

- **COCO [43].** A large-scale dataset for image captioning and object detection, containing diverse natural images with dense object annotations and descriptive captions. We use the captioning task with the train/val/test split protocol proposed by Karpathy and Fei-Fei [37].
- **Flickr30k [97].** A benchmark for image–text retrieval and captioning, consisting of 31K real-world images paired with multiple human-written captions. We use the captioning task with the train/val/test split protocol proposed by Karpathy and Fei-Fei [37].

A.2. Metrics

We evaluate each task using metrics that are commonly adopted for that task in the literature. For semantic segmentation, saliency estimation, and human-part segmentation, we use mean Intersection-over-Union (mIoU). Surface normal prediction is evaluated by the mean angular error between predicted and ground-truth normals. Depth estimation is measured with root mean squared error (RMSE). Edge detection is assessed using the optimal dataset scale F-measure (ods-F). For the Taskonomy, principal curvature uses RMSE, and the remaining tasks use L1 distance, following [7]. For NLI and VLM benchmarks, we report classification accuracy for sequence-classification and visual question answering tasks [47, 105]. For image captioning, we report CIDEr.

A.3. Implementation Details

For the heterogeneous vision setting, we optimize our learnable merging variants using the Adam [39] with a learning rate of 1e-4 for 100 iterations. For conventional learning-free merging methods [29, 30, 71, 90, 98, 101, 107], we choose the global scaling coefficient of the merged parameters by grid search on the validation loss. We first sweep the scale with a step size of 0.1, then run a finer sweep with a step size of 0.01 around the best scale found in the coarse search. Several baselines require additional hyperparameters beyond the global scale. For TIES [90] and KnOTS-TIES [71], we keep the top 20% of parameters by magnitude. DARE-TIES [98] and KnOTS-DARE-TIES [71] use a drop rate of 0.9 when discarding parameters. FR-Merging [107] uses a low-frequency ratio of 0.10 and a high-frequency ratio of 0.70 for frequency-based filtering. For RobustMerge [101], we follow the hyperparameter configuration reported in the original work. For merging both LLMs and VLMs, we optimized the learnable merging coefficients using AdamW [51] with a learning rate of 3×10^{-4} for 500 iterations. For LLM experiments, we used a per-task batch size of 2, whereas for VLMs the batch size was set to 1.

We merge the multimodal projector into a unified module. While learning-free baselines follow their respective merging protocols, gradient-based methods utilize learnable scalar coefficients for each projector. In NSC, these coefficients are optimized via the objective, allowing gradients calculated in the LLM to flow back and update them.

A.4. Fine-tuning Details

For the heterogeneous vision setting, we start from a ViT-B model pretrained on ImageNet-21k [15] and fine-tune it separately on each task. We train with a batch size of 32 for 40,000 iterations using a learning rate of 2e-5, weight decay of 1e-6, and a polynomial learning-rate schedule. LoRA [28] adapters with rank 16 are applied to the Q, K, V, and O projection modules of each attention layer. For both the natural language inference and vision-language task settings, LLaMA [18] and LLaVA [46, 47] were fine-tuned using LoRA with rank 16. LoRA adapters were applied to the query and value projection matrices of the self-attention modules. We used the AdamW optimizer [51] together with a cosine learning rate scheduler [50], employing a warmup phase corresponding to 6% of the total training steps. The learning rate was set to 3e-5.

For LLaVA, the vision encoder was kept frozen during fine-tuning, while the multimodal projectors were fully fine-tuned. For visual question answering tasks, we fine-tuned the model for 5 epochs. For image captioning, we fine-tuned for one epoch and treated each caption associated with the same image as an independent training sample.

B. Additional Related Work

More recent works on model merging directly target VLMs and LLMs and propose strategies tailored to these architectures [5, 17, 53, 83, 101, 101, 109, 110]. In particular, RobustMerge [101] introduces a merging method for MLLMs that exploits directional robustness in a low rank space. Zhang et al. [102] leverage the rotation symmetry of self attention layers, which substantially enlarges the equivalence set of transformer models compared to permutation based symmetries. Another line of work explicitly connects model merging with conventional multi task learning [67, 84, 93, 94], viewing merging as a mechanism for parameter sharing and representation consolidation across tasks. A complementary set of studies aims to localize task specific information in the parameters or to quantify interference within linear layers [9, 14, 72, 81], while Marczak et al. [55] further decompose the parameter space into shared and task specific subspaces. MuDSC [89] also explores heterogeneous settings with diverse dense prediction tasks for evaluating merging performance. However, it is not directly comparable to our setting because it allows branch like architectures such as ZipIt [70], whereas we focus on a strictly shared backbone.

Several methods exploit low rank structures for merging [23, 63, 64]. In particular, Panariello et al. [63] propose a core space that can be efficiently combined with existing

merging baselines. Model merging has also been studied in continual learning scenarios [19, 42, 65, 75, 80, 82, 95], where merging is used to mitigate catastrophic forgetting and to accumulate knowledge across tasks over time. Another line of work leverages intermediate activations during inference or training and uses feature responses in specific layers to guide merging process [48, 62, 87, 96]. Different from other approach, which modifies the merging procedure itself, pre merging methods [73, 103, 104] focus on constructing checkpoints that are more amenable to merging, shifting the emphasis from the merging algorithm to the pre training and finetuning pipeline.

In the supplementary material, we further compare against more recent merging baselines [29, 101, 107] to demonstrate the robustness of our algorithms.

C. More Analysis on Null-Space Compression

C.1. Null-Space Ratio Computation

Null-Space Ratio with Gram-Inverse. We briefly derive Eq. (6). Let $\boldsymbol{A} \in \mathbb{R}^{r\times d}$ denote the LoRA down-projection matrix. The row space of $\boldsymbol{A}$ is

$$\mathcal{R}(\boldsymbol{A}^\top) = \{\boldsymbol{A}^\top \boldsymbol{u} : \boldsymbol{u} \in \mathbb{R}^r\} \subset \mathbb{R}^d,$$

and its orthogonal complement is the null space

$$\mathcal{N}(\boldsymbol{A}) = \{\boldsymbol{z} \in \mathbb{R}^d : \boldsymbol{A}\boldsymbol{z} = \boldsymbol{0}\}.$$

The projection matrix onto $\mathcal{R}(\boldsymbol{A}^\top)$ is

$$\boldsymbol{P} = \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}$$

and, the projection matrix onto the null space is

$$\boldsymbol{P}_{\text{null}} = \boldsymbol{I} - \boldsymbol{P} = \boldsymbol{I} - \boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}.$$

For a feature vector $\boldsymbol{z} \in \mathbb{R}^d$, the null-space ratio is defined as

$$\omega(\boldsymbol{z}) = \frac{\|\boldsymbol{P}_{\text{null}}\boldsymbol{z}\|_2}{\|\boldsymbol{z}\|_2}.$$

We compute the squared ratio:

$$\begin{aligned}\omega(\boldsymbol{z})^2 &= \frac{\|(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{z}\|_2^2}{\|\boldsymbol{z}\|_2^2} \\ &= \frac{\boldsymbol{z}^\top(\boldsymbol{I}-\boldsymbol{P})^\top(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{z}}{\|\boldsymbol{z}\|_2^2}.\end{aligned}$$

Because $\boldsymbol{P}$ is symmetric and idempotent, $(\boldsymbol{I}-\boldsymbol{P})^\top(\boldsymbol{I}-\boldsymbol{P}) = \boldsymbol{I}-\boldsymbol{P}$. Thus,

$$\begin{aligned}\omega(\boldsymbol{z})^2 &= \frac{\boldsymbol{z}^\top(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{z}}{\|\boldsymbol{z}\|_2^2} \\ &= 1 - \frac{\boldsymbol{z}^\top\boldsymbol{P}\boldsymbol{z}}{\|\boldsymbol{z}\|_2^2} \\ &= 1 - \frac{\boldsymbol{z}^\top\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}\boldsymbol{z}}{\|\boldsymbol{z}\|_2^2}.\end{aligned}$$

Taking the square root yields

$$\omega(\boldsymbol{z}) = \sqrt{1 - \frac{\boldsymbol{z}^\top\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{A}^\top)^{-1}\boldsymbol{A}\boldsymbol{z}}{\|\boldsymbol{z}\|_2^2}}.$$

Replacing $\boldsymbol{A}$ with $\boldsymbol{A}_k$ recovers Eq. (6).

In the perspective of a LoRA-equipped neural network, the term $\boldsymbol{A}\boldsymbol{z}$ is already computed during inference. Therefore, rather than directly applying the full projection matrix, which would require constructing and multiplying by a dense $d\times d$ operator and is prohibitively expensive for modern LLMs and VLMs where d easily reaches several thousands, we instead rely on a more efficient formulation based on the Gram-inverse. Using the Gram-inverse $(\boldsymbol{A}_k\boldsymbol{A}_k^\top)^{-1}$ yields a substantially more efficient formulation. All computations remain confined to the r-dimensional LoRA subspace, avoiding the need to store large projection matrices on GPU and reducing the complexity from $\mathcal{O}(d^2)$ to $\mathcal{O}(r^2)$. Since the LoRA rank r is typically tiny relative to d (e.g., $r = 16$ while d is in the thousands), this approach significantly reduces both memory usage and FLOPs. Consequently, the Gram-inverse formulation enables efficient evaluation of the null-space ratio during LoRA merging and analysis, making it suitable for large-scale models, as demonstrated in Sec. 4.3.

From a numerical standpoint, the Gram-inverse is also stable to compute. Because $\boldsymbol{A}_k\boldsymbol{A}_k^\top$ is an $r\times r$ symmetric positive (semi-)definite matrix, its inverse can be obtained efficiently using Cholesky factorization, which offers strong numerical stability and minimal memory overhead.

Impact on Adapter Strength. We analyze how the null-space ratio controls the effective magnitude of the LoRA update. For simplicity, we omit the layer index. Let $\Delta\boldsymbol{W} = \boldsymbol{B}_k\boldsymbol{A}_k$ be a LoRA update for task $k \in \{1,\dots,K\}$.

Proposition 1 (Adapter effect lower bound)**.** *For any LoRA update $\Delta\boldsymbol{W} = \boldsymbol{B}_k\boldsymbol{A}_k$, and $\boldsymbol{z} \in \mathbb{R}^d$, the following inequality holds:*

$$\|\boldsymbol{B}_k\boldsymbol{A}_k\boldsymbol{z}\|_2 \;\geq\; C_k\sqrt{1-\omega_k^2(\boldsymbol{z})}\;\|\boldsymbol{z}\|_2,$$

where $C_k = \sigma_{\min}(\boldsymbol{B}_k)\,\sigma_{\min}(\boldsymbol{A}_k)$.

Proof. We begin with the standard singular value lower bound

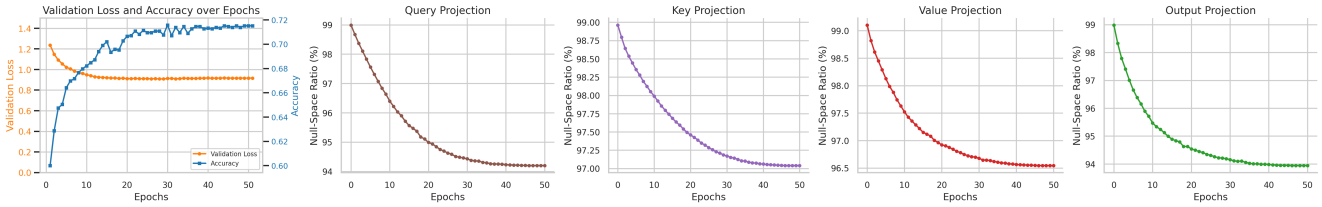
$$\|\boldsymbol{A}_k\boldsymbol{z}\|_2 \;=\; \|\boldsymbol{A}_k\boldsymbol{P}\boldsymbol{z}\|_2 \;\geq\; \sigma_{\min}(\boldsymbol{A}_k)\,\|\boldsymbol{P}\boldsymbol{z}\|_2,$$

where $\boldsymbol{P}$ is the projection matrix onto a row space of $\boldsymbol{A}_k$. Applying the same inequality to $\boldsymbol{B}_k$ yields

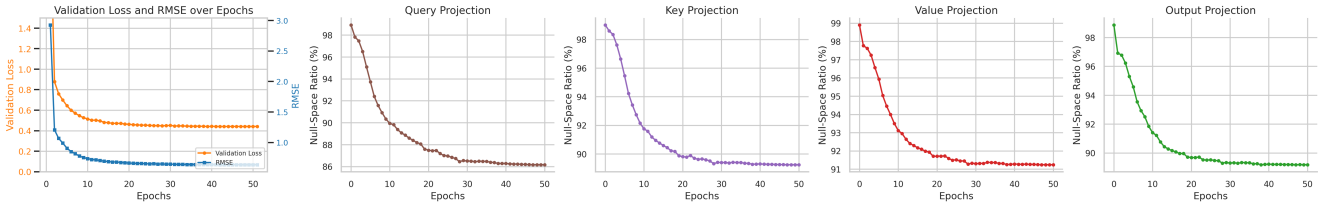
$$\|\boldsymbol{B}_k\boldsymbol{A}_k\boldsymbol{z}\|_2 \;\geq\; \sigma_{\min}(\boldsymbol{B}_k)\,\|\boldsymbol{A}_k\boldsymbol{z}\|_2.$$

Combining the inequalities gives

$$\|\boldsymbol{B}_k\boldsymbol{A}_k\boldsymbol{z}\|_2 \;\geq\; \sigma_{\min}(\boldsymbol{B}_k)\,\sigma_{\min}(\boldsymbol{A}_k)\;\|\boldsymbol{P}\boldsymbol{z}\|_2.$$

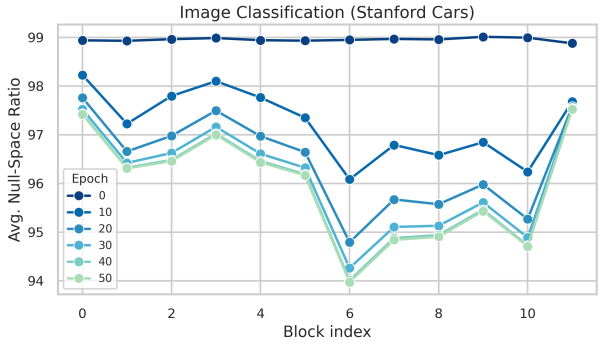


(a) Image Classification (Classification Task)

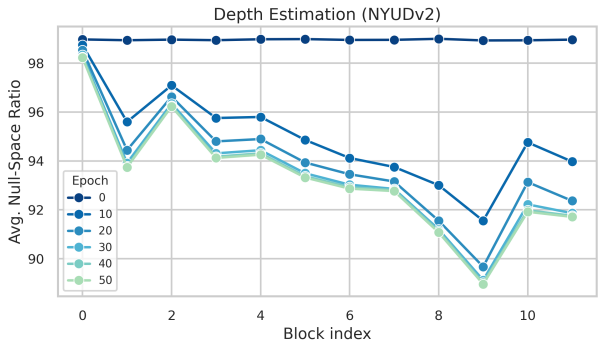


(b) Depth Estimation (Regression Task)

Figure 3. Extended visualization of validation loss, task performance, and the null-space ratio during LoRA fine-tuning on (a) image classification and (b) depth estimation. We additionally show the null-space ratio trajectories of LoRA at query, key, value projection of self-attention module within a single transformer block, further illustrating the null-space compression phenomenon.



(a) Image Classification (Classification Task)



(b) Depth Estimation (Regression Task)

Figure 4. Null-space ratio of each transformer block during LoRA fine-tuning for (a) image classification and (b) depth estimation. Values are averaged across transformer blocks, showing that null-space compression consistently occurs throughout the model.

Finally, using the definition of the null-space ratio,

$$1 - \omega_k^2(\boldsymbol{z}) = \frac{\|\boldsymbol{P}\boldsymbol{z}\|_2^2}{\|\boldsymbol{z}\|_2^2},$$

we obtain the desired result. □

This bound shows that the strength of the LoRA update is fundamentally governed by how much of the input feature $\boldsymbol{z}$ lies inside the row space of the down-projection matrix $\boldsymbol{A}$. The term $(1 - \omega_k^2(\boldsymbol{z}))$ quantifies this alignment: it is large when $\boldsymbol{z}$ contains a substantial row-space component and small when $\boldsymbol{z}$ is mostly aligned with the null space. Because the lower bound grows proportionally to $\sqrt{1 - \omega_k^2(\boldsymbol{z})}$, any decrease in the null-space ratio necessarily increases the minimum possible magnitude of the LoRA update. On multi-task settings, the lower null-space ratio guarantees higher lower-bound for the task-specific activation of the layer. Consequently, the NSC objective implicitly matches task-specific activations without requiring explicit access to them and yielding stronger task-conditioned updates.

C.2. Null-Space Compression during the Fine-tuning of LoRA Adapters

We further analyze null-space compression during fine-tuning of LoRA-equipped models. Specifically, in Fig. 3 we report the null-space ratio for different parts of the self-attention module that were analyzed in the main paper. We track the null-space ratio over 50 epochs of LoRA fine-tuning for all self-attention projections (query, key, value, and output) within a representative transformer block, and we use the 9th block for this visualization. For both image classification and depth estimation, all projections exhibit a

consistent decrease in null-space ratio over training. This trend shows that null-space compression is not confined to a particular projection but emerges across all components of the attention mechanism. Despite the small dimensionality of the LoRA subspace, we still observe a substantial reduction in null-space ratio across all LoRA-equipped modules.

To understand how null-space compression occurs across depth, we measure the null-space ratio for each transformer block during LoRA fine-tuning. Figure 4 reports the evolution of the null-space ratio over 50 epochs for all transformer blocks on both image classification and depth estimation. Across both tasks, all blocks start with relatively high null-space ratios and exhibit a gradual and consistent decrease as training progresses, indicating that null-space compression is not restricted to a few layers but occurs throughout the network. Together with the analysis in Fig. 3, these results show that our objective can be used effectively across both the components and the depth of the attention module.

D. More Experiments

D.1. Extended Analysis of Main Results

Analysis on Heterogeneous Vision Tasks. As presented in Tab. 1 of the main paper, our comparison includes recent merging methods such as EMR-Merging [29], FR-Merging [107], and RobustMerge [101]. Here, we provide a more detailed analysis of their performance across the twenty heterogeneous tasks. EMR-Merging degrades on dense prediction, yielding lower averages overall (77.0% and 79.5%). FR-Merging applies a Fourier-domain filter that emphasizes mid- to high-frequency weight components to reduce interference, which improves over vanilla baselines yet still trails the top methods (83.4%). RobustMerge is the strongest among conventional baselines, showing balanced performance across tasks with notably better preservation of semantic segmentation and depth. NSC achieves the highest overall average (92.0%) and leads on the hardest tasks (for example, 85.1% on NYUD-v2 Semseg and 76.7% on PASCAL-Context Parts), indicating more stable performance when objectives differ across classification and regression.

We observe that collapsed tasks in conventional baselines are predominantly high-level vision tasks (e.g., semantic segmentation), while low-level tasks (e.g., edge detection) tend to dominate the merged model. We posit that *inter-task affinity* serves as an implicit metric for this misalignment. Following [21], we define affinity as the relative reduction in the loss of task j after a gradient update on task i: $\text{Affinity}_{i\to j} = 1 - \frac{\mathcal{L}_j(\theta - \eta\nabla\mathcal{L}_i)}{\mathcal{L}_j(\theta)}$. Empirically, collapsed tasks show notably low affinity with the dominating tasks. For instance, in PASCAL-Context, the affinity of Edge Detection towards Semantic Segmentation is only 0.34, whereas the affinity towards Surface Normals is much higher at 0.79. This suggests that simple parameter averaging (TA, TIES) inherently fails when the dominant task introduces parameter shifts that suppress the activations required by other tasks.

While NSC does not eliminate the inherent dissimilarity between tasks, it prevents the catastrophic collapse stemming from signal vanishing. Theoretically, as proven in Proposition 1, minimizing the null-space ratio provides a mathematical guarantee on the lower bound of the adapter's signal strength ($\|\boldsymbol{B}_k\boldsymbol{A}_k z\|$). In baselines, low affinity causes task signals to cancel out and fall into the null space. In contrast, NSC explicitly optimizes coefficients to preserve these principal subspaces, serving as a safeguard to prevent feature collapse.

Analysis on VLM Benchmarks. Building on the results in Tab. 3, we further analyze per-task performance on the six multi-modal benchmarks. Among the newly added baselines, EMR-Merging [29] and FR-Merging [107] both achieve strong performance on benchmarks such as MNLI and SNLI (e.g., 96.2 and 94.2 for EMR-Merging on MNLI and SNLI, respectively), but exhibit noticeable degradation on SICK, dropping to 76.6 and 80.3. RobustMerge [101] yields the best average among all prior baselines, with consistently high scores across MNLI, SNLI, RTE, and SciTail, yet still underperforms our approach. NSC attains the highest overall average of 92.3, demonstrating that our method works well on NLI datasets compared to recent merging baselines.

Detailed Analysis on VLM Benchmarks. Building on the results in Tab. 3, we further analyze per-task performance on the six multi-modal benchmarks, where all merged results are normalized to their corresponding fine-tuned baselines. Among the recent methods, EMR-Merging [29] and FR-Merging [107] show large gains on IconQA (84.0 and 87.2, respectively), but suffer severe degradation on VizWiz (49.2 and 46.3), leading to lower overall averages of 78.7 and 76.0. As a result, both methods underperform traditional parameter-space baselines such as TA and TIES, which reach 81.4 and 81.8 on average, and also fall behind adaptive approaches like AdaMerging [91], which achieves 80.9 with single-token adaptation and 82.4 with full-token adaptation. NSC attains the highest overall average of 82.7, while maintaining strong performance, indicating that NSC extends well to multi-modal VLM settings and improves over recent baselines.

Analysis on Generalization to Unseen Tasks. Table 4 evaluates generalization on ten seen and ten unseen tasks. Expanding on the main text, we observe that the overall trends mirror those in Tab. 1. NSC attains the best averages on both splits, with 95.1% on seen, 87.1% on unseen, and 91.1% overall, indicating strong retention on seen tasks and robust transfer to tasks without checkpoints. EMR-

Table 8. Ablation on the token position used to compute the NSC objective. We compare First-Token, Last-Token, and Full-Sequence scoring on six VLM benchmarks. We report average normalized performance and optimization time. The chosen design is shaded.

Token Position Strategy			Dataset						Performance	
First-Token	Last-Token	Full-Sequence	IconQA	VizWiz	ChartQA	DocVQA	COCO	Flickr	Avg	Time (min)
		✓	59.5	82.8	78.0	87.0	91.5	97.3	82.7	83.7
	✓		60.0	82.9	77.7	87.1	91.5	97.2	**82.8**	65.1
✓			59.7	82.9	78.1	87.1	91.7	96.8	82.7	13.3

Table 9. Ablation on the input modality used to compute the NSC objective. We compare text-only, vision-only, and joint vision–language scoring on six VLM benchmarks. We report average normalized performance and optimization time. The chosen design is shaded.

Input Modality		Dataset						Performance	
Text	Image	IconQA	VizWiz	ChartQA	DocVQA	COCO	Flickr	Avg	Time (min)
✓		56.6	83.8	74.3	83.9	93.3	97.7	81.6	6.0
	✓	60.2	82.4	78.4	87.0	91.4	97.0	**82.7**	13.1
✓	✓	59.7	82.9	78.1	87.1	91.7	96.8	**82.7**	13.3

Table 10. Detailed compute cost breakdown of AdaMerging and NSC on VLM tasks. All times are in seconds and measured on a single NVIDIA A6000 GPU.

Process Step	AdaMerging	NSC (Ours)
Pre-computation (Gram-Inverse)	✗	0.7
Forward Pass	396.7	407.2
Entropy/NSC Calculation	1.1	4.1
Backpropagation + Optimizer Step	393.4	386.7

Table 11. Ablation on where the NSC objective is applied: target module and number of blocks evaluated on 20 heterogeneous vision tasks.

Targeted Projection Matrix	Number of Activated Transformer Blocks			
	12	6	3	1
QKVO	92.0	92.0	92.0	91.9
KVO	91.6	91.6	91.6	91.4
VO	92.0	92.0	92.0	91.6
O	91.7	91.7	91.8	91.7

Merging similarly degrade on heterogeneous dense tasks. FR-Merging improves some Taskonomy metrics but remains unstable across datasets. RobustMerge is the most balanced among the recent baselines. Overall, NSC provides the most consistent gains across datasets and task types.

Impact of token position and sequence length on NSC. Table 8 ablates where in the generated sequence we compute the NSC objective, comparing First-Token, Last-Token, and Full-Sequence scoring on six VLM benchmarks. All three variants achieve very similar performance, with averages, indicating that NSC is largely insensitive to the exact token position used. We attribute this robustness to the fact that NSC operates on the geometry of the LoRA parameters (through the null-space ratio) rather than relying on a particular token's semantics. Moreover, using the full sequence does not yield noticeable gains over single-token variants. In contrast, First-Token scoring avoids waiting for the full response to be generated before backpropagation, making NSC substantially more efficient in practice. We therefore adopt First-Token as our default design.

Effect of input modality in NSC. Table 9 studies which input modality is used to compute the NSC objective on six VLM benchmarks, comparing text-only, image-only, and joint vision–language inputs. Text-only scoring performs reasonably well but lags behind the configurations that use image features. Image-only and joint scoring both reach an average of 82.7, and consistently outperform text-only, showing that visual information is more critical for guiding NSC in multi-modal settings. Based on this observation, we adopt the joint vision–language configuration as our design.

Compute Accounting. In addition to Tab. 6, we provide a more detailed breakdown of the optimization costs of AdaMerging and NSC in Tab. 10. The Gram-Inverse precomputation step for NSC takes approximately 0.7 seconds, which is a one-time cost at the beginning of optimization and does not affect the per-iteration cost during training. The forward pass and backpropagation steps are comparable between the two methods, with NSC being slightly faster in backpropagation due to more efficient gradient updates. The NSC calculation itself takes about 4.1 seconds per iteration, which is more expensive than the entropy calculation in AdaMerging (1.1 seconds), but this additional cost is relatively small compared to the overall iteration time. Therefore, while NSC introduces some overhead due to its more complex objective, it remains computationally feasible and efficient for practical use in large-scale models. All results are based on VLM tasks, with additional cost analysis reported in Tab. 6, where our method demonstrates faster preparation compared to SVD-based baselines requiring pre-computation.

Robustness to target module and block count. Table 11 ablates where the NSC objective is applied in the attention stack and how many transformer blocks are activated on the

Table 12. Optimization stability of gradient-based merging methods across 5 random seeds.

Benchmark	**Method**	**Avg. Norm. Acc.** (mean $\pm$ std)
LLM Tasks	AdaMerging	90.08 ± 0.44
	NSC (Ours)	92.25 ± 0.08

Table 13. Impact of unlabeled sample size on performance of NSC on LLM tasks.

# Samples per Task	**1**	**10**	**100**	**Full Data**
Avg. Norm. Acc.	91.88 ± 0.35	92.19 ± 0.16	92.25 ± 0.12	92.25 ± 0.08

20 heterogeneous vision tasks. We vary the targeted projection matrix across QKVO, KVO, VO, and O, and sweep the number of activated blocks from all 12 layers down to only the last block. Across all configurations, the average performance remains clustered around 92.0, showing that NSC is highly robust to the choice of projection matrix. Targeting VO or QKVO slightly better performance, but the gap relative to using only O is within 0.3, which is negligible.

Variance Analysis & Data Efficiency. Since gradient-free baselines (e.g., TA [30], TIES [90]) are deterministic, we evaluate optimization stability by comparing against AdaMerging [91] on LLM tasks. As shown in Tab. 12, NSC demonstrates substantially lower variance across random seeds, achieving a standard deviation of 0.08 compared to 0.44 for AdaMerging. This indicates that NSC yields more consistent performance and is less sensitive to initialization and stochastic effects during optimization.

We further analyze data efficiency in Tab. 13. Although performance slightly decreases with fewer unlabeled samples, NSC retains near-peak accuracy even under extremely limited data regimes, achieving competitive results with as few as 10 samples per task.

Zero-shot Image Classification Benchmarks. Following task arithmetic (TA) [30], we also evaluate our method on merging eight image-classification models based on CLIP/ViT-B-32 [66]. All models are fine-tuned using LoRA with rank 16. We report performance on eight widely used classification benchmarks, SUN397 [88], Cars [40], RESISC45 [8], EuroSAT [26], SVHN [61], GTSRB [69], MNIST [41], and DTD [10]. As shown in Table 14, NSC does not always outperform AdaMerging [91] in this CLIP-based classification setting. On these traditional, classification-centric benchmarks, NSC consistently improves over learning-free baselines such as TA and TIES [30, 90], achieving higher averaged normalized accuracy across the eight datasets. However, NSC lags behind AdaMerging, which is specifically designed with entropy-based objectives. These results suggest that NSC plays a complementary role to conventional entropy-driven merging. In particular, NSC is most beneficial in regimes where entropy cannot be computed, such as regression tasks, or where computing or optimizing entropy is not efficient or effective, such as large-scale LLM and VLM settings, while entropy-based methods remain preferable on standard classification benchmarks.

Table 14. Performance comparison across visual classification tasks. Absolute accuracies (top), and normalized accuracies of merged models (bottom).

Method	**Dataset**								
	Cars	**DTD**	**EuroSAT**	**GTSRB**	**MNIST**	**RESISC45**	**SUN397**	**SVHN**	**Avg**
Finetuned	70.7	67.6	98.5	97.6	99.4	91.6	72.0	95.7	86.6
TA [30]	87.8	78.6	62.9	70.5	92.2	78.8	92.4	85.9	81.1
TIES [90]	90.6	80.4	78.8	66.1	91.6	81.2	92.5	82.5	82.9
AdaMerging [91]	90.1	76.8	87.8	86.0	94.7	85.6	90.8	82.8	86.8
NSC (Ours)	89.4	78.7	80.3	74.4	93.4	82.6	92.1	86.0	84.6