Investigating Cross-Modal Skill Injection: Scenarios, Methods, and Hyperparameters

Zhiyu Xu[1], Lean Wang[1], Yuanxin Liu[1], Lei Li[3], Hao Zhou[2], Fandong Meng [2], Jie Zhou[2], Xu Sun[1]
[1]State Key Laboratory for Multimedia Information Processing,
School of Computer Science, Peking University
[2]WeChat AI, Tencent Inc., China [3]The University of Hong Kong
zhiyu_xu@stu.pku.edu.cn, lean@pku.edu.cn, liuyuanxin@stu.pku.edu.cn,
nlp.lilei@gmail.com, {tuxzhou, fandongmeng, withtomzhou}@tencent.com
xusun@pku.edu.cn

Abstract

Vision-Language Models (VLMs) have demonstrated remarkable proficiency in general multimodal understanding; yet they struggle to efficiently acquire continually evolving domain-specific skills. Conventional approaches to enhancing VLM capabilities, such as Supervised Fine-Tuning (SFT), require extensive dataset curation and substantial computational resources. Model merging has emerged as an efficient alternative that enables the transfer of domain-specific expertise from Large Language Models (LLMs) to VLMs without incurring additional training data requirements or significant computational overhead. Unlike conventional merging of homogeneous LLMs, which mainly aggregates existing capabilities, cross-modal skill injection aims to induce emergent cross-modal capabilities by integrating a domain-expert LLM into a VLM. However, existing research lacks a systematic analysis of the applicability and methodology of cross-modal skill injection. In this study, we investigate cross-modal skill injection across three main aspects: scenarios, methods, and hyperparameters. For scenarios, we find that cross-modal skill injection generally performs well in instruction-following and cross-lingual settings, yet struggles with mathematical reasoning. For methods, we find that classic approaches such as TA and DARE consistently achieve superior performance over alternative merging methods. We also provide a systematic and quantitative analysis of the hyperparameter tuning that these classic methods critically depend on.

1 Introduction

Vision-Language Models (VLMs) have garnered increasing attention for their ability to jointly process and comprehend visual and textual information (Alayrac et al., 2022; Li et al., 2023b; Liu et al., 2024a). Despite strong general performance, VLMs remain limited on specialized tasks such as visual mathematical reasoning and multilingual understanding (Lu et al., 2024a; Zhang et al., 2024a).

Fine-tuning on specialized multimodal datasets faces significant challenges despite being a prevalent strategy to enhance domain-specific capabilities (Guo et al., 2025; Srinivasan et al., 2021). First, fine-tuning VLMs requires substantial computational resources. Although Parameter-Efficient Fine-Tuning (PEFT) methods, such as LoRA and QLoRA (Hu et al., 2022; Dettmers et al., 2023), have effectively mitigated computational costs, acquiring sufficient high-quality training data remains a persistent bottleneck. High-quality vision datasets for specialized tasks are scarce, making it necessary to laboriously construct balanced datasets while considering factors such as data mixing ratios and domain coverage (Li et al., 2025b).

Model merging offers a promising alternative for integrating expert capabilities into VLMs without extensive dataset construction or additional retraining (Ilharco et al., 2023; Yadav et al., 2023; Yu et al., 2024). However, merging guidelines developed for homogeneous models may not apply to cross-modal skill injection. First, cross-modal skill injection is asymmetric: rather than combining peer models symmetrically, it integrates a domain-expert LLM into a VLM backbone. Second, while homogeneous merging typically aims to aggregate existing expertise, cross-modal skill injection focuses on enabling new cross-modal capabilities to emerge. For example, merging a mathematics expert with a VLM could yield visual mathematical reasoning capabilities, such as solving geometry problems, that neither model previously possessed.

Despite its potential, cross-modal skill injection remains underexplored. To establish comprehensive guidelines for cross-modal skill injection, we systematically investigate the transfer of expert LLM capabilities to VLMs from three main aspects: scenarios, methods, and hyperparameters.

For scenarios, we examine three distinct set-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12546–12563
July 2-7, 2026

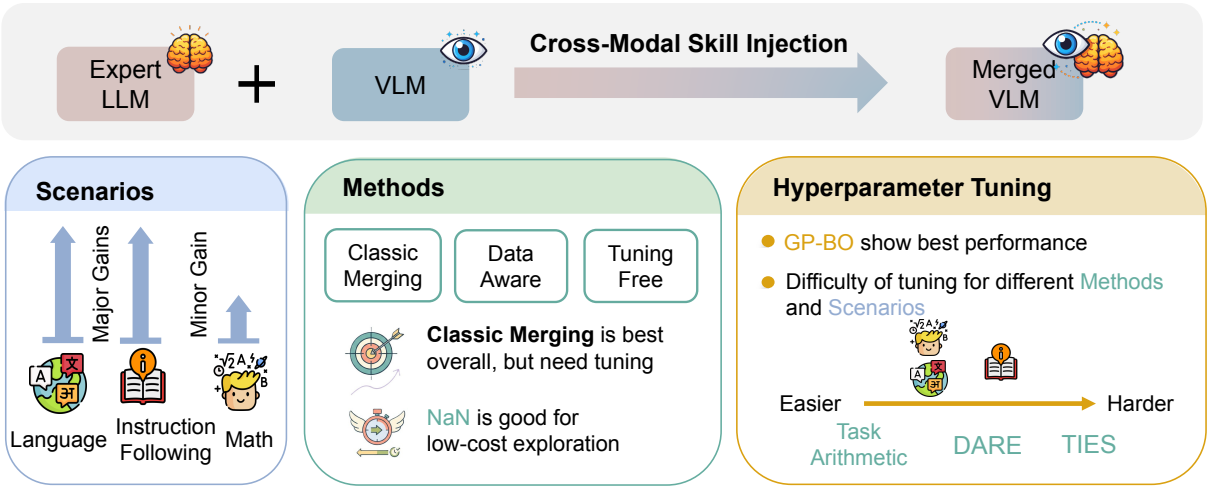


Figure 1: Overview of our work investigating cross-modal skill injection across three dimensions: **scenarios** (language ability, mathematical, instruction following), **methods** (classic, data-aware, and tuning-free merging), and **hyperparameters**. Our findings reveal that classic methods consistently outperform alternatives, language and instruction-following capabilities transfer more readily than mathematical reasoning, and we further provide quantitative analysis of the hyperparameter optimization landscape.

tings: language ability, mathematical reasoning, and instruction following. Our extensive experiments across six benchmarks demonstrate that VLMs can successfully inherit specialized capabilities from expert LLMs in language ability and instruction-following scenarios. However, we observe that transferring mathematical reasoning proves more challenging than transferring language or instruction-following abilities.

For methods, we evaluate three categories of model merging approaches. Our results show that classic merging methods consistently yield superior performance. Certain tuning-free methods also demonstrate surprisingly competitive performance; for instance, NaN (Si et al., 2025) achieves the second-best overall results.

For hyperparameters, since hyperparameter tuning is essential for classic methods and constitutes their primary practical cost, we conduct a thorough analysis of the hyperparameter landscape. The search space is low-dimensional but exhibits slight multimodality. Our analysis identifies GP-BO (Jones et al., 1998) as the most effective optimizer, while local directional search methods, although competitive in performance, are more vulnerable to local optima.

Our main contributions are as follows:

- **Scenario Analysis.** We systematically investigate three representative scenarios for cross-modal skill injection: language ability, mathematical reasoning, and instruction following, providing insight into which scenarios are better suited to cross-modal skill transfer.

- **Method Comparison.** We conduct a comprehensive evaluation of nine merging methods across three categories (classic, data-aware, and tuning-free) on six benchmarks, offering practical guidelines for method selection based on available resources.

- **Hyperparameter Analysis.** We provide a quantitative analysis of the hyperparameter optimization landscape for classic merging methods, comparing the effectiveness of different optimization strategies.

2 Preliminary

2.1 Model Merging Methods

Model merging is closely related to the notion of task vectors (Ilharco et al., 2023), which views the parameter change induced by fine-tuning as a vector in parameter space that can encode task-specific behavior. Formally, the task vector τ is defined as the difference between the parameters of a fine-tuned expert LLM ($\theta_{\text{finetuned}}$) and those of the base model (θ_{base}):

$$\tau = \theta_{\text{finetuned}} - \theta_{\text{base}} \quad (1)$$

Task Arithmetic constructs the merged model by adding a linear combination of these task vectors

to the base model:

$$\theta_{\text{merged}} = \theta_{\text{base}} + \sum_{i=1}^{n} \lambda_i \tau_i \quad (2)$$

where n represents the number of models being merged, and λ_i denotes the merging coefficient for the i-th task vector.

Classic Merging. The approach of linearly combining task vectors to yield a multi-skilled model is known as Task Arithmetic (TA) (Ilharco et al., 2023). In this paper, we refer to this method and its variants, such as TIES (Yadav et al., 2023) and DARE (Yu et al., 2024), as classic merging, characterized by combining task vectors (or their variants) with tunable merging coefficients. In addition to model coefficients, some variants (Davari and Belilovsky, 2024; Deep et al., 2024; Goddard et al., 2024) introduce a density hyperparameter to control the sparsity of task vectors. For instance, TIES-Merging (Yadav et al., 2023) prunes low-magnitude updates and resolves sign conflicts, while DARE (Yu et al., 2024) employs random sparsification followed by rescaling. These classic merging methods typically rely on hyperparameter tuning on in-domain validation data to identify the optimal merging coefficient and, in some cases, the density parameter.

Recent works have improved upon classic merging in two directions.

Data-Aware Merging. This line of work leverages training data from each model's capability domain to refine the merging process. For instance, Fisher Merging (Matena and Raffel, 2022) weights parameters based on their task-specific importance estimated from the empirical Fisher information, while RegMean (Jin et al., 2023) reduces differences between the intermediate representations of different models at each layer through closed-form linear regression. Intuitively, by incorporating auxiliary training data, these approaches exploit richer information and should yield better performance. However, as we will show in Section 3, for cross-modal skill injection, the empirical gains from data-aware methods are often marginal, suggesting that, in cross-modal scenarios, practitioners can safely forgo the overhead of data collection without sacrificing much performance.

Tuning-Free Merging. Tuning-free methods derive merging recipes directly from model parameters, avoiding hyperparameter tuning and the need for external data. One line of work transforms task vectors through subspace operations. WUDI (Cheng et al., 2025) exploits the observation that, in a linear layer, task vectors approximately span the corresponding input subspace, and carries out merging in this subspace. TSV (Gargiulo et al., 2024) constructs layer-wise low-rank subspaces from the SVD of task matrices, and decorrelates singular directions before merging. Another line of work estimates merging coefficients directly. MetaGPT (Zhou et al., 2024) derives closed-form scaling coefficients under the assumption of local linearity and approximate task-vector orthogonality, while NAN (Si et al., 2025) estimates coefficient from inverse parameter norms. Overall, these methods are plug-and-play and appealing when data access is limited or tuning budgets are tight.

2.2 Cross-Modal Skill Injection

Similar to LLM merging, expert LLMs can be merged into the language encoder backbone of a VLM to transfer specialized capabilities, which we refer to as cross-modal skill injection.

Formally, we define cross-modal skill injection as follows: Given a VLM, denoted as $\mathcal{M}_{vlm} = (\mathcal{E}_v, \mathcal{L}_{base})$, where $\mathcal{E}_v$ is the vision encoder, and $\mathcal{L}_{base}$ is the LLM backbone, our goal is to endow $\mathcal{M}_{vlm}$ with domain-specific expert capabilities by integrating a domain expert LLM $\mathcal{L}_{exp}$, without modifying $\mathcal{E}_v$ or requiring full finetuning of $\mathcal{M}_{vlm}$. Specifically, we aim to construct a merged model

$$\mathcal{M}_{merged} = (\mathcal{E}_v, f(\mathcal{L}_{base}, \mathcal{L}_{exp}))$$

where $f(\cdot)$ denotes a merging algorithm that fuses the parameters of the backbone and expert LLMs.

Cross-modal skill injection inherits the core advantages of LLM merging: minimal training overhead, rapid domain specialization, and independence from large-scale datasets. However, it differs fundamentally from conventional same-modality LLM merging in its functional asymmetry. In cross-modal skill injection, the VLM provides visual grounding, while the expert LLM contributes specialized capabilities. Traditional LLM merging instead combines peer models with comparable roles, primarily aiming to aggregate and preserve capabilities within a shared modality.

Rather than merely combining existing skills, cross-modal skill injection seeks to induce capabilities that neither parent model possesses in isolation. The resulting capability is emergent because the injected textual expertise must become usable under

visual input, rather than remaining a purely text-side skill. Unlike same-modality merging, where the combined capabilities remain within a single representational space, cross-modal skill injection requires the interaction between visual understanding and textual expertise to arise through the merging process itself and result in a cross-modal ability.

Because cross-modal skill injection aims to induce capabilities that arise only through the interaction between expert knowledge and visual understanding, it faces challenges beyond those in conventional same-modality LLM merging. In specialized visual domains such as medical imaging, legal document analysis, and scientific figure understanding, paired image-text data for validation is often scarce or expensive to collect, which imposes tighter constraints on hyperparameter tuning. Moreover, merging methods designed to reconcile conflicts among peer models may be less effective in this setting, because the LLM and VLM play asymmetric roles rather than contributing comparable capabilities within a shared modality.

3 Scenario and Methods for Cross-Modal Skill Injection

In this section, we derive practical guidelines regarding which scenarios and merging strategies are more suitable for cross-modal skill injection. Specifically, we focus on three representative visual capability scenarios: language ability, mathematical reasoning ability, and instruction-following ability. These are among the most commonly evaluated and practically relevant abilities in prior work (Chen et al., 2025b; Yang et al., 2025).

Model Settings. We conduct experiments using publicly available models from Hugging Face. Our VLM cover a diverse set of architectures, including Idefics2 (Laurencon et al., 2024), LLaVA with both Mistral and LLaMA backbones (Liu et al., 2023b,a, 2024b), and Qwen2-VL (Wang et al., 2024). For each scenario, we pair the VLM with a domain-expert LLM that shares the same base architecture. Specifically, for language understanding, we use Mistral-7B-v0.3-Chinese-Chat and Llama-3-ELYZA-JP-8B to inject Chinese and Japanese capabilities, respectively. For mathematical reasoning, we use the DART-Math series built on LLaMA and Mistral backbones. For instruction following, we pair Qwen2-VL with Qwen2-Instruct and Idefics2-base with Mistral-7B-Instruct. Appendix A lists the full checkpoint names and pairing.

Evaluation Settings. We employ six benchmarks to evaluate the merged VLMs under different scenarios. For visual mathematical reasoning, we use MathVista (math subset) (Lu et al., 2024a) and MathVerse (Zhang et al., 2024c); for language understanding with visual inputs, CMMMU (Chinese) (Zhang et al., 2024a) and JMMMU (Japanese) (Onohara et al., 2025); and for visual instruction following, MIA-Bench (Qian et al., 2025) and WildVision (Lu et al., 2024b). For each benchmark, 20% of the samples are randomly held out for hyperparameter tuning, while the remaining 80% are used exclusively for testing.

To estimate the cost of merging and hyperparameter tuning, we conduct experiments on a single A800 GPU, utilizing the Python package MergeKit (Goddard et al., 2024) for merging and lmms-eval (Zhang et al., 2024b; Li et al., 2024) for evaluation. The total cost is computed by summing the GPU cost and the OpenAI API cost for evaluating the merged models. The reported total duration includes both merging and evaluation time.

Methods Settings. We employ the three categories of merging methods introduced in Section 2.1; further details on hyperparameter settings and calibration data are provided in the Appendix C. (1) *Classic merging methods* (Task Arithmetic, DARE, TIES-Merging): We perform grid search over VLM and LLM coefficients in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$; for methods requiring a density parameter, we additionally search over $\{0.2, 0.4, 0.6, 0.8\}$. (2) *Data-aware methods* (Fisher, RegMean): We use 500 calibration samples drawn from corresponding domain-specific datasets. Specifically, the VLM is calibrated on a vision-instruction dataset (LLaVA-Instruct-150K) (Liu et al., 2023b,a, 2024b), while the expert LLM is calibrated on scenario-matched text data: Alpaca-Zh for Chinese (Cui et al., 2024), Japanese-Alpaca-Data for Japanese (fujiki, 2023), Competition-Math mixed with GSM8K for mathematical reasoning (Lightman et al., 2023; Cobbe et al., 2021), and Dolly-15K for instruction following (Conover et al., 2023). (3) *Tuning-free methods* (WUDI, TSV, MetaGPT, NaN): These methods can be applied without additional hyperparameter tuning or reliance on external data.

Finding 1: Cross-modal skill injection is more effective for transferring language and instruction-following abilities than for transferring reasoning and mathematical abilities. As

Method	CMMMU	JMMMU	MathVista (math)		MathVerse		MIA-Bench		WildVision		Avg
	Mistral	LLaMA	Mistral	LLaMA	Mistral	LLaMA	Qwen2	Idefics2	Qwen2	Idefics2	
TA	25.0↑2.4	**42.1**↑5.7	26.7↑1.4	29.2↑3.4	17.1↑1.9	15.0↓1.1	**76.1**↑43.1	75.9↑66.1	**50.1**↑34.0	22.8↑19.0	**38.0**
DARE	**28.5**↑5.9	39.0↑2.6	**29.0**↑3.7	30.6↑4.8	16.1↑0.9	**17.0**↑0.9	75.4↑42.4	74.3↑64.5	47.5↑31.4	19.9↑16.1	37.7
TIES	27.4↑4.8	40.9↑4.5	23.7↓1.6	27.1↑1.3	14.5↓0.7	15.3↓0.8	74.7↑41.7	76.3↑66.5	45.1↑29.0	**23.3**↑19.5	36.8
Fisher	25.3↑2.7	36.0↓0.4	22.3↓3.0	**32.4**↑6.6	15.9↑0.7	13.9↓2.2	68.6↑35.6	**78.3**↑68.5	23.1↑7.0	**23.3**↑19.5	33.9
RegMean	28.1↑5.5	40.5↑4.1	27.1↑1.8	26.0↑0.2	**17.9**↑2.7	16.0↓0.1	69.1↑36.1	4.9↓4.9	26.4↑10.3	2.3↓1.5	25.8
WUDI	25.4↑2.8	35.6↓0.8	24.8↓0.5	29.4↑3.6	11.3↓3.9	16.2↑0.1	66.8↑33.8	5.3↓4.5	25.4↑9.3	3.1↓0.7	24.3
TSV	24.0↑1.4	38.3↑1.9	23.5↓1.8	29.7↑3.9	16.0↑0.8	15.3↓0.8	54.8↑21.8	72.4↑62.6	15.2↓0.9	16.5↑12.7	29.9
MetaGPT	28.1↑5.5	37.1↑0.7	18.4↓6.9	25.3↓0.5	12.3↓2.9	15.4↓0.7	60.6↑27.6	76.1↑66.3	36.3↑20.2	22.9↑19.1	33.3
NaN	26.5↑3.9	40.5↑4.1	23.7↓1.6	29.4↑3.6	17.3↑2.1	**17.0**↑0.9	73.8↑40.8	73.0↑63.2	47.8↑31.7	21.5↑17.7	37.1
Base VLM	22.6	36.4	25.3	25.8	15.2	16.1	33.0	9.8	16.1	3.8	20.4

Table 1: Performance comparison across merging methods, datasets, and models, with average scores computed for each method. Colored deltas indicate improvement (green) or degradation (red) relative to the base VLM. **Bold**: best; underline: second best. Key observations: (1) language and instruction-following abilities show substantial improvements, while mathematical reasoning remains challenging; (2) classic methods (TA, DARE) achieve the best overall performance, and NaN offers a competitive tuning-free alternative.

shown in Table 1, the most substantial improvements are observed in language and instruction-following benchmarks, with average gains of 3.4 and 28.1 absolute points, respectively. Instruction following is particularly favorable for merging, as such capabilities are largely modality-agnostic, enabling effective transfer to base VLMs. Notably, merging Idefics2-base with an instruction-following LLM expert achieves over 70 on MIA-Bench, surpassing the fine-tuned Idefics2 (56.4).

In contrast, mathematical ability transfer yields much smaller and less consistent gains. On MathVista, no significant improvement is observed on the full benchmark (see Appendix E); even on the math-specific subset, the average gain is only 1.05 absolute points. On MathVerse, more than half of the merging configurations are detrimental. Overall, DARE is the only method that yields consistent improvements in mathematical scenarios, while more than 40% of merged models are unable to outperform the original VLM.

One plausible explanation of less effective merging outcomes in math scenario is that visual mathematical reasoning is a highly entangled capability, demanding simultaneous coordination of visual perception (e.g., reading diagrams) and multi-step logical reasoning. Such entwinement may be difficult to reconstruct through simple parameter-space interpolation. Moreover, although some mathematical reasoning skills do transfer, these gains are often offset by degraded visual understanding. This interpretation is supported by the full MathVista breakdown (see Appendix E): merging with a math-finetuned LLM often modestly improves the "Math" subset while hurting the more perception-heavy "General" subset. Concretely, merged models become better at reasoning-heavy questions such as "Find the length of AC in the isosceles triangle ABC," yet lose accuracy on visually dependent questions such as "Does Aqua have the minimum area under the curve?"

Finding 2: Classic merging methods consistently outperform other approaches with superior stability. NaN offers a viable low-cost option for preliminary exploration, though with modest performance trade-offs. Among all merging methods, classic merging yield the strongest and most stable results (Table 1). Task Arithmetic outperforms TIES on average, while DARE demonstrates the best consistency, being the only method that yields consistent improvements across all scenarios. These methods do require domain-specific visual-text data and hyperparameter tuning (Table 2), yet this investment consistently translates into superior performance.

For data-aware merging methods, RegMean achieves better results than Fisher while also being more efficient, as it requires only activation information without gradient computation. RegMean even achieves the highest accuracy among all methods on the MathVerse benchmark, though its overall average still falls short of classic merging methods and NaN.

Among tuning-free methods, which require neither additional data nor hyperparameter tuning,

NaN stands out as the most effective. While its performance is slightly inferior and less stable compared to classic methods, NaN serves as a practical low-barrier entry point, enabling practitioners to quickly gauge whether model merging is promising for a new scenario before committing to more resource-intensive hyperparameter search. Moreover, NaN is among the most efficient approaches: along with MetaGPT, it is the least time-intensive and consumes only a fraction of WUDI and TSV's total computing time while surpassing their accuracy (see Appendix G). This combination of minimal overhead and reasonable performance makes NaN particularly well suited for rapid feasibility assessment in unfamiliar domains.

In summary, classic merging methods remain the gold standard when accuracy and stability are paramount, provided that domain visual data is available and hyperparameter tuning is feasible. For scenarios requiring quick preliminary exploration with minimal overhead, NaN offers a reasonable first-pass solution to assess merging potential before deeper investment. When domain text data is accessible, RegMean is preferable to Fisher, though its results may not match classic methods or NaN despite the additional cost.

Dataset	Total cost (dollars)	Total Duration
CMMMU	2040.21	29h 36m 17s
JMMMU	1976.88	29h 43m 15s
MathVista	1644.35	24h 13m 26s
MathVerse	3599.98	45h 49m 2s
Miabench	2008.87	19h 14m 30s
Wildvision	8044.11	45h 0m 22s

Table 2: Average cost and duration of hyperparameter tuning for DARE on each dataset using an A800 GPU, highlighting the substantial computational overhead of the optimization process.

4 Analysis of Hyperparameter Optimization Landscape and Strategies

As discussed in Section 3, hyperparameter-tuning merging methods consistently achieve strong results, but they can be time-consuming and resource-intensive. Therefore, it is essential to provide guidelines for effective hyperparameter tuning.

4.1 Revisiting the Sum-to-One Constraint

Existing approaches often constrain the sum of merging coefficients to 1 (Chen et al., 2025a). However, this constraint is largely intuitive and lacks empirical validation. Our experiments reveal that while the sum-to-one constraint proves effective in most scenarios, it can lead to substantial performance degradation in others.

Scenario	TA		DARE		TIES	
	$S{=}1$	$S{\approx}1$	$S{=}1$	$S{\approx}1$	$S{=}1$	$S{\approx}1$
CMMMU	0	0	0	0	0	0
JMMMU	3.6	0	3.6	0	0.9	0.9
MVis-L	0	0	0	0	0	0
MVis-M	0	0	2.6	2.6	0	0
MV-L3	5.3	0	7.8	2.6	1.5	0
MV-M	2.7	0	13	0	4.5	4.5
MIA-I2	1.5	1.5	3.3	1.3	0	0
MIA-Q2	1.2	0	0	0	1.9	0
WV-I2	25	0	19	8.1	13	13
WV-Q2	0	0	5.7	0	13	0

Table 3: Relative regret (%) under constrained search spaces. Restricting the coefficient sum S to approximately 1 leads to substantial performance degradation in certain scenarios (e.g., WildVision). $S{\approx}1$: $S \in [0.8, 1.2]$. Abbreviations—MV: MathVerse, MVis: MathVista, MIA: MIA-Bench, WV: WildVision; L3: LLaMA3, M: Mistral, I2: Idefics2, Q2: Qwen2.

Let $S := \lambda_{\text{VLM}} + \lambda_{\text{LLM}}$ denote the sum of merging coefficients. To investigate whether restricting the search space around $S = 1$ suffices to identify the optimal hyperparameters, we introduce the notion of *relative regret*. Specifically, we first identify the global optimum over the full coefficient grid obtained in Section 3, then compute the best achievable performance when the search is restricted to $S = 1$ or $S \in [0.8, 1.2]$. The relative regret is defined as the percentage of performance degradation relative to the global optimum caused by the restricted search space.

As shown in Table 3, restricting the search to $S \approx 1$ can be suboptimal. In certain scenarios such as WildVision and MathVerse, the $S = 1$ constraint leads to substantial performance degradation, with relative regret as high as 25%, and relaxing S to $[0.8, 1.2]$ provides only limited mitigation. While most other cases exhibit regret within 5% under the $S = 1$ constraint, and often drop to zero when the search space is relaxed to $S \in [0.8, 1.2]$, the existence of such high-regret cases demonstrates the need for a broader search space.

4.2 Benchmarking Hyperparameter Optimization Algorithms

The demand for comprehensive hyperparameter search is significant, particularly when the optimal merging coefficients do not adhere to the $S = 1$ constraint. Given that the evaluation cost of merged

models is substantial, employing a sample-efficient hyperparameter optimization algorithm is crucial to minimize the number of trials. In this section, we evaluate the performance of various hyperparameter optimization algorithms in the context of cross-modal skill injection.

Optimization Algorithms. To ensure a comprehensive evaluation of the optimization landscape, we consider a diverse spectrum of derivative-free algorithms. We include *Random Search* and *Sobol Sequences* (Sobol, 1967), the latter providing low-discrepancy quasi-random coverage, as reference methods. Beyond these, we evaluate *CMA-ES* (Hansen and Ostermeier, 2001), which adapts its sampling covariance during evolution, and *GP-BO* (Jones et al., 1998), which fits a Gaussian process surrogate to guide the search. We also examine **direct-search methods** (Kolda et al., 2003), including *Pattern Search* (Torczon, 1997), a coordinate-descent variant, and *Powell's Method* (Powell, 1964), which relies on sequential line searches. Comparing these local direct-search methods with global optimizers (e.g., GP-BO) allows us to more thoroughly examine whether the hyperparameter landscape contains multiple local optima: a substantial performance gap would suggest that greedy search methods can easily become trapped in suboptimal regions of the search space.

Evaluation Metric. We measure tuning performance using the *normalized regret* after k evaluations, defined as

$$r_k = \frac{g_{\max} - g^\star(k)}{g_{\max} - g_{\min}},$$

where $g_{\max}$ and $g_{\min}$ denote the global maximum and minimum of the objective function, respectively, and $g^\star(k)$ is the best objective value observed within the first k evaluations. The normalized regret satisfies $r_k \in [0, 1]$, with $r_k = 0$ indicating that the global optimum has been found.

To compare the effectiveness of different optimization algorithms, we report *regret-over-random* (RoR), defined as

$$\mathrm{RoR}_k = \frac{r_k^{\text{method}}}{r_k^{\text{random}}},$$

which normalizes each method's regret by that of Random Search. This ratio eliminates the influence of varying optimization difficulty across datasets, allowing for direct comparison: $\mathrm{RoR}_k < 1$ indicates the method outperforms Random Search, while $\mathrm{RoR}_k > 1$ indicates underperformance.

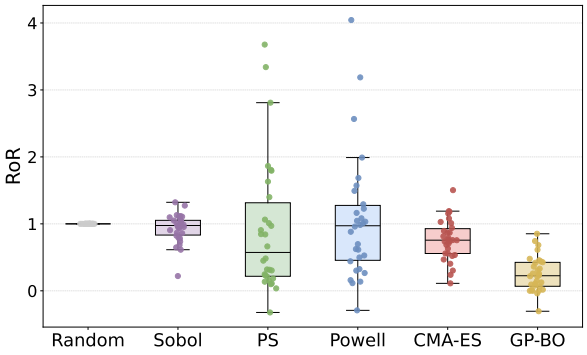


Figure 2: Distribution of regret-over-random (RoR) across optimization algorithms. Values below 1 indicate better performance than random search. GP-BO achieves the lowest median RoR, demonstrating superior sample efficiency. Pattern Search shows competitive mean performance but also high variance across runs due to sensitivity to initialization.

Experimental Setup. We set the evaluation budget k to balance computational cost and search effectiveness: $k = 40$ for Task Arithmetic, which involves two hyperparameters, and $k = 60$ for TIES and DARE, which involve three. All optimizers employ a multi-start strategy, restarting from a new random initial point upon convergence or local budget exhaustion until the total budget is depleted. To ensure fair comparison and reduce variance from random initialization, we conduct 10 independent runs per optimizer with different random seeds and report averaged results.

All hyperparameter searches are conducted on the validation set. The optimization trajectories on the validation and test sets exhibit strong agreement, with normalized regret decreasing consistently on both as the search progresses. (See Appendix H) This close correspondence justifies our use of the validation set as a low-cost proxy, avoiding the prohibitive expense of repeated test-set evaluations while preserving the reliability of our conclusions.

4.2.1 Comparison of Optimization Algorithms

Among all methods evaluated, Gaussian Process Bayesian Optimization (GP-BO) consistently achieves the lowest median regret-over-random (Figure 2), showing the strongest sample efficiency for model merging across our evaluations.

Pattern Search attains competitive mean perfor-

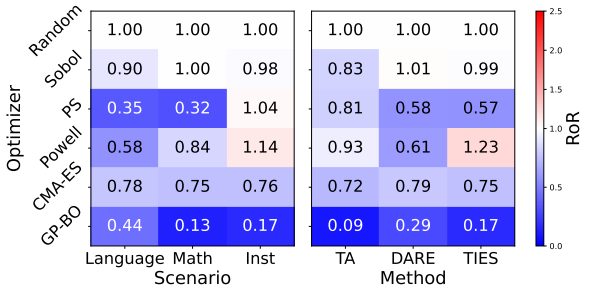


Figure 3: Regret-over-random (RoR) heatmap across optimization algorithms, scenarios and merging methods. Lower values (blue) indicate better performance. Instruction-following tasks show larger gaps between local and global methods.

mance but exhibits high variance across runs. As a local search method that greedily descends along coordinate directions, it is prone to converging to local optima, and its sensitivity to initialization reflects the presence of multiple local optima in the hyperparameter landscape. Nevertheless, its strong average performance suggests that while the objective surface is non-convex, the number of local optima remains limited and their quality is relatively high, rendering the space reasonably tractable despite lacking a unique global optimum.

CMA-ES shows limited effectiveness in our experiments, as it is designed for higher-dimensional problems with larger evaluation budgets. In our low-dimensional setting with only 2–3 hyperparameters and 40–60 evaluations, CMA-ES cannot complete enough generations for its covariance matrix adaptation to converge, and its population-based sampling is less efficient than the sequential, model-guided search used by GP-BO. Powell's method, another gradient-free local optimizer, shows performance broadly comparable to that of Pattern Search, but its overall results are slightly weaker.

4.2.2 Optimization Landscape Across Tasks and Merging Methods

We further analyze how optimization difficulty varies with the combination of merging method and target task, as shown in Figure 3.

Across Task Domains. Instruction-following tasks are harder to optimize with local methods than mathematical and language tasks. For mathematical and language tasks, local methods (Pattern Search, Powell) achieve performance comparable to global optimizers (GP-BO, CMA-ES). In contrast, for instruction-following tasks, local methods suffer significant degradation while global methods remain unaffected, suggesting the presence of local optima that trap greedy searches.

Across Merging Methods. Among the three merging methods, Task Arithmetic is the easiest to optimize, followed by DARE, with TIES being slightly more challenging. Task Arithmetic benefits from a two-dimensional search space that requires fewer optimization rounds, making it straightforward to optimize in absolute terms, despite some degradation observed with local methods. DARE, although involving three hyperparameters, exhibits a smooth optimization landscape where local methods still achieve competitive performance. TIES presents somewhat more difficulty, with Powell's method suffers notable performance degradation.

5 Related Work

VLM Domain Adaptation While supervised fine-tuning (SFT) can adapt VLMs to specific domains such as medicine and mathematics (Li et al., 2023a; Shi et al., 2024), it remains resource-intensive and time-consuming. Parameter-efficient fine-tuning (PEFT) offers an economical alternative by training only a small subset of parameters, such as adapters, LoRA, or soft prompts, while keeping pretrained weights frozen (Hu et al., 2022; Dettmers et al., 2023; Sung et al., 2022). However, PEFT still requires substantial curated image–text pairs, which can be expensive to obtain or scarce in specialized domains. In contrast, cross-modal skill injection (discussed in this paper) offers a promising alternative that efficiently transfers specialized capabilities without requiring large-scale vision data or incurring extra training overhead.

VLM Merging With the growing prominence of VLMs, model merging has been extended from purely language-based settings to multimodal contexts. Most existing work focuses on VLM-to-VLM merging, whether under homogeneous or heterogeneous backbones, exploring techniques such as uncertainty-guided selection, module-level recipes, fine-grained parameter splicing, and cross-architecture alignment (Qu et al., 2025; Zhu et al., 2025; Dai et al., 2025; Du et al., 2025).

More closely related to our work is the VLM-LLM merging paradigm, where a domain-specialized LLM is merged into a VLM backbone to induce cross-modal capabilities. Chen et al. (2025a) inject reasoning abilities into VLMs and investigate how perception and reasoning are dis-

tributed across layers. Jiang et al. (2026) integrate a code-specialized LLM with a VLM to enable multimodal code generation. Li et al. (2025a) merge text-based reward models into VLMs to construct vision-language reward models that preserve the original preferences.

However, these prior efforts remain task-specific (targeting reasoning, code, or reward modeling) and do not systematically examine when cross-modal skill injection succeeds, which merging methods are most effective, or how sensitive performance is to hyperparameter choices. In contrast, we present a comprehensive study across diverse scenarios, merging methods, and hyperparameter choices within a unified framework.

6 Conclusion

We systematically investigate cross-modal skill injection for transferring expert LLM capabilities to VLMs. Our experiments reveal that language and instruction-following abilities transfer effectively, while mathematical reasoning remains challenging. Among merging methods, classic merging (i.e., TA, DARE) achieves the best performance, and NaN offers low-cost tuning-free exploration. GP-BO proves most effective for hyperparameter optimization. Our findings provide practical guidelines for efficient cross-modal model merging.

Limitations

Due to computational resource constraints, our study focuses exclusively on visual–language modalities and does not extend to other modalities such as audio or video. We leave the investigation of cross-modal skill injection across a broader range of modalities to future work.

Risks and Ethical Considerations

This work is primarily methodological and does not involve user data or deployment in high-stakes decision-making. We do not identify significant direct risks. We encourage future work to assess downstream fairness, privacy, and safety impacts in application-specific settings.

Acknowledgements

We thank the anonymous reviewers for their insightful comments and helpful suggestions. This work was supported in part by the National Natural Science Foundation of China under Grant No. 92470205. Xu Sun is the corresponding author.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc.

Shiqi Chen, Jinghan Zhang, Tongyao Zhu, Wei Liu, Siyang Gao, Miao Xiong, Manling Li, and Junxian He. 2025a. Bring reason to vision: Understanding perception and reasoning through model merging. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 9803–9817. PMLR.

Zhipeng Chen, Kun Zhou, Liang Song, Wayne Xin Zhao, Bingning Wang, Weipeng Chen, and Ji-Rong Wen. 2025b. Extracting and combining abilities for building multi-lingual ability-enhanced large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17574–17591, Suzhou, China. Association for Computational Linguistics.

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. 2025. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *Forty-second International Conference on Machine Learning*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Yiming Cui, Ziqing Yang, and Xin Yao. 2024. Efficient and effective text encoding for chinese llama and alpaca. *Preprint*, arXiv:2304.08177.

Yang Dai, Jianxiang An, Tianwei Lin, Hongyang He, Hongzhe Huang, Wenqiao Zhang, Zheqi Lv, Siliang Tang, and Yueting Zhuang. 2025. Graft: Integrating the domain knowledge via efficient parameter synergy for mllms. *ArXiv*, abs/2506.23940.

MohammadReza Davari and Eugene Belilovsky. 2024. Model breadcrumbs: Scalable upcycling of finetuned foundation models via sparse task vectors merging. In *ICML 2024 Workshop on Foundation Models in the Wild*.

Pala Tej Deep, Rishabh Bhardwaj, and Soujanya Poria. 2024. Della-merging: Reducing interference in model merging through magnitude-based sampling. *Preprint*, arXiv:2406.11617.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yiyang Du, Xiaochen Wang, Chi Chen, Jiabo Ye, Yiru Wang, Peng Li, Ming Yan, Ji Zhang, Fei Huang, Zhifang Sui, Maosong Sun, and Yang Liu. 2025. Adamms: Model merging for heterogeneous multimodal large language models with unsupervised coefficient optimization. *CoRR*, abs/2503.23733.

fujiki. 2023. japanese_alpaca_data. Hugging Face Datasets. Accessed 2026-01-01.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. 2024. Task singular vectors: Reducing task interference in model merging. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18695–18705.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's MergeKit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, Miami, Florida, US. Association for Computational Linguistics.

Jiawei Guo, Tianyu Zheng, Yizhi Li, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Graham Neubig, Wenhu Chen, and Xiang Yue. 2025. MAmmoTH-VL: Eliciting multimodal reasoning with instruction tuning at scale. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13869–13920, Vienna, Austria. Association for Computational Linguistics.

Nikolaus Hansen and Andreas Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9:159–195.

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 *International Conference on Learning Representations*.

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

Lingjie Jiang, Shaohan Huang, Xun Wu, Yixia Li, Guanhua Chen, Dongdong Zhang, and Furu Wei. 2026. Viscodex: Unified multimodal code generation via merging vision and coding models. In *The Fourteenth International Conference on Learning Representations*.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*.

Donald R Jones, Matthias Schonlau, and William J Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492.

Tamara G Kolda, Robert Michael Lewis, and Virginia Torczon. 2003. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482.

Hugo Laurencon, Leo Tronchon, Matthieu Cord, and Victor Sanh. 2024. What matters when building vision-language models? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Bo Li, Peiyuan Zhang, Kaichen Zhang, Fanyi Pu, Xinrun Du, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. 2024. Lmms-eval: Accelerating the development of large multimoal models.

Chen-An Li, Tzu-Han Lin, Yun-Nung Chen, and Hung-yi Lee. 2025a. Transferring textual preferences to vision-language understanding through model merging. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 923–943, Vienna, Austria. Association for Computational Linguistics.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2023a. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023b. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Yuan Li, Zhengzhong Liu, and Eric P. Xing. 2025b. Data mixing optimization for supervised fine-tuning of large language models. In *Forty-second International Conference on Machine Learning*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024a. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*.

Yujie Lu, Dongfu Jiang, Wenhu Chen, William Yang Wang, Yejin Choi, and Bill Yuchen Lin. 2024b. Wildvision: Evaluating vision-language models in the wild with human preferences. *Preprint*, arXiv:2406.11069.

Michael Matena and Colin Raffel. 2022. Merging models with fisher-weighted averaging. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Shota Onohara, Atsuyuki Miyai, Yuki Imajuku, Kazuki Egashira, Jeonghun Baek, Xiang Yue, Graham Neubig, and Kiyoharu Aizawa. 2025. JMMMU: A Japanese massive multi-discipline multimodal understanding benchmark for culture-aware evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 932–950, Albuquerque, New Mexico. Association for Computational Linguistics.

Michael JD Powell. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162.

Yusu Qian, Hanrong Ye, Jean-Philippe Fauconnier, Peter Grasch, Yinfei Yang, and Zhe Gan. 2025. MIA-bench: Towards better instruction following evaluation of multimodal LLMs. In *The Thirteenth International Conference on Learning Representations*.

Huaizhi Qu, Xinyu Zhao, Jie Peng, Kwonjoon Lee, Behzad Dariush, and Tianlong Chen. 2025. Uq-merge: Uncertainty guided multimodal large language model merging. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1401–1417, Vienna, Austria. Association for Computational Linguistics.

Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. 2024. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4663–4680, Miami, Florida, USA. Association for Computational Linguistics.

Chongjie Si, Kangtao Lv, Jingjing Jiang, Yadao Wang, Yongwei Wang, Xiaokang Yang, Wenbo Su, Bo Zheng, and Wei Shen. 2025. Nan: A training-free solution to coefficient estimation in model merging. *Preprint*, arXiv:2505.16148.

I.M Sobol. 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2443–2449, New York, NY, USA. Association for Computing Machinery.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. 2022. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5227–5237.

Virginia Torczon. 1997. On the convergence of pattern search algorithms. *SIAM Journal on optimization*, 7(1):1–25.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Ke-Yang Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *ArXiv*, abs/2409.12191.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Junyao Yang, Jianwei Wang, Huiping Zhuang, Cen Chen, and Ziqian Zeng. 2025. Rcp-merging: Merging long chain-of-thought models with domain-specific models by considering reasoning capability as prior. *ArXiv*, abs/2508.03140.

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 *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Ge Zhang, Xinrun Du, Bei Chen, Yiming Liang, Tongxu Luo, Tianyu Zheng, Kang Zhu, Yuyang Cheng, Chunpu Xu, Shuyue Guo, Haoran Zhang, Xingwei Qu, Junjie Wang, Ruibin Yuan, Yizhi Li, Zekun Wang, Yudong Liu, Yu-Hsuan Tsai, Fengji Zhang, and 3 others. 2024a. Cmmmu: A chinese massive multi-discipline multimodal understanding benchmark. *Preprint*, arXiv:2401.11944.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2024b. Lmms-eval: Reality check on the evaluation of large multimodal models. *Preprint*, arXiv:2407.12772.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. 2024c. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VIII*, page 169–186, Berlin, Heidelberg. Springer-Verlag.

Yuyan Zhou, Liang Song, Bingning Wang, and Weipeng Chen. 2024. MetaGPT: Merging large language models using model exclusive task arithmetic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1724, Miami, Florida, USA. Association for Computational Linguistics.

Didi Zhu, Yibing Song, Tao Shen, Ziyu Zhao, Jinluan Yang, Min Zhang, and Chao Wu. 2025. REMEDY: Recipe merging dynamics in large vision-language models. In *The Thirteenth International Conference on Learning Representations*.

Appendix

A The models used in experiments

Table 4 summarizes the models used in our experiments across different scenarios. For each scenario, we pair a VLM with a domain-expert LLM that shares the same base architecture. Specifically, for language understanding tasks, we select Chinese (Mistral-7B-v0.3-Chinese-Chat) and Japanese (Llama-3-ELYZA-JP-8B). For mathematical reasoning, we use the DART-Math series, which are LLaMA and Mistral models fine-tuned on mathematical problem-solving data. For instruction following, we pair Qwen2-VL with Qwen2-Instruct, and Idefics2-base with Mistral-7B-Instruct. All models are publicly available on Hugging Face.

B Hyperparameters Used

Table 5 details the hyperparameter configurations in Section 3 for each merging method. Classic merging methods (Task Arithmetic, DARE, TIES) require tuning merging coefficients for both the VLM and expert LLM components. We perform grid search over coefficient values in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. For DARE and TIES, which incorporate sparsification, we additionally search over density values in $\{0.2, 0.4, 0.6, 0.8\}$. This results in 25 configurations for Task Arithmetic and 100 configurations for DARE and TIES.

Data-aware methods (Fisher and RegMean) and tuning-free methods (WUDI, TSV, MetaGPT, NaN) do not require hyperparameter search, as they either derive merging weights from data statistics or compute them directly from model parameters.

C Calibration Data for Data-Aware Methods

For data-aware merging methods (Fisher and RegMean), we use 500 calibration samples randomly drawn from the corresponding dataset (i.e., for merging between base VLMs and math expert LLMs, we calibrate the VLM on a vision dataset and the math expert LLM on a textual mathematical reasoning dataset, respectively). The specific data sources for each scenario are as follows (all datasets can be downloaded from HuggingFace):

- **Vision Perception:** `liuhaotian/LLaVA-Instruct-150K`, a large-scale visual instruction tuning dataset. (Liu et al., 2023b,a, 2024b)
- **Language (Chinese):** `hfl/alpaca_zh_51k`, a Chinese instruction-following dataset translated and adapted from the original Alpaca dataset. (Cui et al., 2024)
- **Language (Japanese):** `fujiki/japanese_alpaca_data`, a Japanese version of the Alpaca dataset for instruction tuning. (fujiki, 2023)
- **Mathematical Reasoning:** `qwedsacf/competition_math`, a subset of challenging mathematical problems requiring multi-step reasoning, combined with `openai/gsm8k`. We mix the two datasets in a 50/50 ratio and shuffle the samples. (Lightman et al., 2023; Cobbe et al., 2021)
- **Instruction Following:** `databricks/databricks-dolly-15k`, a high-quality instruction-following dataset created by Databricks. (Conover et al., 2023)

D Scaling Results Across Model Sizes

Our main experiments focus on 7B–8B models. This choice is primarily motivated by the current availability of compatible VLM–LLM pairs for cross-modal skill injection: the expert LLM and the VLM backbone must share the same base architecture, and publicly available pairs satisfying this constraint are predominantly concentrated in this size range.

To examine whether our findings generalize beyond this scale, we conduct additional experiments on the Qwen2 family at 2B, 7B, and 72B scales. We evaluate NaN, our recommended tuning-free method, on instruction following, and Task Arithmetic (TA), one of our recommended classic methods, on mathematical reasoning. These results support two observations across model sizes: (1) cross-modal skill injection consistently improves over the corresponding base VLM, and (2) the gains on mathematical reasoning remain markedly smaller than in other scenarios.

Table 6 shows that for instruction following on MIA-Bench, cross-modal skill injection yields substantial gains at every scale. The merged models improve over the base VLM by 29.6 points at 2B, 40.8 points at 7B, and 36.0 points at 72B.

Table 7 shows that for mathematical reasoning on the MathVista math subset, improvements also persist across all three scales, but remain much smaller: 0.46 points at 2B, 2.53 points at 7B, and 0.92 points at 72B. Taken together, these results

Scenario	Expert LLM	VLM
Language	shenzhi-wang/Mistral-7B-v0.3-Chinese-Chat elyza/Llama-3-ELYZA-JP-8B	llava-hf/llava-v1.6-mistral-7b-hf lmms-lab/llama3-llava-next-8b
Math	hkust-nlp/dart-math-llama3-8b-prop2diff hkust-nlp/dart-math-mistral-7b-prop2diff	lmms-lab/llama3-llava-next-8b llava-hf/llava-v1.6-mistral-7b-hf
Instruction Following	Qwen/Qwen2-7B-Instruct mistralai/Mistral-7B-Instruct-v0.1	Qwen/Qwen2-VL-7B HuggingFaceM4/idefics2-8b-base

Table 4: Expert LLMs and VLMs used in our experiments across different scenarios. Each VLM is paired with a domain-expert LLM that shares the same base architecture.

Category	Algorithm	Hyperparameters
Classic merging	Task Arithmetic	`vl_weight` $\in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ `exp_weight` $\in \{0.1, 0.3, 0.5, 0.7, 0.9\}$
	DARE	`vl_weight` $\in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ `exp_weight` $\in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ `density` $\in \{0.2, 0.4, 0.6, 0.8\}$
	TIES	`vl_weight` $\in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ `exp_weight` $\in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ `density` $\in \{0.2, 0.4, 0.6, 0.8\}$
Data-aware	Fisher Merging	n/a
	RegMean	n/a
Tuning-free	WUDI	n/a
	TSV	n/a
	MetaGPT	n/a
	NaN	n/a

Table 5: Overview of merging algorithms and their hyperparameter tuning spaces. Classic merging methods require tuning merging coefficients, while data-aware and tuning-free methods operate without hyperparameter tuning.

show that the two core patterns identified in our main experiments remain stable across model sizes: merging is beneficial across scales, while mathematical reasoning is consistently harder to transfer than instruction-following ability.

Scale	Base VLM	Merged (NaN)	Gain
2B	29.3	58.9	+29.6
7B	33.0	73.8	+40.8
72B	55.5	91.5	+36.0

Table 6: Scaling results on MIA-Bench. Cross-modal skill injection with NaN improves instruction-following performance across Qwen2 model sizes.

Scale	Base VLM	Merged (TA)	Gain
2B	36.32	36.78	+0.46
7B	50.80	53.33	+2.53
72B	60.46	61.38	+0.92

Table 7: Scaling results on the MathVista math subset. Cross-modal skill injection with Task Arithmetic improves mathematical reasoning across Qwen2 model sizes, but the gains are modest.

E Full Results on MathVista

Table 8 presents the complete results on the MathVista benchmark, including the full benchmark as well as its General and Math subsets. On the Math subset, several merging methods achieve small gains over the base VLM, indicating that some mathematical reasoning ability can indeed be injected. For example, RegMean reaches the highest Math-subset score among merged models (27.13 versus 25.29 for the base VLM). However, these gains do not reliably translate to the full benchmark: most merged models underperform the base VLM overall, and even the best merged result remains only comparable.

This gap between the Math subset and the full benchmark provides a plausible explanation for why mathematical transfer remains difficult. The General subset depends more heavily on visual perception and fine-grained visual discrimination, whereas the Math subset more directly rewards symbolic and multi-step reasoning. Merging a math-specialized LLM therefore appears to strengthen reasoning-heavy behavior while simultaneously perturbing perception-related capabilities in the VLM. This interpretation is consistent with the main-text observation that mathematical reasoning can transfer, but the gains are often offset by degraded visual grounding, making mathematical reasoning more challenging.

Method	Overall	General	Math
TA	34.75	44.38	26.67
DARE	**36.12**	**48.22**	25.98
TIES	32.25	41.10	24.83
Fisher	29.88	38.90	22.30
RegMean	30.88	35.34	**27.13**
WUDI	30.38	36.99	24.83
MetaGPT	20.88	23.84	18.39
NaN	29.62	36.71	23.68
Base VLM	36.00	48.77	25.29

Table 8: Performance on MathVista across subsets. Best results among merging methods are in **bold**.

F Qualitative Examples

Here we provide several qualitative examples to help readers better understand the model's capabilities before and after merging.

Instruction following: Merging substantially improves instruction-following performance. In the MIA-Bench example shown in Table 9, the model is given an image of a party room and asked to describe the atmosphere using two adjectives and suggest a possible event type. The base VLM gives only a brief response ("social and lively") and does not suggest an event type, whereas the merged VLM produces a more detailed answer with well-chosen adjectives ("elegant" and "lively") and proposes plausible event types such as a corporate party, product launch, or networking event.

Math: As discussed in the main paper and supported by the full MathVista results in Appendix E, merging often yields modest improvements on the reasoning-heavy "Math" subset while hurting the perception-heavy "General" subset. This suggests that mathematical reasoning does transfer, although some of the gains are offset by degraded visual understanding. In our MathVista evaluation, merged models perform better on reasoning-intensive questions such as "Find the length of AC in the isosceles triangle ABC," which the base model fails to answer correctly, yet they can lose the ability to answer visually dependent questions such as "Does Aqua have the minimum area under the curve?" The examples in Table 10 illustrate this pattern.

Example 1: Qwen2-VL-72B in instruction-following

Input. Give an analysis of the atmosphere conveyed in the picture using two adjectives, and suggest what type of event it might be.

Base VLM. "The atmosphere of the image is social and lively."
Score: Component 1: 4/4; Component 2: 2/3; Component 3: 0/3; Total: 6/10. The response provides two adjectives but does not suggest an event type.

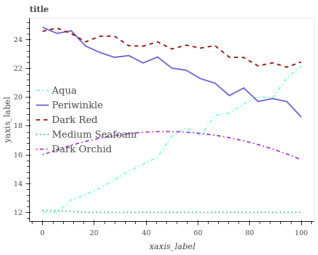
Merged VLM. "The atmosphere conveyed in the picture can be described as elegant and lively. The room is decorated with white tablecloths, candles, and a projector screen, giving it a sophisticated feel. At the same time, the colorful lighting and the presence of people mingling suggest a festive atmosphere. This type of event could be a corporate party, a product launch, or a networking event."
Score: Component 1: 4/4; Component 2: 3/3; Component 3: 3/3; Total: 10/10. The response uses exactly two adjectives ("elegant" and "lively") and suggests plausible types of events, satisfying all requirements.

Table 9: A example from MIA-Bench. The reasoning process of the judge model is summarized. The merged model follows the compositional instruction more completely than the base VLM by both selecting two suitable adjectives and inferring plausible event types from the scene.

Example 2: Qwen2-VL-72B in mathematical reasoning

Merged incorrect

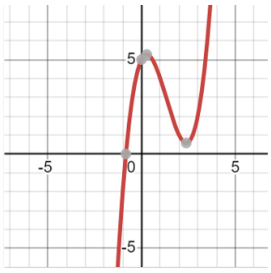
Example A: Visual comparison failure



Question. Does Aqua have the minimum area under the curve?
Gold: No
Base VLM. B. No (**Correct**)
Merged VLM. A. Yes (**Wrong**)
Interpretation: This question is perception-heavy: the model must compare multiple plotted curves, and the merged model still fails on this visually judgment.

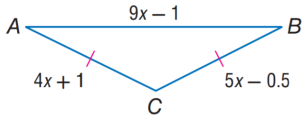
Merged correct

Example B: Simple function reading



Question. Is $f(3) > 0$?
Gold: Yes
Base VLM. B. No (**Wrong**)
Merged VLM. A. Yes (**Correct**)
Interpretation: Once the graph is localized correctly, the remaining step is a simple sign judgment; the merged model handles this reasoning step better.

Example C: Geometric reasoning gain



Question. Find the length of AC in the isosceles triangle ABC.
Gold: 7
Base VLM. C (**Wrong**)
Merged VLM. B. 7 (**Correct**)
Interpretation: The figure provides the setup, but the core difficulty is algebraic reasoning from the isosceles constraint, where merging is beneficial.

Table 10: Math examples for Qwen2 (72B). The merged model improves on reasoning-centric math questions such as function-value judgment and geometric inference, but it can still fail on perception-heavy visual comparison. This supports our claim that some mathematical reasoning transfers, while visually understanding remains a bottleneck.

G Efficiency Comparison of Merging Methods

Table 11 reports the per-configuration merge time measured on a single A800 GPU under the Chinese ability injection setup, with Llava as the base VLM and a Mistral model fine-tuned on Chinese data as the expert LLM. For classic methods, we use MergeKit (Goddard et al., 2024); for the remaining methods, we use our own implementations. Actual runtime may vary with implementation details and, for data-aware methods, with the size of the calibration dataset.

At the per-configuration level, classic methods and NaN are the most efficient, each requiring only a little over three minutes. MetaGPT also remains in the low-cost regime, although it is slightly slower due to its more complex computation. By contrast, subspace-based tuning-free methods such as WUDI and TSV incur substantially higher overhead, as they require additional subspace construction and transformation. Data-aware methods are more expensive still, because they must compute statistics from calibration data rather than derive merged weights directly from model parameters.

It is important to distinguish per-configuration cost from end-to-end cost, which also includes hyperparameter tuning. As shown in Table 2, classic methods still require substantially more total time to reach their best results because they rely on searching over multiple configurations.

Method	Merge Time (s)
Classic	
TA	196.529
DARE	195.360
TIES	194.995
Data-aware	
Fisher	9461.564
RegMean	3489.240
Tuning-free	
WUDI	1085.351
TSV	2676.234
MetaGPT	259.845
NaN	195.667

Table 11: Per-configuration merge time of different methods under the Chinese ability injection setting. Classic methods and NaN are the most efficient, with MetaGPT being slightly slower; WUDI and TSV are intermediate; data-aware methods are the slowest.

H Validation-Test Consistency Analysis

To validate the reliability of using validation sets for hyperparameter optimization, we analyze the correlation between optimization trajectories on the validation and test sets (based on the grid described in Appendix B). Figure 4 illustrates the normalized regret curves for DARE across CMMMU dataset.

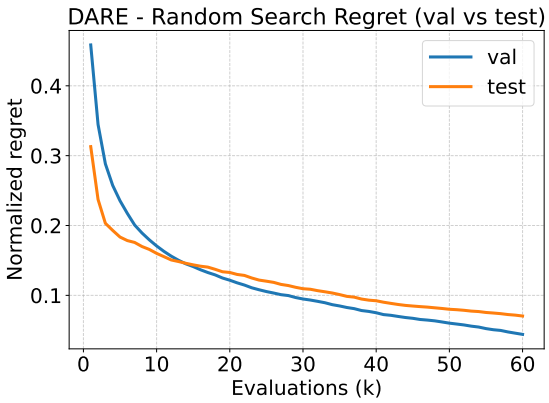


Figure 4: Normalized regret curves for validation and test sets across hyperparameter configurations. The closely aligned trajectories indicate that validation-based optimization effectively generalizes to unseen test data.

The optimization trajectories exhibit strong agreement between validation and test sets across all scenarios. As the number of evaluations increases, normalized regret decreases consistently on both sets, with the validation set serving as a faithful proxy for test performance.

This close correspondence justifies our experimental design: by conducting hyperparameter search exclusively on the 20% validation split, we avoid the prohibitive expense of repeated test-set evaluations while maintaining the reliability of our conclusions.

I Surrogate Model Fidelity Analysis

Running a comprehensive Hyperparameter Optimization (HPO) benchmark across multiple optimizers, datasets, and merging methods is computationally expensive and would incur substantial API cost if every trial were evaluated directly. We therefore approximate the hyperparameter landscape with a surrogate based on Radial Basis Function (RBF) interpolation, which provides a continuous and query-efficient proxy for the true evaluation process.

To assess fidelity, we perform Leave-One-Out

Cross-Validation (LOOCV). For each dataset–method pair, we randomly withhold one interior grid point (excluding boundary points), fit the RBF surrogate on the remaining points, and measure the prediction error on the held-out point. We repeat this process 20 times with different random seeds and report Root Mean Square Error normalized by the metric range (RMSE %).

Table 12 shows that RMSE is below 10% for most settings and remains mostly within 10–20% otherwise. This suggests that the surrogate captures the global structure of the merging landscape well. In higher-RMSE cases, it mainly smooths local noise or sharp fluctuations while preserving broad trends such as the location of the main optimum basin. This level of fidelity is sufficient for our goal of comparing optimizers by their ability to find strong regions of the landscape rather than fit every local irregularity.

Dataset	DARE	TA	TIES
CMMMU	9.27	7.77	15.99
JMMMU	6.96	18.79	8.72
MV-L3	17.42	14.75	9.24
MV-M	9.55	17.53	14.06
MVis-L	10.25	20.35	11.36
MVis-M	11.15	16.69	11.91
MIA-I2	4.58	5.63	7.65
MIA-Q2	4.60	11.37	4.45
WV-I2	7.62	11.47	12.22
WV-Q2	8.26	13.16	4.58

Table 12: Interpolation RMSE (%) for the surrogate model, averaged over 20 runs using a leave-one-out protocol. DARE refers to DARE-Linear. Lower values indicate higher fidelity. Even higher values ($\sim$15%) are acceptable as they reflect a smoothing of local noise while preserving global landscape structure. Abbreviations—MV: MathVerse, MVis: MathVista, MIA: MIA-Bench, WV: WildVision; L3: LLaMA3, M: Mistral, L: LLaMA, I2: Idefics2, Q2: Qwen2.