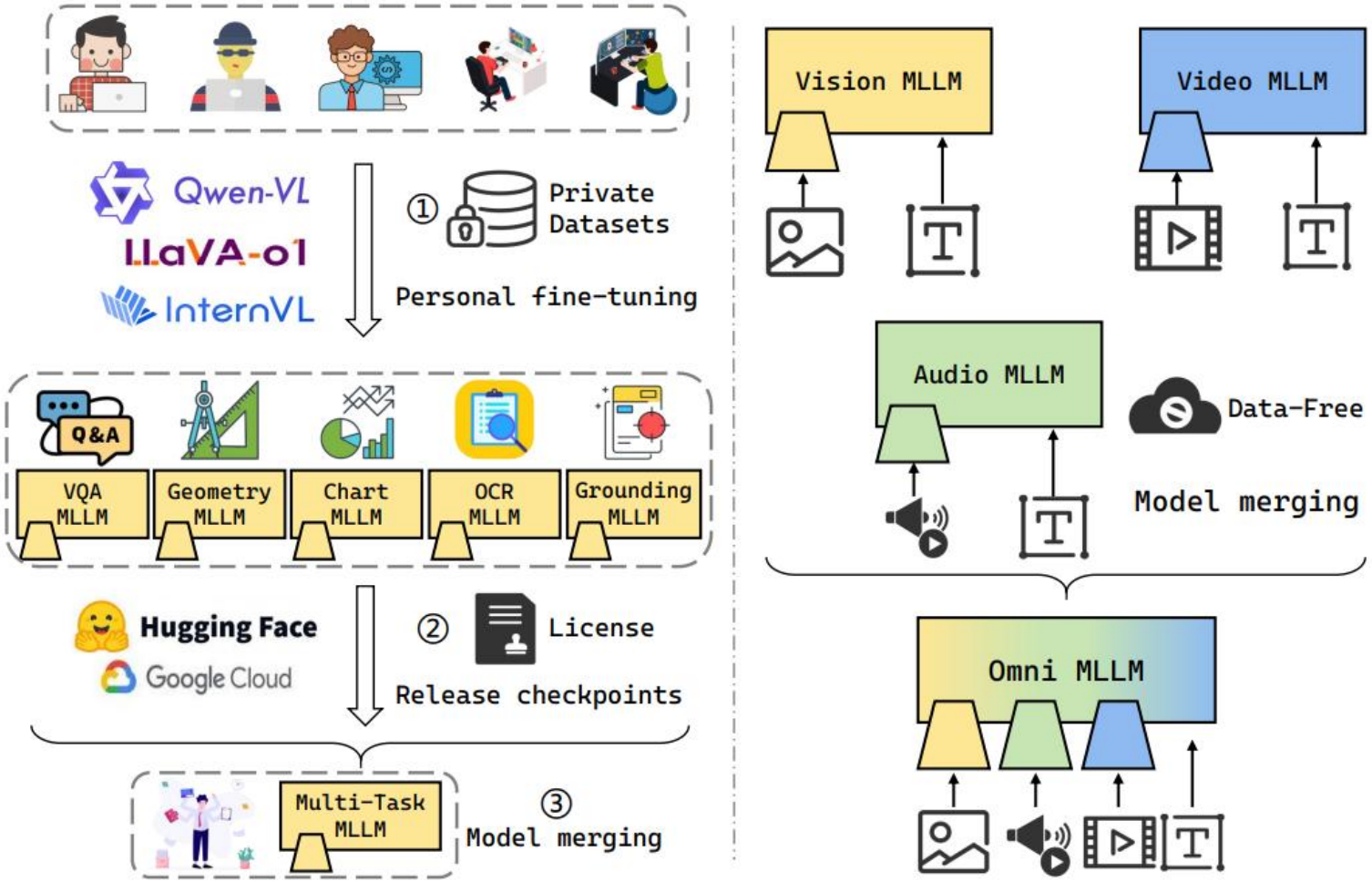
Model merging technology of multimodal large language model

Yongxian Wei

- **Category of Idea**
- **Basic algorithms with broad applicability and excellent performance/representing technological trends**
- exposition of evidence
- Excellent performance: the effect is better than the widely used model merging method in the industry
- Wide applicability: The proposed benchmark is suitable for extensive experiments on merging methods.

Background

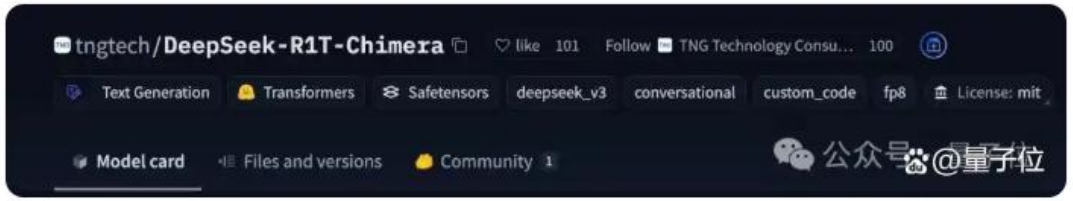
By consolidating capabilities or modalities from open-source communities through model merging, this approach proves cost-effective.



Background

Due to resource-intensive training requirements, the development of foundational models progresses slowly, while domain-specific models continue to evolve during the transition period. Various developers have released their fine-tuned models in open-source communities such as Hugging Face.

现在，等不及DeepSeek官方，**开源社区已经开始自己动手给V3-0324加入深度思考了。**



新模型**DeepSeek-R1T-Chimera**，能力与原版R1相当，但速度更快，输出token减少40%，也是基于MIT协议开放权重。

相当于拥有接近R1的能力和接近V3-0324的速度，结合了两者的优点。

北大联合360发布!5%参数量逼近DeepSeek-R1满血数学性能

2025年2月26日 研发团队表示:"Tiny-R1-32B-Preview的成功是站在了巨人的肩膀上,受益于开源社区精神,结合DeepSeek-R1蒸馏、DeepSeek-R1-Distill-32B增量训练、模型融合等技术,使用360-LLaMA...

北京大学新闻网

Model merging aims to integrate multiple expert models into a unified model with diverse capabilities. This approach reduces storage and service costs through model reuse, while supporting distributed development by enabling independent contributors to build models that can be merged later.

Existing technology and existing problems

Despite its immense potential, prior research has primarily focused on merging visual models or large language models specifically designed for code and mathematical tasks.

- Based on visual classification model[1]

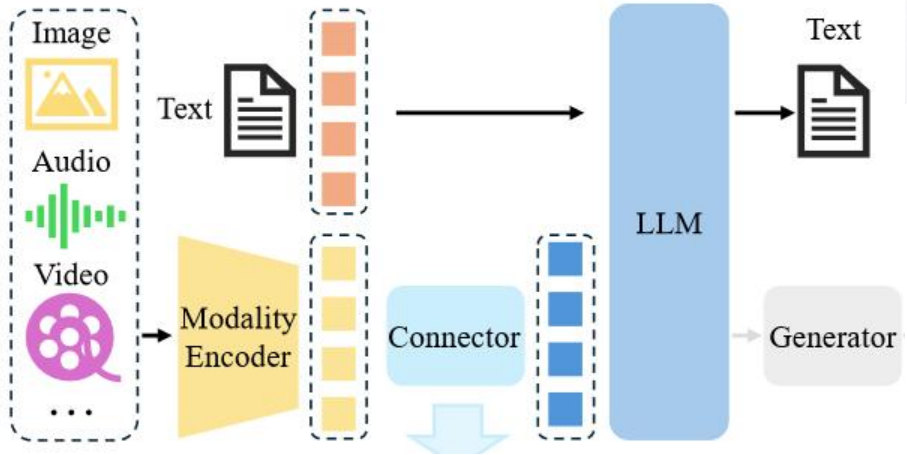
  Modeling multi-task representation extraction through merging multiple datasets

- Based on large language model[2]

  Focus on code, math, and follow instructions tasks

- Based on multimodal LLMs[3]

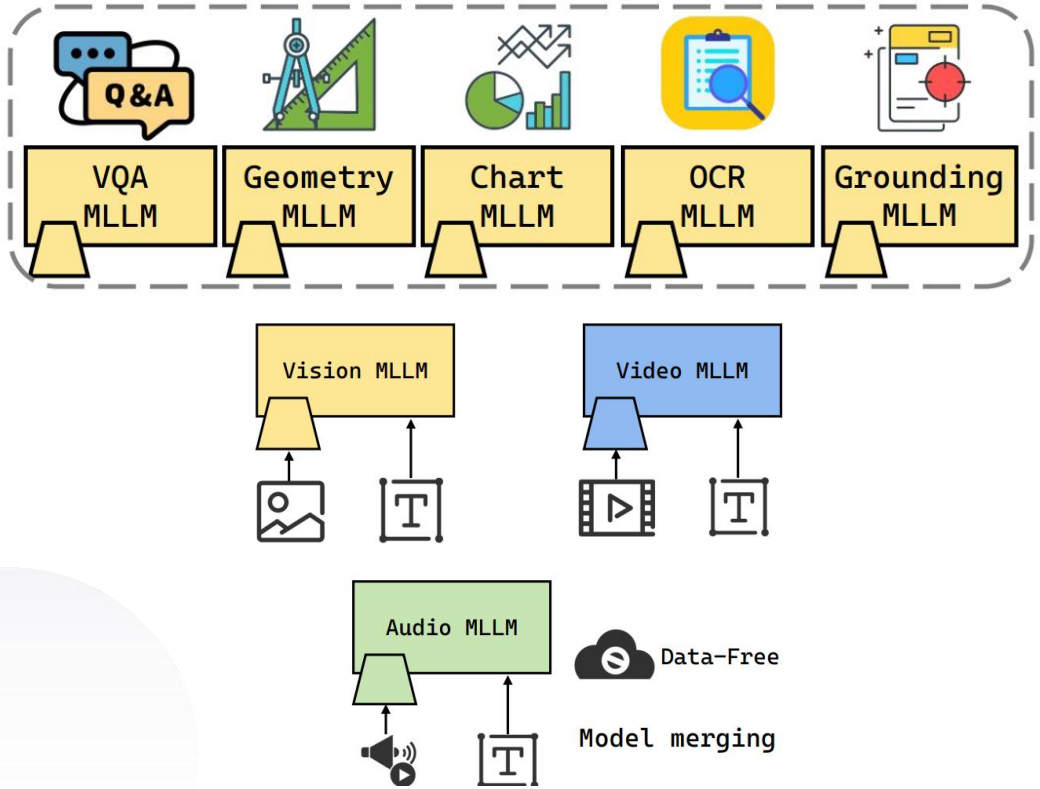
  The absence of a benchmark for model merging studies, coupled with the lack of clearly defined tasks for MLLM training and evaluation.



[1] Editing models with task arithmetic. In ICLR, 2023.
[2] Evolutionary optimization of model merging recipes. Nature Machine Intelligence, 2025.
[3] AdaMMS: Model merging for heterogeneous multimodal large language models with unsupervised coefficient optimization. In CVPR, 2025.

The contribution

- Benchmark: We present a model merging benchmark, provide a detailed classification of MLLM functionalities, and explore how model merging effectively combines different MLLM patterns. We hope this benchmark will assist the model merging community in better evaluating the generalizability of their methods.



- Methods: We further propose a novel merging method that effectively removes noise from task vectors and enhances the robustness of merged vector optimization. Ablation studies demonstrate an average performance gain of 2.48%.

Table 5: **The ablation study.**

	Qwen2-VL	Vicuna-7B
WUDI Merging	58.65	64.65
+ SGD	48.88(-9.77%)	66.91(+2.26%)
+ Initialization	63.08(+4.43%)	**67.07**(+2.42%)
+ Low-rank	**63.30**(+4.65%)	67.00(+2.35%)

Core question: What challenges arise when merging models in multimodal large-scale models, and how can they be addressed?

Core Innovation Point 1: A New Benchmark for Multimodal Large Model Fusion

- Key steps: ① The MLLM integration benchmarks encompass diverse tasks including Visual Question Answering (VQA), geometry, graphics, OCR recognition, and visual localization. For each task, we collected public datasets with at least 100,000 samples to ensure effective supervised fine-tuning, and selected corresponding benchmarks to evaluate different capabilities. ② We explored how model integration can effectively combine different modalities (e.g., visual language, audio language, and video language models) to advance toward an Omni language model. This provides a data-free approach that allows reusing specific modalities' encoders and integrating them into a unified LLM.

Table 1: **Summary of task-specific datasets**, with their corresponding sizes and languages.

Task Category	Size	Datasets (Language)
VQA	588K	GQA (en) [32], VQAv2 (en) [27], OKVQA (en) [54], LLaVA-Instruct (zh) [45], CogVLM-Singleround (en&zh) [75], CogVLM-Multiround (en&zh) [75]
Geometry	190K	GeoQA+ (zh) [5], G-LLaVA (en) [21]
Chart	218K	ChartQA (en) [55], DVQA (en) [35]
OCR	238K	OCRVQA (en) [59], TextCaps (en) [64], SynthDoG (en) [38], LLaVAR (en) [91], ST-VQA (en) [4], TextVQA (en) [65], DocVQA (en) [56], DeepForm (en) [69], KLC (en) [66], TabFact (en) [9]
Grounding	135K	RefCOCO (en) [52, 87], VG (en) [39]

Table 8: Overview of modality components and training data.

Modality	Modality Encoder	Connector	Alignment Data	Fine-tuning Data	Referenced Work
Vision	CLIP-ViT-L-336px [63]	MLP	LCS 558K [46]	LLaVA-mixed 665K [45]	LLaVA-1.5 [45]
Audio	BEATs-Iter3+ [8]	Q-Former	WaveCaps 400K [57]	OpenAQA filtered 350K [25]	X-InstructBLIP [61]
Video	LanguageBind [93]	MLP	LCS 558K [46], Valley 702K [50]	Video-ChatGPT 100K [51], LLaVA-mixed subset 140K [45]	Video-LLaVA [44]

Core Innovation Point 2: Theoretical Derivation of Model Merging

- Key steps: ① We derived the upper bound of error between the merging model and expert model, demonstrating that merging performance is influenced by the learning rate and number of iterations of the control parameter drift. Smaller parameter changes make merging easier. Scheme advantage: When constructing benchmarks, we ensure performance improvement for specific tasks while minimizing parameter variation.

Theorem 3.1. *Let $\boldsymbol{\theta}_0 \in \mathbb{R}^d$ be the initial parameter. For each task i, after T gradient steps with constant learning rate η, we denote the task vector as $\boldsymbol{\tau}_i = \boldsymbol{\theta}_i - \boldsymbol{\theta}_0$. Considering that most methods can be viewed as extensions of linear combinations of task vectors, let $\boldsymbol{\tau}_m = \sum_j \alpha_j \boldsymbol{\tau}_j$ denote the merged vector. The loss on task i is denoted by $\mathcal{L}_i(\Theta)$, which is $\mathcal{C}_i$-Lipschitz continuous. Assuming a constraint on the second moment of the gradient, then:*

$$\|\mathcal{L}_i(\boldsymbol{\theta}_0 + \boldsymbol{\tau}_m) - \mathcal{L}_i(\boldsymbol{\theta}_0 + \boldsymbol{\tau}_i)\| \leq \mathcal{O}(\eta T) \tag{2}$$

This indicates that both the learning rate and iterations influence model merging results. Please refer to App. A for detailed assumptions and proofs.

Core Innovation Point 3: Improved Merge Algorithm Based on Task Vector Optimization

- Key steps: ① The task vector τ approximates a linear subspace of fine-tuning data x. This property enables implicit utilization of training data information solely through the task vector. The interference between the merged vector and the task vector at layer l is defined as: the interference of task i at layer l is $\tau_{m,l} - \tau_{i,l}$. To optimize the merged vector $\tau_{m,l}$, we minimize the interference term $(\tau_{m,l} - \tau_{i,l}) \times_i$, where x_i is the task input. Leveraging the linear subspace relationship, we substitute $x_{i,l}$ with the transpose of $\tau_{i,l}$. ② However, the task vector contains extreme redundancy and noise, which generates mutual interference and hinders effective optimization of the merged vector. Moreover, the merged vector tends to

$$\min_{\boldsymbol{\tau}_{m,l}} \mathcal{L}_l = \sum_{i=1}^{n} \frac{1}{\|\boldsymbol{\tau}_{i,l}\|_F^2} \left\| (\boldsymbol{\tau}_{m,l} - \boldsymbol{\tau}_{i,l})(\boldsymbol{\tau}_{i,l})^\top \right\|_F^2 .$$

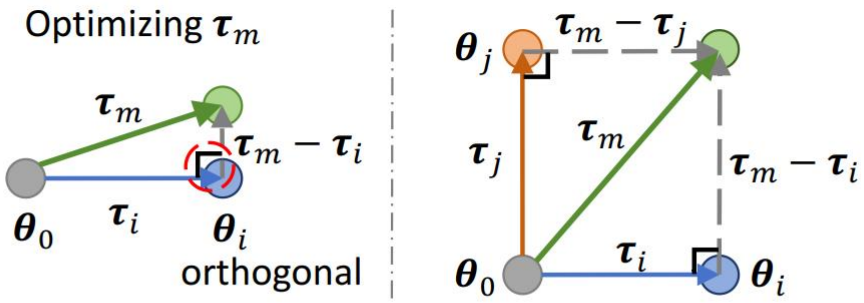


Figure 3: $\boldsymbol{\tau}_m$ tends to take shortcuts by increasing its magnitude to encourage orthogonality.

Core Innovation Point 3: Improved Merge Algorithm Based on Task Vector Optimization (Continued)

- Key Steps: ① We propose a novel merging method that improves task vector optimization (i.e., parameter variations between fine-tuning and base models). This approach optimizes the merged model based on loss functions defined through task vector interactions, while applying low-rank approximation to reduce redundant noise for optimal results. ② Subsequently, we employ low-rank approximation to eliminate redundant noise, where U, Σ, and V represent the first (k) singular components. Additionally, we found that substituting Σ and V with task vectors (τ) as input subspace X allows us to discard secondary row space information and focus solely on column feature space. By truncating singular values, we retain key features V,

$$\mathrm{SVD}(\boldsymbol{\tau}_{i,l} - \bar{\boldsymbol{\tau}}_l) = U\Sigma V^\top, \text{ where } U \in \mathbb{R}^{m\times r}, \Sigma \in \mathbb{R}^{r\times r}, V \in \mathbb{R}^{n\times r}.$$

$$\min_{\boldsymbol{\tau}_{m,l}} \mathcal{L}_l = \sum_{i=1}^{n} \frac{1}{\|\boldsymbol{\tau}_{i,l}\|_F^2} \left\| (\boldsymbol{\tau}_{m,l} - U_{1:k}\Sigma_{1:k}V_{1:k}^\top - \bar{\boldsymbol{\tau}}_l)(\Sigma_{1:k}V_{1:k}^\top)^\top \right\|_F^2.$$

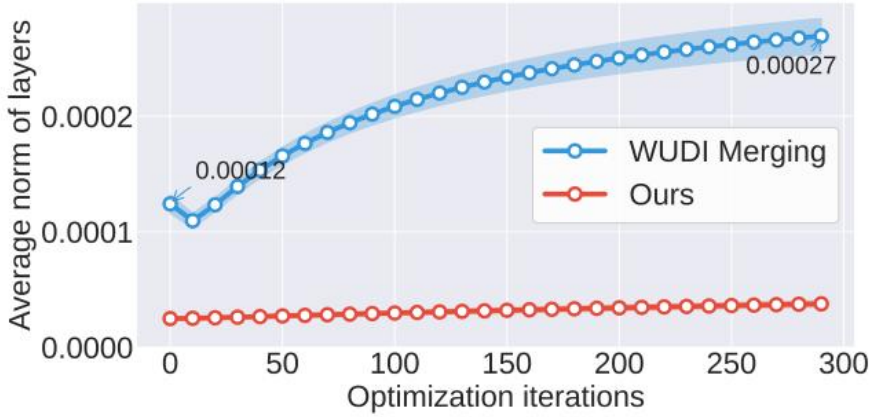


Figure 4: Frobenius norm of the merged vector changes during optimization (average by layers).

Beneficial effects: The model merging combines the strengths of expert models

Task vectors effectively capture the specialized capabilities of individual models, enabling successful model merging. By combining these advantages, the merged models outperform expert-level multimodal large language models (MLLMs)in their respective tasks. For instance, the merged Qwen2-VL significantly outperforms single models in geometric tasks (51.05 vs. 40.79 vs. 42.50 vs. 28.95) and shows marked improvement in chart tasks (79.76 vs. 61.08). Similar enhancements are observed in OCR and object localization tasks. For InternVL2.5-Instruct, we achieved SFT by combining all task-specific training data for hybrid training.

Table 2: **Capability merging results on InternVL2.5 (full fine-tuning) across multiple tasks**. For the merging methods, we highlight the best score in bold and the second-best score with underlining.

Methods	VQA		Geometry		Chart	OCR		Grounding			Avg.
	VizWiz	GQA (test)	MathVista (mini)	MATH-Vision (mini)	ChartQA (test)	TextVQA (val)	OCRVQA (test)	RefCOCO	RefCOCO+	RefCOCOg	
InternVL2.5-Instruct	29.15	54.62	46.80	18.42	69.48	72.51	41.08	71.69	65.41	67.40	53.66
Individual VQA	30.58	60.91	35.50	17.11	48.76	63.68	36.04	-	-	-	41.80
Individual Geometry	13.45	32.80	55.20	25.00	51.76	56.91	35.35	24.73	19.61	23.84	33.86
Individual Chart	20.16	40.39	23.84	10.53	69.52	54.36	34.83	-	-	-	36.23
Individual OCR	12.40	22.22	23.31	10.53	36.88	73.00	54.79	73.65	68.01	69.10	44.39
Individual Grounding	19.09	25.88	28.91	14.47	41.32	58.39	74.87	76.67	71.35	70.09	48.10
Weight Average [73]	29.96	54.89	49.60	18.42	71.64	74.54	41.86	52.62	45.29	52.39	49.12
Task Arithmetic [34]	30.67	56.34	45.36	21.05	72.88	76.26	43.39	74.90	68.15	72.75	56.18
TIES Merging [75]	30.63	56.48	44.50	23.68	72.28	76.29	44.01	76.01	68.45	73.65	56.70
TA w/ DARE [84]	30.61	56.48	48.45	21.05	**73.08**	**76.30**	43.03	74.94	68.07	73.02	56.50
TIES w/ DARE [84]	30.65	56.11	43.85	27.63	72.72	76.19	43.33	75.10	68.48	73.55	56.76
TSV Merging [24]	**31.15**	56.67	52.45	**28.95**	70.56	75.66	45.38	65.19	58.51	59.17	54.37
Iso-C [51]	28.21	55.36	48.96	21.05	70.56	69.34	**46.51**	72.72	66.56	68.50	54.78
WUDI Merging [14]	31.02	56.96	53.03	17.11	69.19	75.95	46.12	**76.06**	**70.14**	**74.48**	57.00
WUDI v2 (Ours)	30.97	**57.13**	**54.48**	21.05	68.72	76.01	46.35	75.97	69.72	73.94	**57.44**
Mixture Training	29.79	61.33	52.83	23.68	70.32	72.96	60.25	72.06	65.93	67.46	57.66

Beneficial effects: The model merging combines the strengths of expert models

For Qwen2-VL-Base, we directly adopted Qwen2-VL-Instruct as the upper limit for hybrid training, given its extensive SFT across diverse datasets. Notably, our optimal model merging approach achieved performance that rivals or even surpasses both hybrid training and the Instruct version. These results demonstrate that model merging not only holds the potential to outperform multi-task learning but also provides a scalable solution for creating high-performance MLLMs with reduced computational costs and time.

Table 3: **Capability merging results on Qwen2-VL (LoRA fine-tuning) across multiple tasks.** For the merging methods, we highlight the best score in bold and the second-best score with underlining.

Methods	VQA		Geometry		Chart	OCR		Grounding			Avg.
	VizWiz	GQA (test)	MathVista (mini)	MATH-Vision (mini)	ChartQA (test)	TextVQA (val)	OCRVQA (test)	RefCOCO	RefCOCO+	RefCOCOg	
Qwen2-VL-Base	5.52	5.39	47.85	23.68	0.36	20.22	1.07	45.32	37.55	31.26	21.82
Individual VQA	41.38	62.60	33.71	28.94	66.56	80.21	55.33	39.31	32.71	38.01	47.88
Individual Geometry	35.57	44.63	42.50	28.95	14.56	73.95	45.96	5.57	2.31	3.90	29.79
Individual Chart	38.58	24.24	49.28	32.89	61.08	79.75	63.67	46.28	36.67	34.06	46.65
Individual OCR	28.38	37.53	31.81	13.16	57.40	70.50	64.68	0.59	0.46	0.26	30.48
Individual Grounding	38.60	32.92	36.17	19.74	18.08	75.05	48.27	72.14	65.33	66.48	47.28
Weight Average [73]	41.47	57.33	<u>50.21</u>	34.21	59.56	81.09	57.85	80.72	65.37	77.68	60.55
Task Arithmetic [34]	40.52	<u>62.31</u>	40.36	26.31	<u>79.67</u>	81.09	59.50	75.96	61.33	75.85	60.29
TIES Merging [75]	41.38	59.08	46.87	34.21	67.24	81.42	58.53	80.63	65.36	77.65	61.24
TA w/ DARE [84]	40.64	**62.38**	40.67	26.31	**79.76**	81.04	59.34	75.83	61.41	75.80	60.32
TIES w/ DARE [84]	**41.63**	59.96	45.72	<u>35.53</u>	70.68	<u>81.53</u>	59.63	<u>80.73</u>	<u>65.65</u>	<u>77.77</u>	<u>61.88</u>
TSV Merging [24]	41.43	57.31	**51.05**	34.21	59.44	81.25	57.81	80.71	65.34	77.76	60.63
Iso-C [51]	12.31	13.44	39.96	27.63	2.80	30.05	6.12	53.68	38.96	41.90	26.69
WUDI Merging [14]	37.19	56.45	42.96	27.63	67.84	79.92	**65.56**	76.25	60.72	71.99	58.65
WUDI v2 (Ours)	<u>41.61</u>	61.16	48.66	**40.79**	74.08	**81.54**	<u>60.06</u>	**80.92**	**65.90**	**78.24**	**63.30**
Qwen2-VL-Instruct	44.09	62.18	46.02	19.73	70.04	78.38	65.42	82.89	77.87	75.63	62.23

Beneficial effects: Model merging across modalities

The merging approach effectively integrates information from three modalities, outperforming models trained on individual visual, audio, or video inputs. This highlights the complementary nature of modalities and their potential for integration. Online Composing dynamically merges activations from different modalities in LLMs during inference, requiring separate parameter storage for each modality (i.e., 3× static merging). NaiveMC performs simple activation averaging, while DAMC decouples parameters during training to reduce modal interference. Notably, the best merging method even outperforms these online combination approaches.

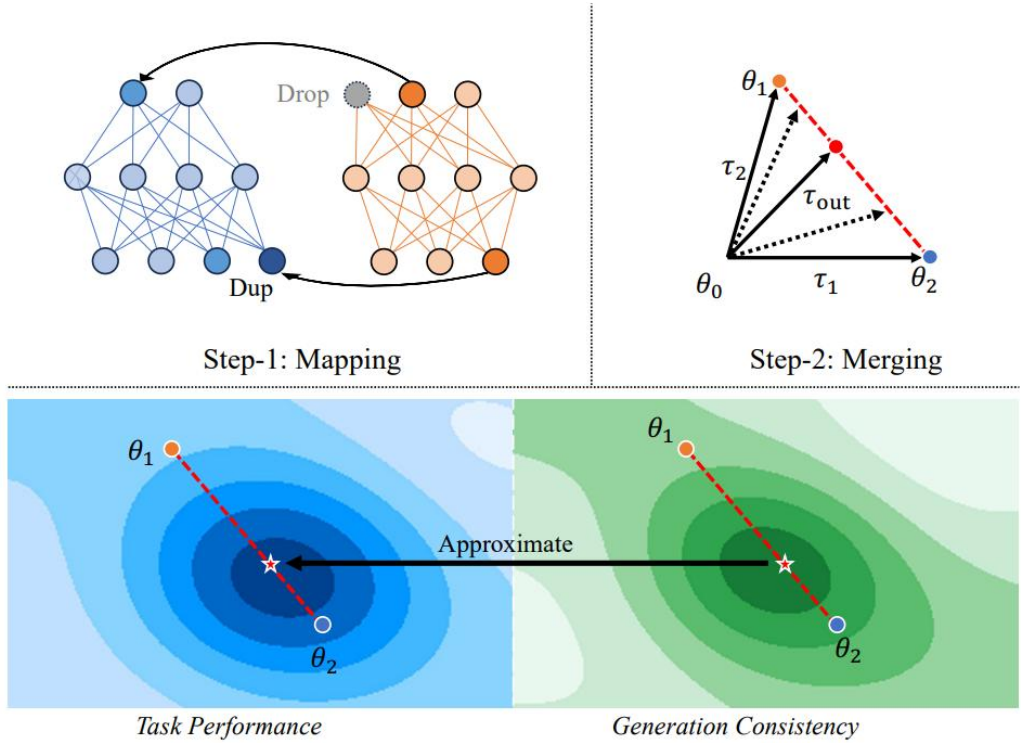
Table 4: **Modality merging results on zero-shot image-audio-video question answering tasks** by merging vision-language, audio-language, and video-language models. The "Individual Modalities" columns show baseline performance for each single-modality model.

	Individual Modalities			**Merging Methods**							**Online Composing**	
Datasets	Vision	Audio	Video	Weight Average [77]	Task Arithmetic [33]	Ties Merging [79]	TSV Merging [22]	Iso-C [53]	WUDI Merging [13]	WUDI v2 (Ours)	NaiveMC [6]	DAMC [6]
MUSIC-AVQA	50.77	27.93	49.02	47.75	52.14	50.35	**53.78**	52.77	52.43	<u>53.17</u>	53.50	52.80
AVQA	75.55	47.57	79.20	69.39	78.62	75.84	**80.90**	77.51	76.86	<u>80.82</u>	80.26	80.78
Avg.	63.16	37.75	64.11	58.57	65.38	63.10	**67.34**	65.14	64.65	<u>67.00</u>	66.88	66.79

Key Point 1: Establishing a Benchmark for Multi-modal Large Model Fusion ① Clearly categorize multi-modal tasks and construct models using publicly available datasets ② Integrate multiple bimodal models into a unified multimodal model

Key Point 2: Optimized Fusion Algorithm Based on Task Vector Refinement ① Remove task vector noise through low-rank estimation to enhance input data accuracy ② During LoRA fine-tuning, employ more robust optimization for fusion vector generation

document analysis

AdaMMS: Model Merging for Heterogeneous Multimodal Large Language Models with Unsupervised Coefficient Optimization, in CVPR 2025	
Existing literature solutions	**Differences from the present paper**
for implementing the multimodal large model integration scheme  	AdaMMS proposes an unsupervised hyperparameter selection method for model merging. However, its time-consuming process requires generating responses for each candidate hyperparameter and assumes the availability of test sets during merging. Additionally, it can only merge two models at a time. For example, merging LLaVA OneVision Qwen into Qwen2 VL on the Qwen2 architecture, or integrating LLaVA-v1.5 into CogVLM-Chat on the LLaMA architecture. Our benchmark tests utilize more comprehensive data with clearer MLLM task divisions for fine-tuning, and we propose a data-free approach that eliminates the need for hyperparameter search.

Thank You