

This Appendix first provides a discussion of the border impact of the paper in appendix A. The details of experimental settings and models are in appendix B. Further experiments are illustrated in appendix C. More analysis of selected parameters is given in appendix D. The figure of our method is in appendix E.

A Border Impact

This work presents IP-Merging, a novel tuning-free method that effectively transfers mathematical reasoning capabilities from specialized math LLMs to multi-modal LLMs (MLLMs). Our approach does not pose any potential societal impacts or ethical concerns. We rely solely on publicly available models and datasets, none of which require specific acknowledgment.

B Detailed Experimental Settings

B.1 Datasets

We test our models math reasoning benchmarks MathVista [21] and MathVerse [48], one general QA benchmark MMMU [45]:

- **MathVista** assesses the MLLMs’ multimodal mathematical skills, the testing data can be divided into five subsets: Figure Question Answering (FQA), Geometry Problem Solving (GPS), Math Word Problems (MWP), Textbook Question Answering (TQA), and Visual Question Answering (VQA). For evaluation, following [21, 28], we first employ GPT-4 to extract the predicted choices or answers from responses, then report the answer accuracy, which determines whether the final answer matches the ground truth.
- **MathVerse** includes a diverse set of math problems that require understanding and reasoning over both textual and visual information, such as charts, diagrams, and equations. The testing data can be divided into five subsets, i.e., Text Dominant, Text Lite, Vision Intensive, Vision Dominant and Vision Only.
- **MMMU** includes 900 evaluation samples and covers six core disciplines: Art & Design, Business, Science, Health & Medicine, Humanities & Social Science, and Technology & Engineering, making it suitable for assessing the general knowledge of MLLLM.

B.2 Models and Comparison Methods

We employ the LLaVA series [19] and Qwen 2 series [32] as our base model. We use other fine-tuned Math LLMs such as Tora series models, MetaMath models [8, 42] and Qwen2-Math models [37]. The pretrained foundation LLM of LLaVA models, Tora models and Metamath is the LLaMA-2 [31] model. The pretrained foundation of Qwen series models is the Qwen-2 model [32]. We compare our proposed methods with prevailing model merging techniques:

- **Base Model** The performance of base MLLM on three tasks, we reproduce the results with the officially released code.
- **Task Arithmetic** [14] combines all the task vectors extracted from the models into one multi-task model.
- **Ties Merging** [36] addresses task inference by pruning redundant parameters. The process involves three steps: Trim, Elect Sign, and Disjoint Merge.
- **EMR Merging** [13] computes one unified task vector and computes task-specific masks based on a unified task vector. The final task vector is computed by combining all the masked task vectors and weighting all the task vectors by rescale parameters.

Table 5: List of MLLMs and LLMs.

Models	Type	Pretrained Base Model	Source LLM
LLaVA-V1.5-7B	MLLM	Vincuna-v1.5-7B	Llama-2-7B
Table-LLaVA-V1.5-7B	MLLM	Vincuna-v1.5-7B	Llama-2-7B
LLaVA-Next-7B	MLLM	Vincuna-v1.5-7B	Llama-2-7B
LLaVA-V1.6-7B-Llama3-8B	MLLM	Llama-3-8B	Llama-3-8B
LLaVA-V1.5-13B	MLLM	Vincuna-v1.5-13B	Llama-2-13B
Qwen2-VL-7B	MLLM	Qwen2-7B	Qwen2-7B
Qwen2-Math-7B-base	LLM	Qwen2-7B	Qwen2-7B
Tora-7b	LLM	Llama-2-7B	Llama-2-7B
Tora-Code-7B	LLM	CodeLLaMA-7B	Llama-2-7B
Tora-Code-13B	LLM	CodeLLaMA-13B	Llama-2-7B
WizardMath-7B-V1.0	LLM	Llama-2-7B	Llama-2-7B
Tora-7b	LLM	Llama-2-7B	Llama-2-7B
MetaMath-7B	LLM	Llama-2-7B	Llama-2-7B
OpenO1-Llama3-8B	LLM	Llama-3-8B	Llama-3-8B
DeepSeek-R1-distilled-Llama-3-8B	LLM	Llama-3-8B	Llama-3-8B

927 We list the models used in the experiments in table 5.

928 B.3 Details of Hyperparameter Selection

929 Following previous works in model merging [36, 13], we adopt the grid search for hyperparameters
 930 for the baseline methods, specifically, we set the hyperparameters based on the following range:

- 931 • **Task Arithmetic** [14] involves the scaling coefficients for merged task vectors, which are
 932 set ranging from [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9].
- 933 • **Ties Merging** [36] involves the scaling coefficient and ratio to retain large parameters, the
 934 scaling coefficients are set ranging from [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9], ratio to
 935 retain parameters with largest-magnitude values: [0.1, 0.2, 0.3].
- 936 • **EMR Merging** [13] does not involve specific hyperparameters.
- 937 • **IP Merging** involves the similarity threshold to determine whether the layer should be
 938 selected for merging.

C Further Experiments

C.1 Merging Math MLLM with Math LLM

We conduct the experiments of merging fine-tuned math MLLM (TableLLaVA-1.5-7B [51], G-LLaVA-7B [7]) and Tora model using our proposed merging method, validating the effectiveness of improving fine-tuned math MLLM further. G-LLaVA is obtained by further fine-tuning the LLaVA using geometry reasoning data. After merging the math LLM, the geometry reasoning ability is further enhanced, which is validated by the 2.8% and 1.5% improvement on GPS and FQA tasks. TableLLaVA is obtained by further fine-tuning the LLaVA using table reasoning data. By merging math reasoning LLM, our method can further improve the model’s performance on table reasoning tasks such as FQA by 3.7%. All of these demonstrate that our method can further improve the math reasoning abilities of both base MLLM and fine-tuned math MLLM.

Table 6: We merge Math MLLM with the Tora-code-7B model. **Bold** represents the best performance. “Avg” is the average performance.

Approach	MathVista						
	Params	TQA	GPS	VQA	FQA	MWP	Avg
<i>G-LLaVA-7B as Base Model</i>							
Base Model	7B	29.1	48.6	33.5	19.3	11.3	28.0
+IP Merging	7B	32.9	51.4	31.8	20.8	12.9	29.6 +1.6 ↑
<i>TableLLaVA-1.5-7B as Base Model</i>							
Base Model	7B	34.2	27.4	29.6	24.9	41.9	30.9
+IP Merging	7B	41.8	27.9	30.2	28.6	43.6	34.0 +3.1 ↑

C.2 Merge Larger Models

We conduct experiments on merging larger models in table 7. The experiments are conducted using models with 13B parameters. When dealing with larger models, our approach can also boost the performance compared to other model merging methods by 2.8% on average. In subtasks such as VQA, where math-related knowledge is highly demanded, our method achieves a 3.3% performance gain.

Table 7: Experiments of merging 13B models. We merge LLaVA-13B with Tora-code-13B using different merging methods. **Bold** represents the best performance. “Avg” is the average performance.

Approach	MathVista						
	Param	TQA	GPS	VQA	FQA	MWP	Avg
Base Model	13B	41.1	25.0	34.1	21.9	16.1	26.7
Task Arithmetic	13B	39.9	34.6	29.1	22.3	7.0	26.0
Ties Merging	13B	38.6	29.3	34.1	22.3	17.2	25.9
IP-Merging	13B	42.4	26.0	37.4	23.1	18.8	28.5 +1.8 ↑

C.3 Merge Multiple Models

We conduct experiments on emerging multiple models to verify the scalability of our method in table 8. We use LLaVA-Next-7B as the foundation MLLM, then merge multiple LLMs, such as the Tora Model and MetaMath-based models. As is shown in the table 8, by merging more math reasoning models, the performance of math reasoning MLLM can be further improved.

Table 8: Experiments of merging multiple models on MathVista.

Models	MathVista								
	Params	Approach	Merged LLM	TQA	GPS	VQA	FQA	MWP	Avg
MLLM	7B	Base Model	None	44.9	26.9	33.5	32.3	17.8	30.7
MLLM+ 1 LLM	7B	IP-Merging	Tora-code-7B	46.2	32.2	30.2	33.8	18.3	31.9 +1.2 ↑
MLLM+ 2 LLMs	7B	IP-Merging	Tora-code-7B MetaMath-7B	47.5	26.0	32.4	35.7	23.7	32.7 +2.0 ↑

C.4 Results of Hyperparameter Experiments

We conduct ablation experiments for the hyperparameters in table 9. We show the results of different scaling coefficients in the task vector, the proportion of the retained parameters and the scaling coefficients in ties merging, and similarity thresholds (i.e., S_α in eq. (6)) in our method. We can see that for the Llava models, the threshold around 0.3 and 0.4 provides the best performance, balancing the number of layers associated with math reasoning to be merged and the importance of the layers to math reasoning. For the Qwen model, 0.6 is the optimal choice. We can also see that with higher thresholds, the performance fluctuates within a small range. We also show the comparative methods of ties merging and task vector, the performance is sensitive to the different scaling coefficients or the rate of the retained parameters.

Table 9: Results of different hyperparameters. **Bold** represents the best performance.

Methods	LLaVA-1.5-7B			Qwen-2-VL		
	MathVista	MathVerse	MMMU	MathVista	MathVerse	MMMU
Base Model	25.2	11.3	34.2	55.4	24.8	50.7
Task Vector						
scale=0.1	21.0	6.5	24.4	25.8	6.0	29.3
scale=0.2	25.2	7.9	25.6	27.8	5.9	32.6
scale=0.3	22.2	8.5	26.6	30.8	8.8	29.1
scale=0.4	24.5	3.6	26.3	31.1	9.3	31.0
scale=0.5	23.1	4.4	30.2	28.9	7.6	26.0
scale=0.6	24.8	5.4	25.8	28.1	4.9	26.9
scale=0.7	23.3	6.2	26.2	28.5	1.4	24.9
scale=0.8	23.9	8.0	24.4	29.2	0.0	25.2
scale=0.9	20.9	9.8	25.6	24.1	0.0	23.9
Ties Merging						
retain=0.3,scale=0.1	22.9	5.9	25.8	24.9	1.5	27.6
retain=0.3,scale=0.2	22.9	4.8	24.2	26.5	1.2	27.0
retain=0.3,scale=0.3	24.8	7.3	25.8	26.2	2.3	27.2
retain=0.3,scale=0.4	23.1	6.9	25.6	27.9	0.6	26.4
retain=0.3,scale=0.5	26.1	1.9	22.6	27.6	0.6	26.2
retain=0.3,scale=0.6	25.7	1.8	26.2	26.7	0.4	27.1
retain=0.3,scale=0.7	26.6	0.0	25.7	26.8	0.1	25.7
retain=0.3,scale=0.8	26.6	0.0	22.6	27.0	0.0	24.8
retain=0.3,scale=0.9	25.0	0.0	25.8	24.6	0.0	26.2
retain=0.2,scale=0.1	23.6	6.4	25.3	26.3	2.1	28.1
retain=0.2,scale=0.2	23.6	4.6	24.0	29.2	2.3	26.0
retain=0.2,scale=0.3	24.1	5.3	26.0	28.5	1.9	29.2
retain=0.2,scale=0.4	25.6	7.1	26.0	27.9	1.2	26.9
retain=0.2,scale=0.5	26.1	0.5	21.0	27.5	3.4	25.2
retain=0.2,scale=0.6	26.8	0.0	27.3	27.2	0.3	23.9
retain=0.2,scale=0.7	24.9	0.0	21.4	25.0	0.3	23.1
retain=0.2,scale=0.8	24.8	0.0	24.1	24.8	0.0	28.3
retain=0.2,scale=0.9	23.3	0.0	23.9	24.8	0.0	25.8
retain=0.1,scale=0.1	23.2	6.2	25.7	27.5	1.5	30.0
retain=0.1,scale=0.2	22.9	5.7	25.0	27.5	3.5	30.6
retain=0.1,scale=0.3	23.6	6.4	24.4	27.5	3.4	30.8
retain=0.1,scale=0.4	22.6	10.7	23.4	29.3	5.3	28.8
retain=0.1,scale=0.5	24.3	6.7	25.0	27.0	2.9	27.3
retain=0.1,scale=0.6	24.8	0.3	24.3	25.0	0.9	27.3
retain=0.1,scale=0.7	27.1	0.0	26.9	26.3	0.5	24.4
retain=0.1,scale=0.8	27.1	0.0	20.4	25.8	0.3	27.1
retain=0.1,scale=0.9	24.8	0.0	25.8	26.1	0.2	29.1
EMR Merging	25.0	10.4	34.8	40.8	17.6	41.8
IP Merging						
Sim threshold=0.1	25.4	14.5	34.0	59.3	27.8	50.2
Sim threshold=0.2	26.2	14.8	34.2	59.7	28.0	50.2
Sim threshold=0.3	26.4	15.3	34.4	59.8	27.9	49.8
Sim threshold=0.4	28.4	14.7	33.9	59.7	28.0	49.8
Sim threshold=0.5	26.4	14.9	34.2	59.7	27.9	49.8
Sim threshold=0.6	27.3	14.6	34.2	60.2	28.5	50.7
Sim threshold=0.7	27.6	14.4	34.2	60.1	28.5	50.7
Sim threshold=0.8	27.0	14.4	34.2	60.1	28.4	50.7
Sim threshold=0.9	27.3	14.3	34.2	60.1	28.4	50.7

C.5 Merge Different System-1 LLMs and System-2 LLMs

By employing the proposed method, we conduct experiments on merging different reasoning pattern LLMs with base MLLM LLaVA-7B in table 10. We compare system-1 thinking LLMs such as WizardMath, MetaMath, Tora and Tora-code models. Merging Tora-code yields the best performance. Different from other models, Tora-code uses the high-quality reasoning data involved with the critique process. We believe Tora outperforms others for two main reasons: (1) CoT & PoT Collaboration: Tora employs a collaborative reasoning approach that integrates CoT and PoT to solve problems. In contrast, WizardMath and MetaMath rely solely on CoT. (2) Reflection Mechanism: Tora’s training data incorporates a reflection-based correction process, where incorrect responses are analyzed and revised, contributing to improved reasoning abilities. Others merely filter incorrect reasoning data. By further fine-tuning the code llama, Tora-code is able to perform code-like reasoning on math problems, which exhibits strong performance compared to other math LLM on text-based math reasoning datasets such as GSM8K and Math. This experiment also reveals one interesting observation: math reasoning abilities obtained by Tora series models are more transferable to MLLM. We further conduct experiments on merging system-2 thinking LLMs such as OpenO1 [25] and the DeepSeek-R1-distilled-LLaMA3 model [9], demonstrating the effectiveness of our method. Our method further improves the model performance by merging the long CoT LLMs, obtaining a 2.3% performance gain on average.

Table 10: Experiments of merging different System-1 LLMs and System-2 LLMs on MathVista.

MathLLM	MathVista						
	Params	TQA	GPS	VQA	FQA	MWP	Avg
<i>LLaVA-1.5-7B with System-1 LLMs</i>							
Base Model	7B	36.1	22.1	37.4	20.8	14.0	25.2
MetaMath-V1.0-7B	7B	36.1	22.6	34.1	23.4	15.6	25.7 +0.5 ↑
WizardMath-7B	7B	41.4	25.0	34.1	23.4	13.4	26.6 +1.4 ↑
Tora-V1.0-7B	7B	39.9	26.9	34.6	26.7	12.9	27.4 +2.2 ↑
Tora-code-V1.0-7B	7B	43.7	21.6	40.8	24.9	15.1	28.2 +3.0 ↑
<i>LLaVA-1.6-LLaMA3-8B with System-2 LLMs</i>							
Base Model	8B	50.6	29.3	38.6	46.5	23.1	37.8
OpenO1-LLaMA3-8B	8B	51.9	29.8	40.2	46.5	24.7	38.7 +0.9 ↑
DeepSeek-R1-distilled-LLaMA3	8B	51.9	32.7	40.8	45.7	29.6	40.1 +2.3 ↑

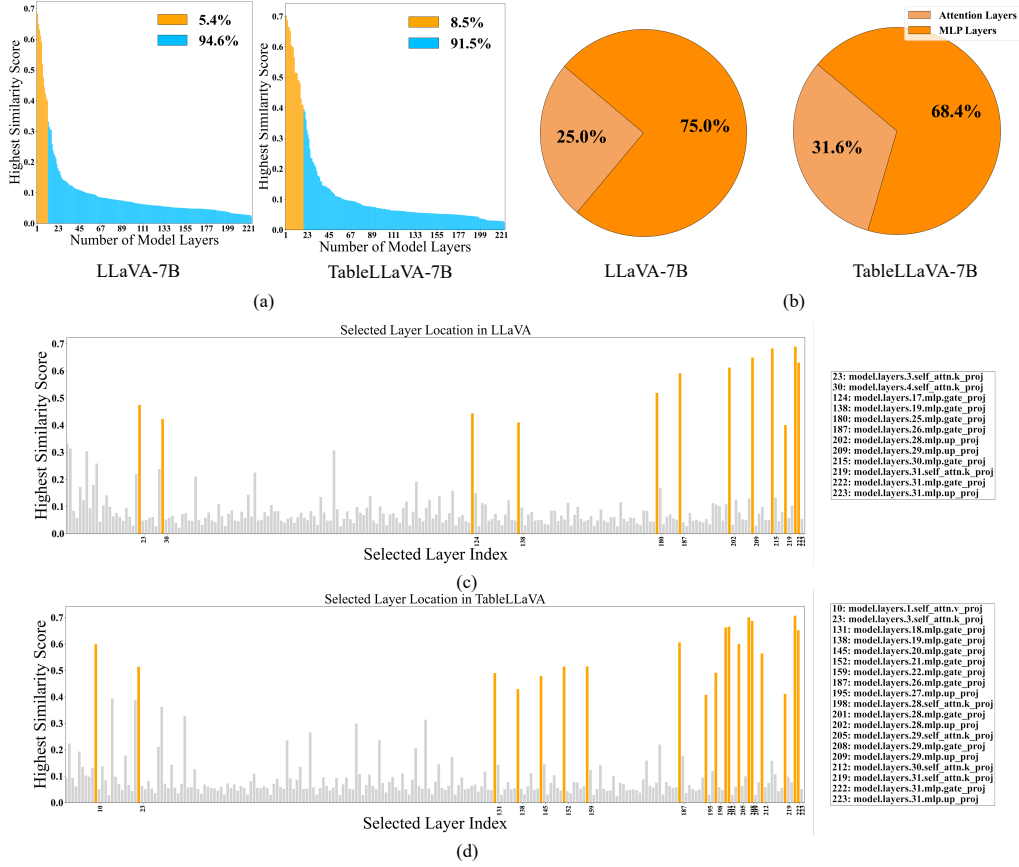


Figure 6: (a) The proportion of selected layer in two MLLMs. (b) The proportion of attention layers and MLP layers in selected layers.(c) Selected layers’ location in LLaVA. (d) Selected layers’ location in Table-LLaVA.

D Analysis of Selected Parameters

We visualize the proportion and composition of the selected parameters in fig. 6. As shown in the figure, the selected layers account for less than 10% of the total model parameters, with the majority concentrated in the MLP layers. This observation aligns with recent studies on knowledge storage in LLMs, which suggest that most knowledge and skills are encoded within the MLP layers [6, 47]. Since Table-LLaVA is fine-tuned on math reasoning datasets, it has already acquired a certain level of mathematical reasoning ability. Consequently, our selection process identifies a higher proportion of reasoning-related layers in Table-LLaVA compared to the base model, LLaVA. To further analyze the distribution of these selected layers, we plot their locations in fig. 6(c) and (d). The visualization reveals that most reasoning-associated layers are concentrated in the latter part of the model, suggesting that deeper layers play a crucial role in encoding mathematical reasoning skills.

1001 E Method Overview

1002 IP merging firstly identifies key parameters in both the MLLM and the math LLM. It then projects
 1003 the rescaled, selected parameters from the LLM into the subspace of the MLLM to achieve better
 1004 alignment. Finally, the aligned parameters are merged into the MLLM. During the parameter
 1005 identification phase, reasoning-related parameters are selected based on their similarity within a
 1006 shared subspace. In the projection phase, these parameters are rescaled and aligned to minimize the
 1007 discrepancy between the two models. The complete procedure is illustrated in fig. 7. We visualize
 the process of IP-Merging as follows:

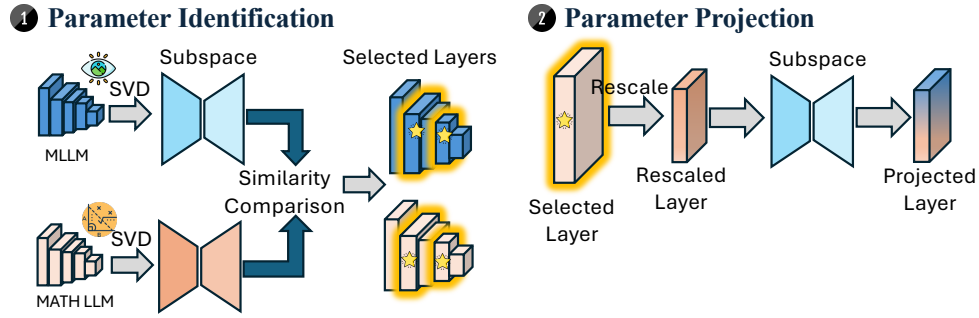


Figure 7: The general process of IP merging.

1008