

# Transferring Textual Preferences to Vision-Language Understanding through Model Merging

Anonymous ACL submission

## Abstract

Large vision-language models (LVLMs) perform outstandingly across various multimodal tasks. However, their ability to evaluate generated content remains limited, and training vision-language reward models (VLRMs) with preference data is computationally expensive. This paper explores a training-free alternative by merging text-based reward models (RMs) with LVLMs to create VLRMs. Our approach shows that integrating these models leads to improved performance over LVLMs' scoring and text-based RMs, offering an efficient method for incorporating textual preferences into LVLMs.

## 1 Introduction

Large vision-language models (LVLMs) have shown exceptional performance across a wide range of multimodal tasks (Hurst et al., 2024; Team et al., 2024; Anthropic, 2024), primarily due to the implementation of reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022), which utilizes preference data (Sun et al., 2024; Li et al., 2024b). This process often requires the use of reward models (RMs). However, LVLMs still struggle to assess generated content effectively (Chen et al., 2024a; Li et al., 2024a), and training an RM with preference data is resource-intensive.

In this work, we investigate an alternative approach: *Can knowledge derived from text-only preference data be transferred to LVLMs without additional training?* Several state-of-the-art LVLMs are built upon pre-trained language models with vision encoders and adapters (Dubey et al., 2024; Team, 2025; Lu et al., 2024). This architectural design suggests that textual preferences learned by text-based RMs may potentially integrate into LVLMs through parameter merging.

Building on this idea, we propose merging LVLMs with text-based RMs to create vision-language reward models (VLRMs), as illustrated

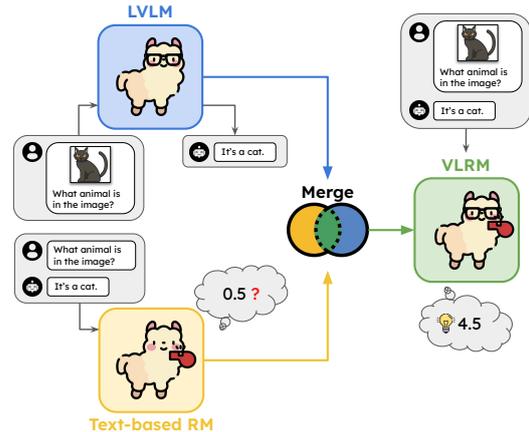


Figure 1: Framework for merging a text-based RM with an LVLM. LVLMs excel at visual tasks, while text-based RMs struggle to provide accurate rewards without visual cues. We transfer textual preferences to the vision-language understanding, resulting in a VLRM.

in Figure 1. Our approach leverages existing RMs and LVLMs, eliminating the need for costly multimodal preference data collection and training. We explore various merging strategies, ranging from simple weighted averaging (Wortsman et al., 2022) to advanced techniques such as task arithmetic (Iiharco et al., 2023), TIES (Yadav et al., 2024), and DARE (Yu et al., 2024a).

We assess performance using VL-RewardBench (Li et al., 2024a) and Best-of-N sampling with TextVQA (Singh et al., 2019) and MMMU-Pro (Yue et al., 2024b). The results show that our combined VLRMs outperform scoring through LVLMs and reward generation with text-based RMs. Our approach offers a training-free method for transferring textual preferences to LVLMs via model merging, and we provide a detailed analysis of merging strategies, demonstrating its effectiveness across multiple benchmarks.

## 2 Related Work

**Preference Dataset** A common approach to train a reward model is to use the Bradley-Terry

model (Bradley and Terry, 1952), which relies on paired data for learning. In NLP, many high-quality preference datasets are already available (Stienon et al., 2020; Bai et al., 2022; Ethayarajh et al., 2022; Köpf et al., 2023; Cui et al., 2024; Zhu et al., 2024; Wang et al., 2024). Similarly, in the vision-language domain, several preference datasets have been introduced (Yu et al., 2024b,c; Chen et al., 2024b; Wijaya et al., 2024; Li et al., 2024c; Zhou et al., 2024; Xiao et al., 2024). In this work, we explore the potential of transferring textual preferences to LVLMs in a training-free manner, specifically through model merging.

**LVLM-as-a-Judge & Evaluation** LVLM-as-a-Judge refers to utilizing strong large vision-language models for evaluation and judgment. These LVLMs can be either closed-source (OpenAI, 2023; Hurst et al., 2024; Team et al., 2024; Anthropic, 2024) or open-source (Lee et al., 2024; Dubey et al., 2024; Deitke et al., 2024; Team, 2025). To assess LVLMs as generative reward models, Chen et al. (2024a) established benchmarks and found that LVLMs exhibit high agreement with humans in pairwise comparison judgments, but perform poorly in scoring evaluation and batch ranking tasks. Recently, VL-RewardBench (Li et al., 2024a) introduced challenging cases and complex multimodal reasoning tasks, revealing that most off-the-shelf LVLMs struggle with such evaluations.

**Model Merging** Model merging is a common, training-free method for combining skills from multiple models within the parameter space. A basic approach involves simple weighted averaging (Wortsman et al., 2022), while more advanced techniques have been developed (Yadav et al., 2024; Yu et al., 2024a; Yang et al., 2024). These techniques have already proven effective in reward modeling (Rame et al., 2024; Lin et al., 2024) and LLM-as-a-judge (Kim et al., 2024) in NLP. Recently, REMEDY (Zhu et al., 2025) introduced strategies for merging LVLMs. In contrast, our work focuses on merging textual reward models into the language modeling components of LVLMs.

### 3 Methodology

We propose a training-free method to transfer textual preferences from a text-based RM  $\theta^{\text{RM}}$  to a LVLM  $\theta^{\text{LVLM}}$  through model merging.

Since both models originate from the same pre-trained language model  $\theta^{\text{PRE}}$ , we merge modules that appear in both models and preserve the

LVLM’s vision capabilities and text-based RM reward function, resulting in a VLRM that can assess textual and visual content without additional training. Below, we outline the components and merging strategies involved.

#### 3.1 Model Components

The pre-trained language model consists of:

$$\theta^{\text{PRE}} = \{\theta_{\text{emb}}^{\text{PRE}}, \theta_{\text{trans}}^{\text{PRE}}, \theta_{\text{lm}}^{\text{PRE}}\},$$

where  $\theta_{\text{emb}}^{\text{PRE}}$  is the embedding layer,  $\theta_{\text{trans}}^{\text{PRE}}$  is the transformer, and  $\theta_{\text{lm}}^{\text{PRE}}$  is the language modeling head, which maps the final hidden state of the transformer to the vocabulary.

The LVLM expands upon this with:

$$\theta^{\text{LVLM}} = \{\theta_{\text{venc}}^{\text{LVLM}}, \theta_{\text{adapt}}^{\text{LVLM}}, \theta_{\text{emb}}^{\text{LVLM}}, \theta_{\text{trans}}^{\text{LVLM}}, \theta_{\text{lm}}^{\text{LVLM}}\},$$

where  $\theta_{\text{venc}}^{\text{LVLM}}$  is the vision encoder, and  $\theta_{\text{adapt}}^{\text{LVLM}}$  is the adapter that integrates the vision encoder outputs into the language model.

Similarly, the text-based RM is defined as:

$$\theta^{\text{RM}} = \{\theta_{\text{emb}}^{\text{RM}}, \theta_{\text{trans}}^{\text{RM}}, \theta_{\text{rm}}^{\text{RM}}\},$$

where  $\theta_{\text{rm}}^{\text{RM}}$  is the reward modeling head, which projects the transformer’s final hidden state to a scalar value as the reward for a given input.

#### 3.2 Merging Strategies

We explore four merging strategies.

**Weighted Averaging** The weighted averaging strategy is defined as:

$$\theta_{\text{trans}}^{\text{MERGE}} = \lambda \cdot \theta_{\text{trans}}^{\text{LVLM}} + (1 - \lambda) \cdot \theta_{\text{trans}}^{\text{RM}},$$

where  $\lambda$  is a hyperparameter that controls the weight distribution between the two terms.

**Task Arithmetic** Task arithmetic strategy is defined as:

$$\tau^{\text{LVLM}} = \theta_{\text{trans}}^{\text{LVLM}} - \theta_{\text{trans}}^{\text{PRE}},$$

$$\tau^{\text{RM}} = \theta_{\text{trans}}^{\text{RM}} - \theta_{\text{trans}}^{\text{PRE}},$$

$$\theta_{\text{trans}}^{\text{MERGE}} = \lambda \cdot \tau^{\text{LVLM}} + \lambda \cdot \tau^{\text{RM}},$$

where  $\tau^{\text{LVLM}}$  represents the task vector derived from instruction tuning, and  $\tau^{\text{RM}}$  is the task vector obtained from reward modeling. The hyperparameter  $\lambda$  controls the contribution of the task vectors.

**TIES & DARE** For the TIES and DARE strategies, we simplify the expression to:

$$\theta_{\text{trans}}^{\text{MERGE}} = \lambda \cdot f(\tau^{\text{LVLM}}, d) + \lambda \cdot f(\tau^{\text{RM}}, d),$$

where  $f(\cdot)$  denotes the function for trimming, selecting, and rescaling the task vector, and  $d$  is the density determining how many parameters are retained. The two strategies apply different methods for trimming, selecting, and rescaling. See Appendix A for more details on TIES and DARE.

Method	VL-RewardBench				TextVQA	MMMU-Pro		
	General	Hallucination	Reasoning	Overall	Macro Avg.	Overall	Standard	Vision
Llama-3.2-Vision	33.3*	38.4*	56.6*	42.9*	42.8*	46.4	28.8	19.8
Tulu-2.5-RM	43.2	31.4	54.1	38.9	42.9	42.6	29.8	21.4
Random	<b>50.0</b>	50.0	50.0	50.0	50.0	48.2	29.2	18.4
Cascade	44.8	37.8	57.2	43.8	46.6	43.2	30.9	<b>23.4</b>
Linear	39.3	52.3	54.4	51.0	48.7	54.7	27.8	22.1
Task Vec.	48.6	59.4	59.7	57.9	55.9	59.0	31.0	22.7
TIES	43.7	58.2	58.5	56.2	53.5	<b>64.2</b>	29.1	22.6
DARE + Task Vec.	49.2	<b>61.7</b>	<b>61.0</b>	<b>59.7</b>	<b>57.3</b>	58.8	30.3	22.4
DARE + TIES	49.2	59.1	58.2	57.4	55.5	57.3	<b>31.6</b>	22.0

Table 1: Comparison of merging methods across the VL-RewardBench, TextVQA, and MMMU-Pro datasets using TULU-2.5-RM for merging. \*Indicates results from Li et al. (2024a).

### 3.3 Merged VLRM

The merged embedding parameters,  $\theta_{\text{emb}}^{\text{MERGE}}$  are obtained following standard embedding merging techniques outlined in MergeKit (Goddard et al., 2024), as detailed in Appendix A.

Finally, the merged VLRM  $\theta^{\text{MERGE}}$  is obtained by combining several components:

$$\theta^{\text{MERGE}} = \{\theta_{\text{venc}}^{\text{LVLM}}, \theta_{\text{adapt}}^{\text{LVLM}}, \theta_{\text{emb}}^{\text{MERGE}}, \theta_{\text{trans}}^{\text{MERGE}}, \theta_{\text{rm}}^{\text{RM}}\},$$

As a result, the merged VLRM can be used to provide rewards for both text and image content.

## 4 Experiments

### 4.1 Experimental Setup

#### 4.1.1 Models

In this paper, we employ Llama-3.2-11B-Vision-Instruct (Dubey et al., 2024) as our LVLM, referred to as Llama-3.2-Vision. For text-based RMs, we use Llama-3.1-Tulu-2-8B-uf-mean-rm (Iverson et al., 2024) and Llama-3.1-Tulu-3-8B-RM (Lambert et al., 2024), which we denote as Tulu-2.5-RM and Tulu-3-RM, respectively. All models derive from the same pre-trained language model Llama-3.1-8B. Our main results focus on Tulu-2.5-RM since it outperforms Tulu-3-RM on several VQA tasks with text-based input. Please refer to Appendix E for the model details.

#### 4.1.2 Model Merging

We use MergeKit for model merging and apply several techniques: weighted averaging, task arithmetic, TIES, and DARE—labeled as Linear, Task Vec., TIES, and DARE, respectively. Additionally, we explore combining DARE with task arithmetic and TIES for a more thorough analysis. To determine the optimal merging hyperparameters, we conduct a hyperparameter search and sample 400 instances from the RLAIIF-V (Yu et al., 2024c) training set as our validation set. More details are provided in Appendix A.

### 4.2 Reward Model Evaluation

#### 4.2.1 VL-RewardBench

We assess the merged VLRMs using VL-RewardBench (Li et al., 2024a), a benchmark that includes three domains: general multimodal instructions, hallucination-related tasks, and multimodal reasoning tasks. Each instance includes a multimodal query that consists of an image and a user prompt, along with a chosen response and a rejected response.

#### 4.2.2 Best-of-N Sampling

We assess our reward model’s effectiveness in enhancing performance through reranking using Best-of-N sampling, where  $N = 8$  in our work. This method scores and ranks responses to check if the highest-scoring one matches the correct answer. Specifically, we use Llama-3.2-11B-Vision-Instruct to generate eight candidates for the TextVQA (Singh et al., 2019) and MMMU-Pro (Yue et al., 2024b) datasets. See Appendix B for dataset details.

### 4.3 Main Results

Table 1 demonstrates the effectiveness of merging methods for combining an LVLM with a text-based RM. The baseline approaches include Llama-3.2-Vision, which utilizes the LVLM for direct scoring—pairwise scoring in VL-RewardBench and verbalized scoring in Best-of-N sampling tasks. Another baseline method, Tulu-2.5-RM, utilizes the text-based RM that focuses solely on evaluating the textual elements of questions and responses. We also incorporate a Random baseline that randomly selects responses. Furthermore, we implement a Cascade approach that employs a two-stage process: it first uses the LVLM to generate text descriptions of images based on the given question, then passes these de-

Method	VL-RB	TextVQA	MMMU-Pro	
	Overall	Overall	Standard	Vision
Linear	51.0	54.7	27.8	22.1
w/o image input	39.8	45.8	29.1	21.6
Task Vec.	57.9	59.0	31.0	22.7
w/o image input	44.9	38.7	31.8	21.0
TIES	56.2	64.2	29.1	22.6
w/o image input	42.7	40.9	31.2	21.0
DARE + Task Vec.	59.7	58.8	30.3	22.4
w/o image input	44.5	36.2	32.1	20.8
DARE + TIES	57.4	57.3	31.6	22.0
w/o image input	45.6	36.9	32.1	20.8

Table 2: Comparison of merging methods with and without image input, using TuLu-2.5-RM for merging. VL-RB stands for VL-RewardBench.

descriptions with the original text inputs through the text-based RM to produce final scores.

As shown in Table 1, merged VLRLMs consistently outperform Llama-3.2-Vision and TuLu-2.5-RM across nearly all merging methods and benchmarks. This result demonstrates that combining a text-based RM with an LVLMM effectively transfers textual preferences without training. Different merging strategies achieve the highest scores in different benchmarks, but overall, more advanced methods outperform simpler ones, highlighting the advantages of structured merging techniques. Additionally, in several benchmarks, merged VLRLMs surpass or match the strong Cascade baseline, suggesting that model merging captures more information than merely cascading two models. Notably, in VL-RewardBench, our merged VLRLMs even exceed the performance of the 90B LVLMM and achieve results comparable to commercial models. A similar trend emerges when using TuLu-3-RM as the text-based RM; further details are provided in Appendix G.1.

#### 4.4 Analysis

**Without Image Input** To further investigate whether the merged VLRLMs effectively use the vision encoder, we conduct an ablation study by evaluating the models without image input. As shown in Table 2, most models with image input outperform those without it across various merging techniques. This result suggests that the vision encoder plays an active role after merging, with performance gains not solely attributed to the text-based RM. These findings highlight how merging methods effectively combine textual and visual information. However, image input does not improve performance in the MMMU-Pro Standard

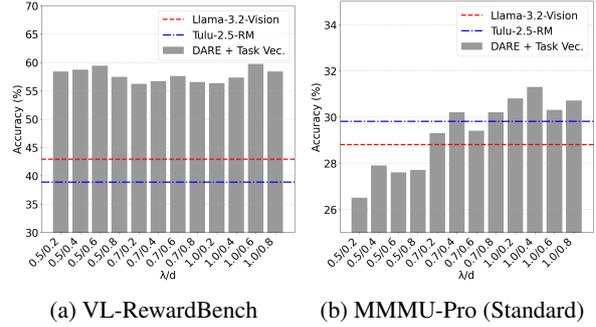


Figure 2: Effect of Dare + Task Vec. merging hyperparameters with TuLu-2.5-RM as the text-based RM.

set, likely because this set emphasizes reasoning, where reward assessments depend more on textual coherence than visual understanding, limiting the vision encoder’s contribution. A similar trend occurs when using TuLu-3-RM as the text-based RM; see Appendix G.2 for details.

**Effect of Merging Hyperparameters** We also investigate how merging hyperparameters impacts performance. Figure 2 presents the results of searching for  $d$  within the range [0.2, 0.4, 0.6, 0.8] and  $\lambda$  within [0.5, 0.7, 1.0] for DARE + Task Vec.. Our findings indicate that optimal hyperparameter values vary across benchmarks. For example, in VL-RewardBench,  $\lambda$  values do not have a significant effect, but in the MMMU-Pro standard set, we observe that  $\lambda = 1.0$  performs best. This variation indicates that the choice of hyperparameters affects the performance of the final merged VLRLM differently across tasks. Consequently, it highlights the importance of a well-curated validation set when selecting the optimal hyperparameters, which could be further explored in future research.

Furthermore, our results for  $d$  align with previous studies on TIES and DARE: even when task vectors are trimmed to lower rates (e.g., 0.4, 0.2), the merged VLRLMs maintain strong performance, consistent with the findings on LLM merging. For further hyperparameter search results across other methods and benchmarks, refer to Appendix G.3.

## 5 Conclusion

This work presents a training-free approach for integrating text-based RMs into LVLMMs through model merging. Our method enables the efficient transfer of textual preferences without the expensive multimodal preference data collection or additional training. Experimental results show that our approach outperforms LVLMM scoring and text-based RMs in multimodal reward assessment tasks.

## 306 Limitations

307 Our study has several limitations. First, we focused  
308 on a specific 11B vision-language model paired  
309 with an 8B text-based reward model, primarily due  
310 to limitations in computational resources. Addition-  
311 ally, we focused solely on the LLaMA architecture  
312 and did not explore alternatives like Qwen (Bai  
313 et al., 2023a,b) due to the absence of a suitable  
314 Qwen-based reward model for our experiments.  
315 Furthermore, we did not perform extensive abla-  
316 tion studies on the validation set. Our experimental  
317 results highlight the importance of a well-curated  
318 validation set in selecting optimal hyperparameters,  
319 which could be explored further in future research.  
320 Finally, due to the sensitivity of RLHF to hyperpa-  
321 rameter tuning and our computational constraints,  
322 we did not implement algorithms like PPO (Schul-  
323 man et al., 2017). Future work could explore inte-  
324 grating RLHF with merged VLRMs to assess its  
325 potential impact.

## 326 Ethics Statement

327 Our approach leverages pre-trained language and  
328 reward models, which may inherit biases from the  
329 training data. While merging models can enhance  
330 efficiency, it does not inherently mitigate existing  
331 biases. We encourage further research to evaluate  
332 and address potential biases in merged models to  
333 ensure fairness across diverse user groups.

## 334 References

335 Anthropic. 2024. [Claude 3.5 sonnet](#). Accessed: 2025-  
336 02-04.

337 Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang,  
338 Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei  
339 Huang, et al. 2023a. Qwen technical report. *arXiv*  
340 *preprint arXiv:2309.16609*.

341 Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang,  
342 Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou,  
343 and Jingren Zhou. 2023b. Qwen-vl: A frontier large  
344 vision-language model with versatile abilities. *arXiv*  
345 *preprint arXiv:2308.12966*.

346 Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda  
347 Askell, Anna Chen, Nova DasSarma, Dawn Drain,  
348 Stanislav Fort, Deep Ganguli, Tom Henighan,  
349 Nicholas Joseph, Saurav Kadavath, Jackson Kernion,  
350 Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac  
351 Hatfield-Dodds, Danny Hernandez, Tristan Hume,  
352 Scott Johnston, Shauna Kravec, Liane Lovitt, Neel  
353 Nanda, Catherine Olsson, Dario Amodei, Tom  
354 Brown, Jack Clark, Sam McCandlish, Chris Olah,  
355 Ben Mann, and Jared Kaplan. 2022. [Training](#)

a helpful and harmless assistant with reinforce-  
ment learning from human feedback. *Preprint*,  
arXiv:2204.05862. 356  
357  
358

Ralph Allan Bradley and Milton E Terry. 1952. Rank  
analysis of incomplete block designs: I. the method  
of paired comparisons. *Biometrika*, 39(3/4):324–  
345. 359  
360  
361  
362

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen  
Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao  
Wan, Pan Zhou, and Lichao Sun. 2024a. [MLLM-as-  
a-judge: Assessing multimodal LLM-as-a-judge with  
vision-language benchmark](#). In *Forty-first Interna-  
tional Conference on Machine Learning*. 363  
364  
365  
366  
367  
368

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Cong-  
hui He, Jiaqi Wang, Feng Zhao, and Dahua Lin.  
2024b. [Sharegpt4v: Improving large multi-modal  
models with better captions](#). In *European Confer-  
ence on Computer Vision*, pages 370–387. Springer. 369  
370  
371  
372  
373

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao,  
Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie,  
Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong  
Sun. 2024. [ULTRAFEEDBACK: Boosting language  
models with scaled AI feedback](#). In *Forty-first Inter-  
national Conference on Machine Learning*. 374  
375  
376  
377  
378  
379

Matt Deitke, Christopher Clark, Sangho Lee, Rohun  
Tripathi, Yue Yang, Jae Sung Park, Mohammadreza  
Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini,  
et al. 2024. [Molmo and pixmo: Open weights and  
open data for state-of-the-art multimodal models](#).  
*arXiv preprint arXiv:2409.17146*. 380  
381  
382  
383  
384  
385

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey,  
Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman,  
Akhil Mathur, Alan Schelten, Amy Yang, Angela  
Fan, et al. 2024. [The llama 3 herd of models](#). *arXiv*  
*preprint arXiv:2407.21783*. 386  
387  
388  
389  
390

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang,  
Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy  
Liang, and Tatsunori Hashimoto. 2023. [Alpacafarm:  
A simulation framework for methods that learn from  
human feedback](#). In *Thirty-seventh Conference on  
Neural Information Processing Systems*. 391  
392  
393  
394  
395  
396

Kawin Ethayarajh, Yejin Choi, and Swabha  
Swayamdipta. 2022. [Understanding dataset  
difficulty with  \$\mathcal{V}\$ -usable information](#). In *Proceedings  
of the 39th International Conference on Machine  
Learning*, volume 162 of *Proceedings of Machine  
Learning Research*, pages 5988–6008. PMLR. 397  
398  
399  
400  
401  
402

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 lan-  
guage models](#). In *Proceedings of the 2024 Confer-  
ence on Empirical Methods in Natural Language  
Processing: Industry Track*, pages 477–485, Miami,  
Florida, US. Association for Computational Linguis-  
tics. 403  
404  
405  
406  
407  
408  
409  
410  
411

412	Aaron Hurst, Adam Lerer, Adam P Goucher, Adam	Sujian Li, Bill Yuchen Lin, et al. 2024a. Vlr-	470
413	Perelman, Aditya Ramesh, Aidan Clark, AJ Os-	wardbench: A challenging benchmark for vision-	471
414	trow, Akila Welihinda, Alan Hayes, Alec Radford,	language generative reward models. <i>arXiv preprint</i>	472
415	et al. 2024. Gpt-4o system card. <i>arXiv preprint</i>	<i>arXiv:2411.17451</i> .	473
416	<i>arXiv:2410.21276</i> .		
417	Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Worts-	Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi	474
418	man, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali	Wang, Liang Chen, Yazheng Yang, Benyou Wang,	475
419	Farhadi. 2023. <a href="#">Editing models with task arithmetic</a> .	Lingpeng Kong, and Qi Liu. 2024b. <a href="#">VLFeedback:</a>	476
420	In <i>The Eleventh International Conference on Learn-</i>	<a href="#">A large-scale AI feedback dataset for large vision-</a>	477
421	<i>ing Representations</i> .	<a href="#">language models alignment</a> . In <i>Proceedings of the</i>	478
422	Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu,	<i>2024 Conference on Empirical Methods in Natu-</i>	479
423	Valentina Pyatkin, Nathan Lambert, Noah A. Smith,	<i>ral Language Processing</i> , pages 6227–6246, Miami,	480
424	Yejin Choi, and Hannaneh Hajishirzi. 2024. <a href="#">Unpack-</a>	Florida, USA. Association for Computational Lin-	481
425	<a href="#">ing DPO and PPO: Disentangling best practices for</a>	guistics.	482
426	<a href="#">learning from preference feedback</a> . In <i>The Thirty-</i>	Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi	483
427	<i>eighth Annual Conference on Neural Information</i>	Wang, Liang Chen, Yazheng Yang, Benyou Wang,	484
428	<i>Processing Systems</i> .	Lingpeng Kong, and Qi Liu. 2024c. <a href="#">Vlfeedback:</a>	485
429	Hamish Ivison, Yizhong Wang, Valentina Pyatkin,	<a href="#">A large-scale ai feedback dataset for large vision-</a>	486
430	Nathan Lambert, Matthew Peters, Pradeep Dasigi,	<a href="#">language models alignment</a> . In <i>Proceedings of the</i>	487
431	Joel Jang, David Wadden, Noah A Smith, Iz Belt-	<i>2024 Conference on Empirical Methods in Natural</i>	488
432	agy, et al. 2023. Camels in a changing climate: En-	<i>Language Processing</i> , pages 6227–6246.	489
433	hancing lm adaptation with tulu 2. <i>arXiv preprint</i>	Tzu-Han Lin, Chen-An Li, Hung-yi Lee, and Yun-Nung	490
434	<i>arXiv:2311.10702</i> .	Chen. 2024. <a href="#">DogeRM: Equipping reward models</a>	491
435	Seungone Kim, Juyoung Suk, Shayne Longpre,	<a href="#">with domain knowledge through model merging</a> . In	492
436	Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham	<i>Proceedings of the 2024 Conference on Empirical</i>	493
437	Neubig, Moontae Lee, Kyungjae Lee, and Minjoon	<i>Methods in Natural Language Processing</i> , pages	494
438	Seo. 2024. <a href="#">Prometheus 2: An open source language</a>	15506–15524, Miami, Florida, USA. Association for	495
439	<a href="#">model specialized in evaluating other language mod-</a>	Computational Linguistics.	496
440	<a href="#">els</a> . In <i>Proceedings of the 2024 Conference on Empir-</i>	Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai	497
441	<i>ical Methods in Natural Language Processing</i> , pages	Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhu-	498
442	4334–4353, Miami, Florida, USA. Association for	oshu Li, Hao Yang, et al. 2024. <a href="#">Deepseek-vl: towards</a>	499
443	Computational Linguistics.	<a href="#">real-world vision-language understanding</a> . <i>arXiv</i>	500
444	Andreas Köpf, Yannic Kilcher, Dimitri von Rütte,	<i>preprint arXiv:2403.05525</i> .	501
445	Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens,	OpenAI. 2023. <a href="#">Gpt-4v system card</a> . Accessed: 2025-	502
446	Abdullah Barhoum, Duc Nguyen, Oliver Stan-	02-04.	503
447	ley, Richárd Nagyfi, Shahul ES, Sameer Suri,	Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida,	504
448	David Glushkov, Arnav Dantuluri, Andrew Maguire,	Carroll Wainwright, Pamela Mishkin, Chong Zhang,	505
449	Christoph Schuhmann, Huu Nguyen, and Alexan-	Sandhini Agarwal, Katarina Slama, Alex Ray, et al.	506
450	der Mattick. 2023. <a href="#">Openassistant conversations -</a>	2022. Training language models to follow instruc-	507
451	<a href="#">democratizing large language model alignment</a> . In	tions with human feedback. <i>Advances in neural in-</i>	508
452	<i>Advances in Neural Information Processing Systems</i> ,	<i>formation processing systems</i> , 35:27730–27744.	509
453	volume 36, pages 47669–47681. Curran Associates,	Alexandre Rame, Nino Vieillard, Leonard Hussenot,	510
454	Inc.	Robert Dadashi, Geoffrey Cideron, Olivier Bachem,	511
455	Nathan Lambert, Jacob Morrison, Valentina Pyatkin,	and Johan Ferret. 2024. <a href="#">WARM: On the benefits</a>	512
456	Shengyi Huang, Hamish Ivison, Faeze Brahma,	<a href="#">of weight averaged reward models</a> . In <i>Forty-first</i>	513
457	Lester James V Miranda, Alisa Liu, Nouha Dziri,	<i>International Conference on Machine Learning</i> .	514
458	Shane Lyu, et al. 2024. T\ " ulu 3: Pushing frontiers	John Schulman, Filip Wolski, Prafulla Dhariwal,	515
459	in open language model post-training. <i>arXiv preprint</i>	Alec Radford, and Oleg Klimov. 2017. Proxi-	516
460	<i>arXiv:2411.15124</i> .	mal policy optimization algorithms. <i>arXiv preprint</i>	517
461	Seongyun Lee, Seungone Kim, Sue Park, Geewook	<i>arXiv:1707.06347</i> .	518
462	Kim, and Minjoon Seo. 2024. <a href="#">Prometheus-vision:</a>	Amanpreet Singh, Vivek Natarajan, Meet Shah,	519
463	<a href="#">Vision-language model as a judge for fine-grained</a>	Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh,	520
464	<a href="#">evaluation</a> . In <i>Findings of the Association for Compu-</i>	and Marcus Rohrbach. 2019. Towards vqa models	521
465	<i>tational Linguistics: ACL 2024</i> , pages 11286–11315,	that can read. In <i>Proceedings of the IEEE/CVF con-</i>	522
466	Bangkok, Thailand. Association for Computational	<i>ference on computer vision and pattern recognition</i> ,	523
467	Linguistics.	pages 8317–8326.	524
468	Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yi-		
469	fan Song, Peiyi Wang, Chenxin An, Tianyu Liu,		

525	Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. <a href="#">Learning to summarize with human feedback</a> . In <i>Advances in Neural Information Processing Systems</i> , volume 33, pages 3008–3021. Curran Associates, Inc.	581
526		582
527		583
528		584
529		585
530		
531		
532	Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. 2024. <a href="#">Aligning large multimodal models with factually augmented RLHF</a> . In <i>Findings of the Association for Computational Linguistics: ACL 2024</i> , pages 13088–13110, Bangkok, Thailand. Association for Computational Linguistics.	586
533		587
534		588
535		589
536		590
537		591
538		592
539		593
540	Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. <i>arXiv preprint arXiv:2403.05530</i> .	594
541		595
542		596
543		597
544		598
545		599
546	Qwen Team. 2025. <a href="#">Qwen2.5-vl</a> .	
547	Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. 2024. <a href="#">Helpsteer 2: Open-source dataset for training top-performing reward models</a> . In <i>The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track</i> .	600
548		601
549		602
550		603
551		604
552		605
553		606
554	Robert Wijaya, Ngoc-Bao Nguyen, and Ngai-Man Cheung. 2024. <a href="#">Multimodal preference data synthetic alignment with reward model</a> . <i>Preprint</i> , arXiv:2412.17417.	607
555		608
556		609
557		610
558	Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In <i>International conference on machine learning</i> , pages 23965–23998. PMLR.	611
559		612
560		613
561		614
562		615
563		616
564		617
565		
566	Wenyi Xiao, Ziwei Huang, Leilei Gan, Wanggui He, Haoyuan Li, Zhelun Yu, Hao Jiang, Fei Wu, and Linchao Zhu. 2024. Detecting and mitigating hallucination in large vision language models via fine-grained ai feedback. <i>arXiv preprint arXiv:2404.14233</i> .	618
567		619
568		620
569		621
570		622
571	Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. <i>Advances in Neural Information Processing Systems</i> , 36.	623
572		624
573		625
574		626
575		627
576	Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2024. <a href="#">Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities</a> . <i>Preprint</i> , arXiv:2408.07666.	628
577		629
578		630
579		631
580		632
	Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024a. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In <i>Forty-first International Conference on Machine Learning</i> .	633
		634
		635
		636
		637
	Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwan He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, and Tat-Seng Chua. 2024b. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In <i>Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)</i> , pages 13807–13816.	
	Tianyu Yu, Haoye Zhang, Yuan Yao, Yunkai Dang, Da Chen, Xiaoman Lu, Ganqu Cui, Taiwan He, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2024c. Rlaif-v: Aligning mllms through open-source ai feedback for super gpt-4v trustworthiness. <i>arXiv preprint arXiv:2405.17220</i> .	
	Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. 2024a. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In <i>Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition</i> , pages 9556–9567.	
	Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. 2024b. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. <i>arXiv preprint arXiv:2409.02813</i> .	
	Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. 2024. Lmms-eval: Reality check on the evaluation of large multimodal models. <i>arXiv preprint arXiv:2407.12772</i> .	
	Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. <a href="#">Wildchat: 1m chatGPT interaction logs in the wild</a> . In <i>The Twelfth International Conference on Learning Representations</i> .	
	Yiyang Zhou, Chenhang Cui, Rafael Rafailov, Chelsea Finn, and Huaxiu Yao. 2024. Aligning modalities in vision large language models via preference fine-tuning. In <i>ICLR 2024 Workshop on Reliable and Responsible Foundation Models</i> .	
	Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, Karthik Ganesan, Wei-Lin Chiang, Jian Zhang, and Jiantao Jiao. 2024. <a href="#">Starling-7b: Improving helpfulness and harmlessness with RLAI</a> . In <i>First Conference on Language Modeling</i> .	
	Didi Zhu, Yibing Song, Tao Shen, Ziyu Zhao, Jinluan Yang, Min Zhang, and Chao Wu. 2025. <a href="#">REMEDY: Recipe merging dynamics in large vision-language models</a> . In <i>The Thirteenth International Conference on Learning Representations</i> .	

## A Merging Details

**Weighted Averaging** Wortsman et al. (2022) showed that combining the weights of multiple models fine-tuned with varying hyperparameter settings often leads to improved accuracy and robustness. In this work, we employ a weighted averaging strategy as a straightforward method to merge a large vision-language model with a text-based reward model. The weighted averaging strategy is formally defined as:

$$\theta_{\text{trans}}^{\text{MERGE}} = \lambda \cdot \theta_{\text{trans}}^{\text{LVLM}} + (1 - \lambda) \cdot \theta_{\text{trans}}^{\text{RM}},$$

where  $\lambda$  is a hyperparameter that determines the weight distribution between the two models. We explore  $\lambda$  values in the range: [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0].

**Task Arithmetic** Ilharco et al. (2023) demonstrated that the task vector, obtained by subtracting the weights of a pre-trained model from those of the same model after fine-tuning for a specific task, defines the task direction. Utilizing this task vector can improve task performance. We also apply the task arithmetic approach to develop a vision-language reward model. The task arithmetic strategy is formally defined as:

$$\begin{aligned}\tau_{\text{trans}}^{\text{LVLM}} &= \theta_{\text{trans}}^{\text{LVLM}} - \theta_{\text{trans}}^{\text{PRE}}, \\ \tau_{\text{trans}}^{\text{RM}} &= \theta_{\text{trans}}^{\text{RM}} - \theta_{\text{trans}}^{\text{PRE}}, \\ \theta_{\text{trans}}^{\text{MERGE}} &= \lambda \cdot \tau_{\text{trans}}^{\text{LVLM}} + \lambda \cdot \tau_{\text{trans}}^{\text{RM}},\end{aligned}$$

where  $\tau_{\text{trans}}^{\text{LVLM}}$  denotes the task vector derived from instruction tuning, and  $\tau_{\text{trans}}^{\text{RM}}$  refers to the task vector obtained from reward modeling. The hyperparameter  $\lambda$  controls the relative contribution of task vectors. We explore  $\lambda$  values in the range: [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0].

**TIES** Yadav et al. (2024) consider the interference between parameters from different models during the model merging process. Their approach consists of three main steps. First, they prune task vector values based on magnitude, retaining only a proportion  $d$  of the task vector. Second, they resolve sign conflicts by calculating the total magnitude of parameter values in positive and negative directions and selecting the direction with the larger total magnitude. Only values that match the chosen sign are retained. Finally, they compute the mean of the retained values to determine the final parameter value. The TIES method can be simply

expressed as:

$$\theta_{\text{trans}}^{\text{MERGE}} = \lambda \cdot f(\tau_{\text{trans}}^{\text{LVLM}}, d) + \lambda \cdot f(\tau_{\text{trans}}^{\text{RM}}, d),$$

where  $f(\cdot)$  denotes the function for trimming, selecting, and rescaling the task vector, and  $d$  is the density determining how many parameters are retained. We search for optimal values of  $\lambda$  within the range [0.5, 0.7, 1.0] and  $d$  within the range [0.2, 0.4, 0.6, 0.8].

**DARE** Yu et al. (2024a) also addresses the interference between parameters from different models during the model merging process. They randomly drop delta parameters with a probability of  $p$  and rescale the remaining ones by  $1/(1-p)$ . The DARE method can be combined with both the Task Arithmetic and TIES approaches. When combined with Task Arithmetic, a proportion  $p$  of task vectors is randomly dropped, and the remaining ones are rescaled by  $1/(1-p)$ . When DARE is combined with TIES, a proportion  $p$  of task vectors is randomly dropped, and the sign of each parameter is determined by comparing the total magnitude in the positive and negative directions. The sign corresponding to the larger total magnitude is selected, and only values matching this sign are retained. Their mean is then computed as the final parameter value, and the result is rescaled by  $1/(1-p)$ . The DARE method can also be expressed as:

$$\theta_{\text{trans}}^{\text{MERGE}} = \lambda \cdot f(\tau_{\text{trans}}^{\text{LVLM}}, d) + \lambda \cdot f(\tau_{\text{trans}}^{\text{RM}}, d),$$

where  $d$  represents the density, determining the proportion of retained parameters, with  $d = 1 - p$ . We search for optimal values of  $\lambda$  within the range [0.5, 0.7, 1.0] and  $d$  within the range [0.2, 0.4, 0.6, 0.8].

**Merging Embeddings** We follow the embedding merging procedure from MergeKit (Goddard et al., 2024). The process is as follows:

1. If a token exists in the pre-trained model, we use its embedding from that model.
2. If a token appears in only one model (either the LVLM or the text-based RM), we use its embedding from that model.
3. If a token appears in multiple models, we compute the average of its embeddings.

Notably, the pre-trained model is not required for the weighted averaging method. Therefore, we omit the first step when applying this merging approach.

729	<b>Merging Hyperparameter Selection</b>	We select the merging hyperparameter by using a sampled set of 400 instances from the RLAIIF-V (Yu et al., 2024c) training set as our validation set. In case of a tie in scores, an additional 100 sampled instances will be used for evaluation. Results are discussed in Appendix G.3.	778
730			779
731			780
732			781
733			782
734			783
735			784
736	<b>B Dataset Details</b>		785
737	<b>VL-RewardBench</b>	VL-RewardBench (Li et al., 2024a) is a benchmark comprising 1,250 high-quality examples spanning three domains: general multimodal instructions, hallucination-related tasks, and multimodal reasoning tasks. Each example includes a multimodal query—consisting of an image and a user prompt—along with a selected response and a rejected response.	786
738			787
739			788
740			789
741			790
742			791
743			792
744			793
745	<b>TextVQA</b>	TextVQA (Singh et al., 2019) is a dataset designed to evaluate the ability of visual question-answering (VQA) models to read and reason about text within images. We use its validation set, which contains 5,000 instances, to assess our merged VLRMs.	794
746			795
747			796
748			797
749			798
750			799
751	<b>MMMU-Pro</b>	MMMU-Pro (Yue et al., 2024b) is an advanced benchmark designed to assess the understanding and reasoning abilities of multimodal models. It is derived from the original MMMU (Yue et al., 2024a) dataset and consists of two subsets: a standard set, which includes image and text queries with 10 answer options, and a vision set, which features a vision-only input scenario. In the vision set, the questions are embedded within screenshots or photos, with no explicit text provided.	800
752			801
753			802
754			803
755			804
756			805
757			806
758			807
759			808
760			809
761			810
762	<b>RLAIIF-V</b>	RLAIIF-V (Yu et al., 2024c) preference dataset is created by generating multiple candidate responses for a given prompt and image using various random seeds. Each response is divided into individual claims, which are then assessed using an open-source large vision-language model. This model assigns confidence scores to each claim, which are combined to form an overall response score. Preference pairs are generated by comparing the response scores for the same prompt, selecting the preferred response and the less favorable one based on the score differences. Pairs with significant length disparities are excluded to avoid bias. We select 400 instances from this preference dataset to serve as our validation set for selecting the hyperparameters of merging methods.	811
763			812
764			813
765			814
766			815
767			816
768			817
769			818
770			819
771			820
772			821
773			822
774			823
775			824
776			825
777			
	<b>C Best-of-N Sampling Details</b>	We use lmms-eval (Zhang et al., 2024) for response generation with the Best-of-N sampling technique. For the TextVQA dataset, we set both the temperature and top-p to 1.0, sampling 8 responses. To encourage concise answers, we append “Answer the question using a single word or phrase.” after the generation prompt. For the MMMU-Pro dataset, we also set the temperature and top p to 1.0, with a maximum token limit of 4096, to sample 8 responses. Additionally, we apply chain-of-thought (CoT) for generating both answers and their reasoning.	
	<b>D Prompt Template</b>	For Best-of-N sampling using LLaMA-3.2-Vision as the generative reward model, the prompt template is provided in Table 3. For image captioning with LLaMA-3.2-Vision and reward modeling using Tulu-3-RM and Tulu-2.5-RM, the detailed prompt template can also be found in Table 3.	
	<b>E Open-Source Model Details</b>		
	<b>Llama-3.2-11B-Vision-Instruct</b>	Llama-3.2-11B-Vision-Instruct (Dubey et al., 2024) is an 11B-parameter LVM consisting of three main components: a vision encoder, an adapter, and a pre-trained language model. The language model is based on Llama-3.1-8B-Instruct. The adapter incorporates cross-attention layers to integrate image representations into the language model. During adapter training, the language model remains frozen, enabling seamless drop-in replacement for Llama-3.1 series models without requiring re-training.	
	<b>Tulu-2.5-RM</b>	Tulu-2.5-RM (Iverson et al., 2024) is a reward model initialized from Llama-3.1-8B and fine-tuned using the Tulu 2 recipe (Iverson et al., 2023). It is adapted for reward modeling by replacing the language modeling head with a linear layer and fine-tuning it on preference data from diverse sources, including Ultrafeedback (Cui et al., 2024), Nectar (Zhu et al., 2024), HH-RLHF (Bai et al., 2022), and AlpacaFarm (Dubois et al., 2023), among others.	
	<b>Tulu-3-RM</b>	Tulu-3-RM (Lambert et al., 2024) is another reward model initialized from Llama-3.1-8B and fine-tuned following the Tulu 3 recipe (Lambert et al., 2024). Like Tulu-2.5-RM, it is adapted for reward modeling by replacing	

the language modeling head with a linear layer. However, Tulu-3-RM is trained on a mixture of on-policy and off-policy preference data collected through an enhanced version of the Ultrafeedback (Cui et al., 2024) pipeline. This dataset includes prompts from various sources, such as the SFT dataset in the Tulu 3 recipe, WildChat (Zhao et al., 2024), Ultrafeedback (Cui et al., 2024), and synthetic persona-augmented instructions.

## F Qualitative Results

We investigate reward model behavior before and after merging, and we evaluate qualitatively on VL-RewardBench. Tables 4 and 5 present results for Tulu-2.5-RM, while Tables 6 and 7 show Tulu-3-RM. Red text indicates misalignment with the image. Before merging, the text-based reward model made incorrect predictions. After merging, the vision-language reward models correctly identified the better response. In most cases, more advanced merging methods—such as task arithmetic, TIES, and DARE—produce larger reward differences between chosen and rejected responses than simple weighted averaging.

## G Full Results

### G.1 Main Results

The main results of merging with Tulu-2.5-RM are discussed in Section 4.3 of the main text. As shown in Table 1, merged VLRMs consistently outperform Llama-3.2-Vision and Tulu-2.5-RM across nearly all merging methods and benchmarks. Notably, in VL-RewardBench, they show the greatest improvement in the Hallucination domain. In Best-of-N evaluation, they perform well in both TextVQA and MMMU-Pro. Additionally, merged VLRMs match or surpass the strong Cascade baseline, suggesting that merging captures more information than simply cascading two models.

A similar trend is observed when merging with Tulu-3-RM. As shown in Table 8, merged VLRMs outperform Llama-3.2-Vision and Tulu-3-RM across most methods and benchmarks. In VL-RewardBench, they improve mainly in the General and Hallucination domains. For Best-of-N evaluation, they perform well in MMMU-Pro, but only a few achieve results comparable to Llama-3.2-Vision in TextVQA, likely due to Tulu-3-RM’s weaker performance in this task. While merging with Llama-3.2-Vision enhances

performance over Tulu-3-RM, it does not surpass Llama-3.2-Vision’s score. Additionally, merged VLRMs exceed the strong Cascade baseline in other benchmarks and remain competitive with it in TextVQA.

In Table 11, we compare our merged VLRMs with large open-source LVRMs and commercial systems on VL-RewardBench. Surprisingly, our merged VLRMs outperform 90B LVRMs and achieve performance comparable to commercial models, demonstrating the effectiveness of transferring textual preferences from text-based RMs to LVRMs.

### G.2 Without Image Input

We conduct an ablation study by evaluating models without image input. Full results with Tulu-2.5-RM are shown in Table 9. Models with image input consistently outperform those without it across various merging techniques, suggesting that the vision encoder actively contributes after merging rather than performance gains being solely due to the text-based RM. This indicates that merged VLRMs effectively utilize the vision encoder in most cases. Notably, in VL-RewardBench, merged VLRMs match or surpass those without image input, especially in the hallucination domain, where image input significantly improves performance. In Best-of-N evaluation, models with image input perform better in the TextVQA and MMMU-Pro Vision sets. However, in the MMMU-Pro Standard set, image input does not provide an advantage, likely because this set emphasizes text reasoning, where reward assessments depend more on textual coherence than visual information.

Full results with Tulu-3-RM are shown in Table 10, following a similar trend. In VL-RewardBench, merged VLRMs outperform those without image input in the hallucination domain and are comparable to or surpass them in general and reasoning domains. Image input also enhances Best-of-N evaluation, particularly in TextVQA and MMMU-Pro Vision. However, in the MMMU-Pro Standard, image input does not provide a clear advantage, reaffirming that this set prioritizes text reasoning over visual input.

### G.3 Effect of Merging Hyperparameters

In this study, we optimize hyperparameter merging using sampled instances from RLAIIF-V. The results, based on 400 sampled RLAIIF-V instances used as a validation set, are presented in Tables 12

925 to 21. Bold text highlights the best performance,  
926 while **text with \*** indicates cases where scores are  
927 tied. In these cases, an additional 100 samples are  
928 used, and \* marks the top-performing result among  
929 them.

930 Figures 3 to 12 show the effect of hyperparame-  
931 ters across various benchmarks, merging methods,  
932 and text-based RMs. The results reveal that op-  
933 timal hyperparameters differ across these factors,  
934 emphasizing the importance of a well-constructed  
935 validation set. Future research could further ex-  
936 plore this. For example, Figure 3 shows the results  
937 of searching for  $\lambda$  values between 0 and 1 for the  
938 Linear method using Tulu-2.5-RM. In the VL-  
939 RewardBench, a mid-range  $\lambda$  produces the best  
940 performance, while in the MMMU-Pro vision set,  
941 a smaller  $\lambda$  yields better results. This variation  
942 suggests that hyperparameter choices influence the  
943 performance of the final merged VLRMs differ-  
944 ently depending on the task.

945 Moreover, we observe a trend consistent with  
946 prior studies (Yadav et al., 2024; Yu et al., 2024a):  
947 even when task vectors are reduced to lower rates  
948 (e.g., 0.4, 0.2), merged VLRMs continue to per-  
949 form well, aligning with findings on LLM merging.

<b>Best-of-N Sampling Prompt template for Llama-3.2-Vision</b>
<pre> &lt; start_header_id &gt;system&lt; end_header_id &gt; You are a highly capable multimodal AI assistant tasked with evaluating answers to visual questions. &lt; eot_id &gt;&lt; start_header_id &gt;user&lt; end_header_id &gt;  Please analyze the following image and question, then evaluate the provided answer:  Question:  {INSTRUCTION}  Answer:  {RESPONSE}  Evaluate the answer based on the following criteria: 1. Accuracy: How well does the answer align with the visual information in the image? Score: [1 (Poor) to 5 (Excellent)]  2. Completeness: Does the answer fully address all aspects of the question? Score: [1 (Poor) to 5 (Excellent)]  3. Clarity: Is the answer well-articulated and easy to understand? Score: [1 (Poor) to 5 (Excellent)]  4. Relevance: Does the answer directly relate to the question and the image? Score: [1 (Poor) to 5 (Excellent)]  After your evaluation, please include: 1. Reasoning: A detailed explanation for each criterion, highlighting why you assigned the given score. 2. Overall Assessment: Provide a n overall quality score (1 to 5) for the answer. &lt; eot_id &gt; </pre>
<b>Image Captioning Prompt template using Llama-3.2-Vision</b>
<pre> &lt; start_header_id &gt;system&lt; end_header_id &gt; You are a highly capable multimodal AI assistant tasked with evaluating answers to visual questions. &lt; eot_id &gt;&lt; start_header_id &gt;user&lt; end_header_id &gt;  {IMAGE} Please describe this image according to the given question: {INSTRUCTION} &lt; eot_id &gt; </pre>
<b>Prompt template for reward modeling</b>
<pre> &lt; start_header_id &gt;user&lt; end_header_id &gt;  {INSTRUCTION} &lt; eot_id &gt;&lt; start_header_id &gt;assistant&lt; end_header_id &gt;  {RESPONSE} &lt; eot_id &gt; </pre>

Table 3: Prompt template for best-of-n sampling, image captioning and reward modeling.

**Image:**

**Query:** Describe this image in detail.

**Response 1:**

The image showcases a lively scene of a couple of kids playing soccer on a field. They are kicking a sports ball around while several people, including young men and other onlookers, are watching and enjoying the game.

The spectators are spread throughout the scene, with some standing in different spots around the grassy area. There are also numerous chairs set up in the vicinity, possibly for the audience to take a rest or gather at a later time. Furthermore, various personal items, such as backpacks and a handbag, can be observed near the chairs.

Additionally, a few cars and an umbrella are present in the background, hinting at an outdoor, casual setting for this friendly soccer match.

**Response 2:**

The image showcases a lively scene of a couple of kids playing soccer on a field. They are kicking a sports ball around while several people, including young men and other onlookers, are watching and enjoying the game. The spectators are spread throughout the scene, with some standing in different spots around the grassy area. There are also numerous chairs set up in the vicinity, possibly for the audience to take a rest or gather at a later time. **Moreover, the soccer goalposts stand tall, ready for the players to score.** Additionally, the grassy terrain adds to the natural charm of the setting, making it an ideal place for a friendly soccer match.

**Score comparison:**

Method	Response 1 Score	Response 2 Score
Tulu-2.5-RM	2.17188	<b>2.27930</b>
Linear	<b>1.89551</b>	1.60156
Task Vec.	<b>3.50391</b>	1.82422
TIES	<b>2.67773</b>	1.96387
DARE + Task Vec.	<b>3.51758</b>	1.69141
DARE + TIES	<b>3.44336</b>	1.65430

Table 4: Qualitative results on VL-RewardBench using TULU-2.5-RM for merging.

**Image:**



**Query:** What is still on the tray?

**Response 1:**

There are still two portions of fries left on the tray.

**Response 2:**

There are still **two hot dogs** and some fries left on the tray.

**Score comparison:**

Method	Response 1 Score	Response 2 Score
Tulu-2.5-RM	0.56885	<b>1.20508</b>
Linear	<b>0.60986</b>	0.38477
Task Vec.	<b>2.47461</b>	0.55273
TIES	<b>1.76953</b>	1.10840
DARE + Task Vec.	<b>2.5918</b>	0.62988
DARE + TIES	<b>2.46484</b>	0.43701

Table 5: Qualitative results on VL-RewardBench using TULU-2.5-RM for merging.

**Image:**



**Query:** How many bananas are there in the image?

**Response 1:**

In the picture, there are at least 26 bananas. The bananas are divided into several bunches, which may indicate that they vary in size and ripeness.

**Response 2:**

There are 47 bananas in the picture. The bananas are divided into several bunches, which may indicate that their size and ripeness are different.

**Score comparison:**

Method	Response 1 Score	Response 2 Score
Tulu-3-RM	1.4043	<b>2.33594</b>
Linear	<b>1.33984</b>	0.51074
Task Vec.	<b>5.58203</b>	4.90234
TIES	<b>7.99219</b>	7.55469
DARE + Task Vec.	<b>4.35938</b>	2.88672
DARE + TIES	<b>5.51953</b>	5.20312

Table 6: Qualitative results on VL-RewardBench using TULU-3-RM for merging.

**Image:**



**Query:** Describe this image in detail.

**Response 1:**

In the image, there is a twin-propeller airplane parked on the tarmac, surrounded by a crowd of people. Some people seem to be disembarking from the plane, while others are walking around in the vicinity. There are at least 15 people in total, spread out near the airplane.

Luggage is being unloaded, with a suitcase visible near the center of the scene and a handbag located towards the right side. It appears that the passengers are either arriving at their destination or preparing to board the aircraft, as the luggage is being sorted and attended to.

**Response 2:**

In the image, the flight crew consisting of a pilot and co-pilot can be seen conversing near the twin-propeller airplane parked on the tarmac. A crowd of people surrounds them, with some individuals disembarking from the plane and others preparing to board. Ground support equipment such as baggage carts and fuel trucks are visible in the background near the airport terminal building. Luggage is being unloaded and sorted, with a suitcase and handbag visible in the foreground. The blue stripe on the airplane adds a pop of color to the scene. It appears that the passengers are either arriving at their destination or preparing to depart on their journey. An information sign can be seen towards the left side of the image.

**Score comparison:**

Method	Response 1 Score	Response 2 Score
Tulu-3-RM	3.94531	<b>4.74219</b>
Linear	<b>3.66016</b>	2.74414
Task Vec.	<b>5.23828</b>	2.99219
TIES	<b>7.72656</b>	5.67188
DARE + Task Vec.	<b>4.67188</b>	2.24414
DARE + TIES	<b>5.79688</b>	2.88477

Table 7: Qualitative results on VL-RewardBench using TULU-3-RM for merging.

Method	VL-RewardBench					TextVQA	MMMU-Pro	
	General	Hallucination	Reasoning	Overall	Macro Avg.	Overall	Standard	Vision
Llama-3.2-Vision	33.3*	38.4*	56.6*	42.9*	42.8*	46.4	28.8	19.8
Tulu-3-RM	45.4	36.6	56.6	43.0	46.2	27.4	29.4	20.4
Random	50.0	50.0	50.0	50.0	50.0	<b>48.2</b>	29.2	18.4
Cascade	54.1	40.5	57.2	46.7	50.6	38.3	31.3	<b>23.7</b>
Linear	47.5	51.0	55.0	51.5	51.2	45.8	29.1	19.0
Task Vec.	63.4	66.4	57.5	63.7	62.4	36.0	<b>31.6</b>	20.9
TIES	59.0	<b>74.1</b>	50.9	<b>66.0</b>	61.4	28.3	30.7	20.6
DARE + Task Vec.	63.4	68.9	<b>58.5</b>	65.4	<b>63.6</b>	36.1	30.2	20.9
DARE + TIES	<b>63.9</b>	65.6	57.2	63.2	62.2	56.9	31.4	21.8

Table 8: Comparison of merging methods across the VL-RewardBench, TextVQA, and MMMU-Pro datasets using TULU-3-RM for merging. \*Indicates results from Li et al. (2024a).

Method	VL-RewardBench					TextVQA	MMMU-Pro	
	General	Hallucination	Reasoning	Overall	Macro Avg.	Overall	Standard	Vision
Linear	39.3 (-2.2)	52.3 (+20.8)	54.4 (-4.1)	51.0 (+11.2)	48.7 (+4.9)	54.7 (+8.9)	27.8 (-1.3)	22.1 (+0.5)
w/o image input	41.5	31.5	58.5	39.8	43.8	45.8	29.1	21.6
Task Vec.	48.6 (+4.3)	59.4 (+20.4)	59.7 (+0.6)	57.9 (+13.0)	55.9 (+8.4)	59.0 (+20.3)	31.0 (-0.8)	22.7 (+1.7)
w/o image input	44.3	39.0	59.1	44.9	47.5	38.7	31.8	21.0
TIES	43.7 (-1.1)	58.2 (+23.0)	58.5 (-0.6)	56.2 (+13.5)	53.5 (+7.1)	64.2 (+23.3)	29.1 (-2.1)	22.6 (+1.6)
w/o image input	44.8	35.2	59.1	42.7	46.4	40.9	31.2	21.0
DARE + Task Vec.	49.2 (+4.4)	61.7 (+23.4)	61.0 (+2.2)	59.7 (+15.2)	57.3 (+10.0)	58.8 (+22.6)	30.3 (-1.8)	22.4 (+1.6)
w/o image input	44.8	38.3	58.8	44.5	47.3	36.2	32.1	20.8
DARE + TIES	49.2 (+3.3)	59.1 (+19.2)	58.2 (-0.6)	57.4 (+11.8)	55.5 (+7.3)	57.3 (+20.4)	31.6 (-0.5)	22.0 (+1.2)
w/o image input	45.9	39.9	58.8	45.6	48.2	36.9	32.1	20.8

Table 9: Full results comparing merging methods with and without image input, using TULU-2.5-RM for merging.

Method	VL-RewardBench					TextVQA	MMMU-Pro	
	General	Hallucination	Reasoning	Overall	Macro Avg.	Overall	Standard	Vision
Linear	47.5 (-1.1)	51.0 (+1.1)	55.0 (0.0)	51.5 (+0.5)	51.2 (0.0)	45.8 (+25.5)	29.1 (+0.5)	19.0 (-1.3)
w/o image input	48.6	49.9	55.0	51.0	51.2	20.3	28.6	20.3
Task Vec.	63.4 (+3.8)	66.4 (+19.3)	57.5 (+4.4)	63.7 (+13.2)	62.4 (+9.1)	36.0 (+1.2)	31.6 (-0.1)	20.9 (+0.3)
w/o image input	59.6	47.1	53.1	50.5	53.3	34.8	31.7	20.6
TIES	59.0 (-0.6)	74.1 (+33.5)	50.9 (-3.2)	66.0 (+19.2)	61.4 (+10.0)	28.3 (-0.3)	30.7 (-1.0)	20.6 (-0.9)
w/o image input	59.6	40.6	54.1	46.8	51.4	28.6	31.7	21.5
DARE + Task Vec.	63.4 (+3.8)	68.9 (+18.4)	58.5 (+2.2)	65.4 (+12.1)	63.6 (+8.2)	36.1 (-5.8)	30.2 (-1.9)	20.9 (+0.7)
w/o image input	59.6	50.5	56.3	53.3	55.4	41.9	32.1	20.2
DARE + TIES	63.9 (+8.7)	65.6 (+20.9)	57.2 (+1.9)	63.2 (+14.2)	62.2 (+10.4)	56.9 (+29.2)	31.4 (+0.6)	21.8 (+1.4)
w/o image input	55.2	44.7	55.3	49.0	51.8	27.7	30.8	20.4

Table 10: Full results comparing merging methods with and without image input, using TULU-3-RM for merging.

Method	General	Hallucination	Reasoning	Overall	Macro Avg.
<i>Open-Source Models*</i>					
Llama-3.2-Vision-11B-Instruct	33.3	38.4	56.6	42.9	42.8
Llama-3.2-Vision-90B-Instruct	42.6	57.3	61.7	56.2	53.9
Qwen2-VL-72B-Instruct	38.1	32.8	58.0	39.5	43.0
Molmo-72B-0924	33.9	42.3	54.9	44.1	43.7
NVLM-D-72B	38.9	31.6	62.0	40.1	44.1
<i>Proprietary Models*</i>					
Gemini-1.5-Flash (2024-09-24)	47.8	59.6	58.4	57.6	55.3
Gemini-1.5-Pro (2024-09-24)	50.8	72.5	64.2	<b>67.2</b>	62.5
Claude-3.5-Sonnet (2024-06-22)	43.4	55.0	62.3	55.3	53.6
GPT-4o-mini (2024-07-18)	41.7	34.5	58.2	41.5	44.8
GPT-4o (2024-08-06)	49.1	67.6	<b>70.5</b>	65.8	62.4
<i>Using TULU-2.5-RM for merging</i>					
Linear	39.3	52.3	54.4	51.0	48.7
Task Vec.	48.6	59.4	59.7	57.9	55.9
TIES	43.7	58.2	58.5	56.2	53.5
DARE + Task Vec.	49.2	61.7	61.0	59.7	57.3
DARE + TIES	49.2	59.1	58.2	57.4	55.5
<i>Using TULU-3-RM for merging</i>					
Linear	47.5	51.0	55.0	51.5	51.2
Task Vec.	63.4	66.4	57.5	63.7	62.4
TIES	59.0	<b>74.1</b>	50.9	66.0	61.4
DARE + Task Vec.	63.4	68.9	58.5	65.4	<b>63.6</b>
DARE + TIES	<b>63.9</b>	65.6	57.2	63.2	62.2

Table 11: Full results on VL-RewardBench, compared with current strong large vision-language models. \*Indicates results from Li et al. (2024a).

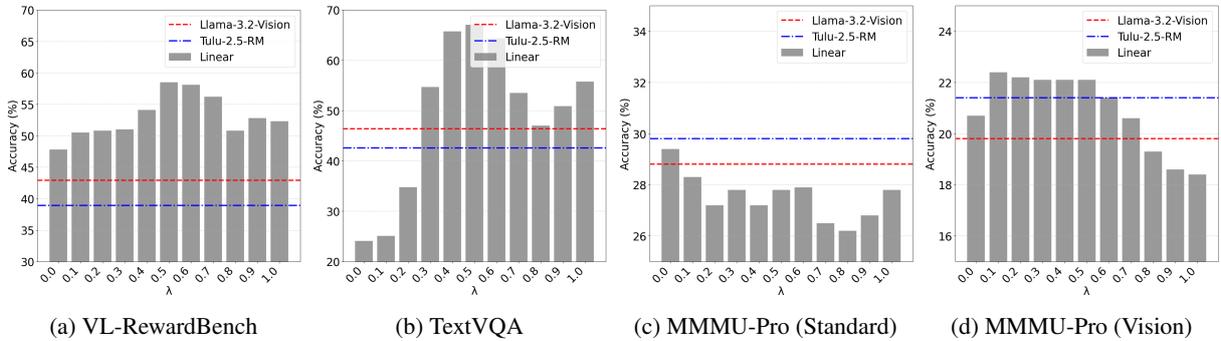


Figure 3: Full results of merging Llama-3.2-Vision and Tulu-2.5-RM (Linear)

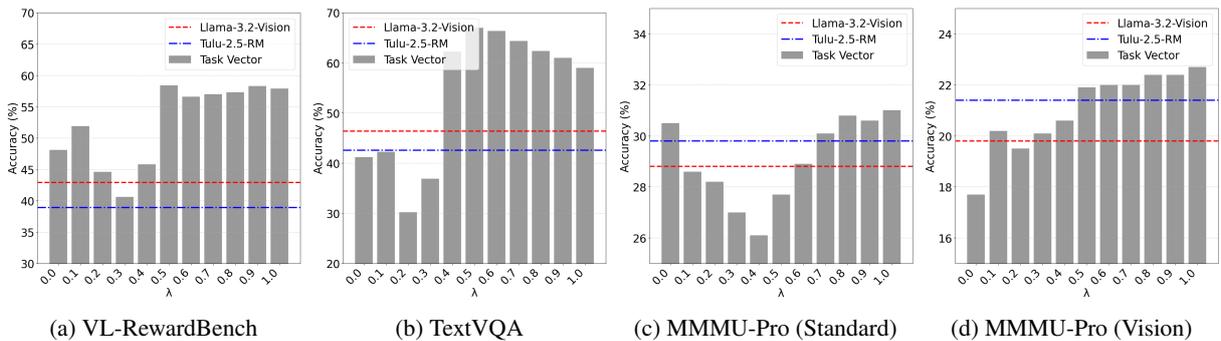


Figure 4: Full results of merging Llama-3.2-Vision and Tulu-2.5-RM (Task Vec.)

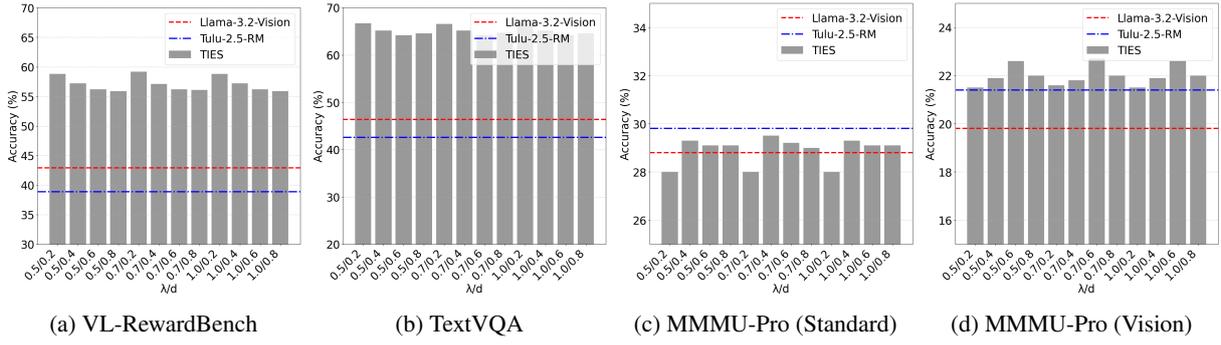


Figure 5: Full results of merging Llama-3.2-Vision and Tulu-2.5-RM (TIES)

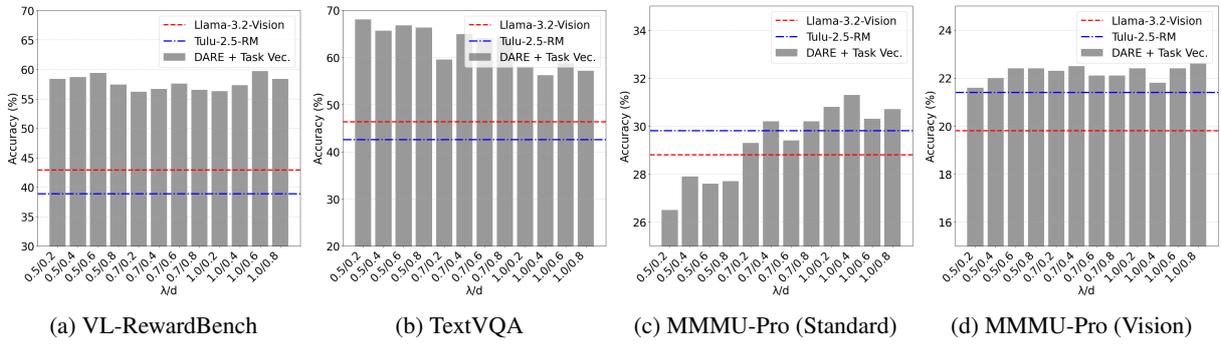


Figure 6: Full results of merging Llama-3.2-Vision and Tulu-2.5-RM (DARE + Task Vec.)

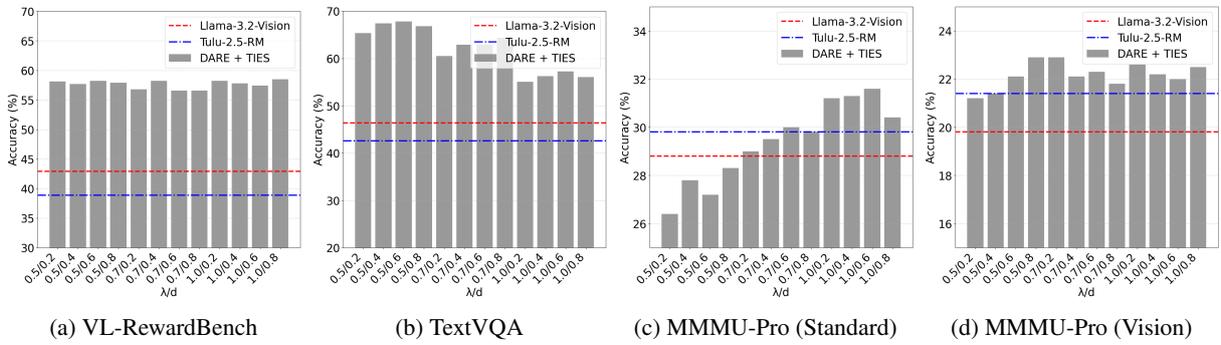


Figure 7: Full results of merging Llama-3.2-Vision and Tulu-2.5-RM (DARE + TIES)

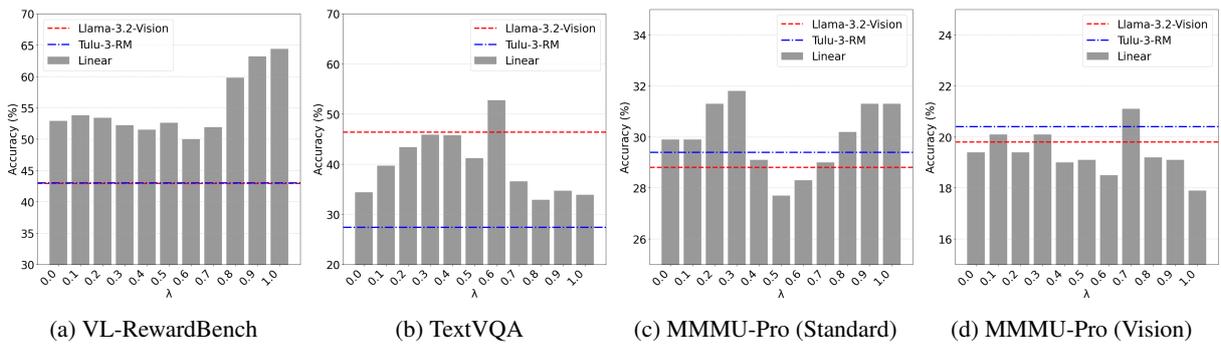


Figure 8: Full results of merging Llama-3.2-Vision and Tulu-3-RM (Linear)

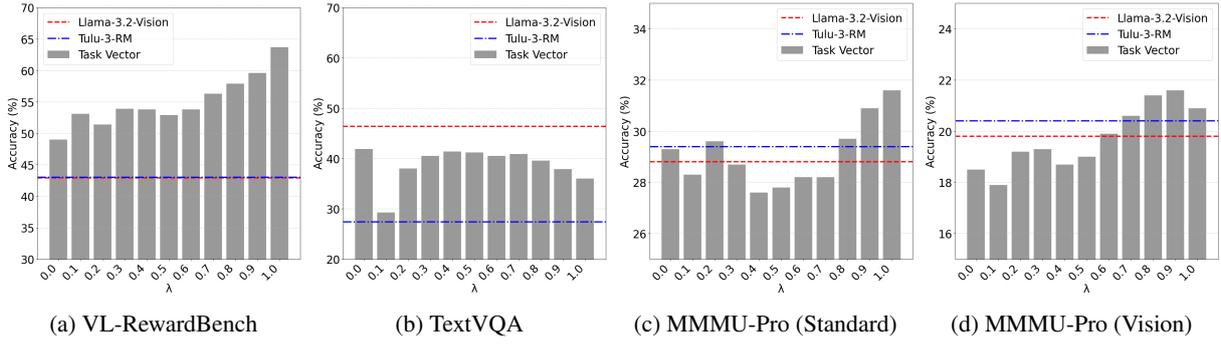


Figure 9: Full results of merging Llama-3.2-Vision and Tulu-3-RM (Task Vec.)

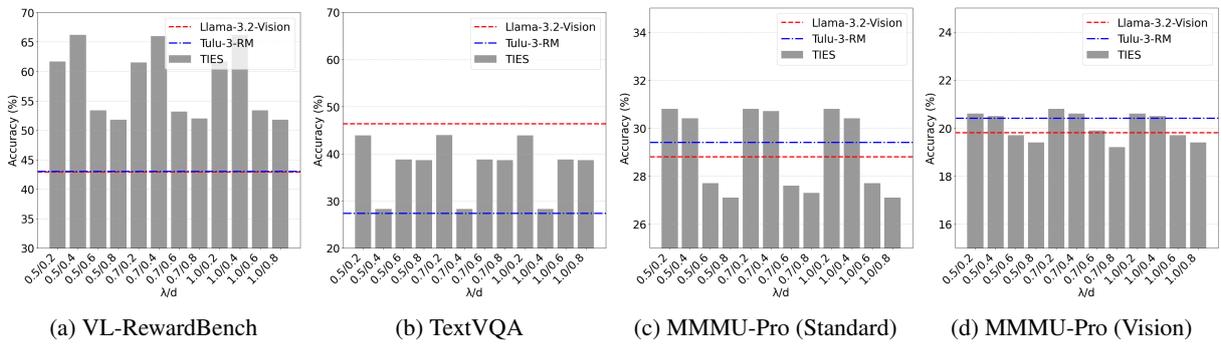


Figure 10: Full results of merging Llama-3.2-Vision and Tulu-3-RM (TIES)

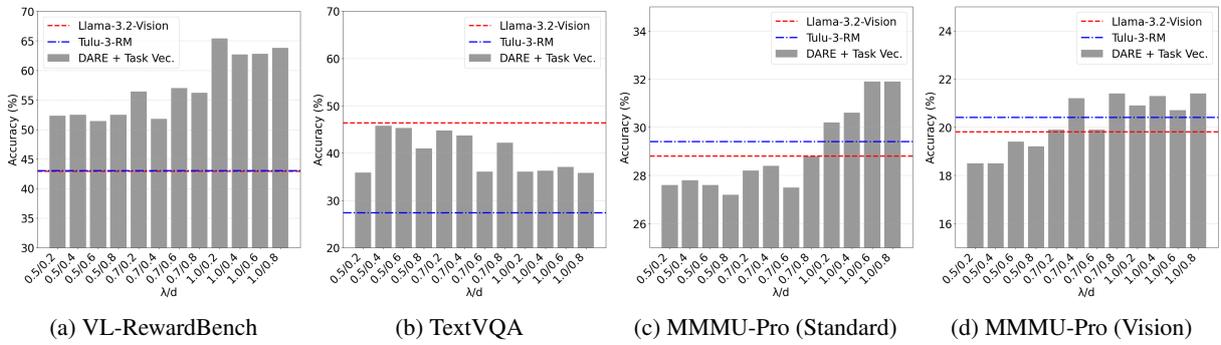


Figure 11: Full results of merging Llama-3.2-Vision and Tulu-3-RM (DARE + Task Vec.)

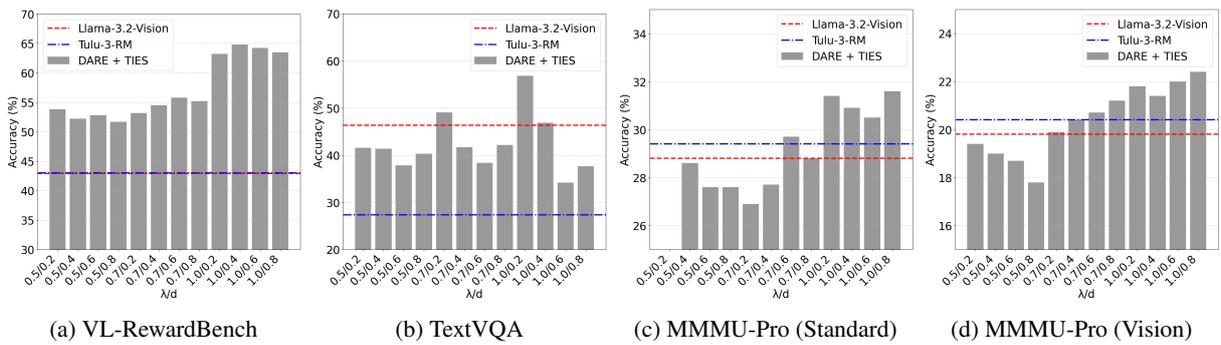


Figure 12: Full results of merging Llama-3.2-Vision and Tulu-3-RM (DARE + TIES)

$\lambda$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Overall Acc.	49.8	52.3	50.3	<b>52.5</b>	52.0	49.0	47.3	46.5	46.5	50.3	47.0

Table 12: Linear merging using Tulu-2.5-RM as the text-based RM, evaluated on sampled RLAIIF-V.

$\lambda$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Overall Acc.	<b>55.3</b>	50.0	53.3	54.5	53.5	49.3	52.8	54.0	53.8	54.8	<b>55.3*</b>

Table 13: Task Vec. merging using Tulu-2.5-RM as the text-based RM, evaluated on sampled RLAIIF-V.

$\lambda$	1.0				0.7				0.5			
$d$	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	53.5	<b>53.8*</b>	52.3	50.0	53.5	<b>53.8</b>	52.3	50.3	53.5	<b>53.8</b>	52.3	50.0

Table 14: TIES merging using Tulu-2.5-RM as the text-based RM, evaluated on sampled RLAIIF-V.

$\lambda$	1.0				0.7				0.5			
$d$	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	55.3	<b>56.5</b>	54.5	55.3	54.5	54.0	53.5	55.8	49.0	49.3	51.8	54.8

Table 15: DARE + Task Vec. merging using Tulu-2.5-RM as the text-based RM, evaluated on sampled RLAIIF-V.

$\lambda$	1.0				0.7				0.5			
$d$	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	55.5	<b>56.0*</b>	<b>56.0</b>	55.5	53.3	54.3	53.8	52.3	51.5	49.8	51.5	51.8

Table 16: DARE + TIES merging using Tulu-2.5-RM as the text-based RM, evaluated on sampled RLAIIF-V.

$\lambda$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Overall Acc.	51.5	46.8	50.3	49.3	<b>52.0</b>	50.8	49.3	47.3	49.5	49.3	51.3

Table 17: Linear merging using Tulu-3-RM as the text-based RM, evaluated on sampled RLAIIF-V.

$\lambda$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Overall Acc.	49.3	53.5	49.8	49.8	51.0	51.0	53.8	53.0	53.0	50.3	<b>55.3</b>

Table 18: Task Vec. merging using Tulu-3-RM as the text-based RM, evaluated on sampled RLAIIF-V.

$\lambda$	1.0				0.7				0.5			
$d$	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	53.5	53.3	54.0	51.0	53.8	<b>54.3</b>	<b>54.3*</b>	51.5	53.5	53.3	54.0	51.0

Table 19: TIES merging using Tulu-3-RM as the text-based RM, evaluated on sampled RLAIIF-V.

$\lambda$	1.0				0.7				0.5			
$d$	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	54.8	55.8	55.3	<b>58.0</b>	53.8	53.8	52.3	50.3	50.0	50.3	51.0	51.5

Table 20: DARE + Task Vec. merging using Tulu-3-RM as the text-based RM, evaluated on sampled RLAIIF-V.

$\lambda$	1.0				0.7				0.5			
$d$	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2	0.8	0.6	0.4	0.2
Overall Acc.	55.8	55.8	56.0	<b>56.8</b>	52.8	52.5	52.5	52.3	55.3	53.8	48.0	54.5

Table 21: DARE + TIES merging using Tulu-3-RM as the text-based RM, evaluated on sampled RLAIIF-V.