
Model Merging Does Not Require Uniform LoRA Ranks

Xie Zirong
Amazon
zirongx@amazon.co.jp

Zainab Afolabi
Amazon Web Services
zafolabi@amazon.com

Nikita Kozodoi
Amazon Web Services
kozodoi@amazon.com

Jack Butler
Amazon Web Services
jackbtlr@amazon.com

Abstract

Model merging combines separately fine-tuned experts into a single model without retraining. Standard practice trains every expert at the same LoRA rank, which couples the rank choice across experts even though each domain has its own best rank. We show this coupling is a property of the merge method, not of merging. We train 36 experts on three domains (mathematics, instruction following, multilingual) at four ranks and three model scales, then merge with six methods at uniform ranks and at each domain’s own best rank. Giving experts different ranks costs averaging and utility-based selection nothing, and gains them up to 1.8 pp of multi-domain accuracy, while the conflict-resolution methods are inconsistent and can lose up to 4 pp. The split has a simple cause. Merging keeps only a fixed share of the pooled spectral components, and methods that choose them by size discard many that would have reduced loss. They also tend to discard the same expert: the instruction adapter has the smallest singular values, so it is dropped almost entirely and IFEval falls by 16 to 25 pp. Choosing components by their effect on loss instead keeps all three experts, and leaves accuracy nearly flat across an $8\times$ rank range (0.7 pp, against 6.7 pp for TIES at 4B scale). Rank can therefore be chosen per expert whenever the merge operator does not select by magnitude.

1 Introduction

Post-training on specialized domains produces experts that excel individually but cannot be deployed together without multiplying inference cost. Model merging is the post-training step that resolves this, combining the experts into one model that handles all tasks. The technique has matured rapidly: from model soups [1] to TIES [2], DARE [3] and KnOTS [4], each method improves on simple averaging, and toolkits like MergeKit [5] make experimentation accessible. Yet across this line of work the LoRA rank is fixed uniformly across experts, and merging adapters of differing rank is largely unexplored.

This convention has a cost. Rank sets adapter capacity, and how much is needed varies with the task, since domains differ in how far they shift the base model and in how much training data they have. A shared rank therefore forces at least one expert away from the rank that suits it. Rank is fixed during fine-tuning, yet the constraint on it comes from the later merging step, so the two stages cannot be designed independently. We measure the per-domain optima in Section 3 and then ask whether merging really requires this coupling at all.

The answer depends on the merge method, and the reason is structural. When experts share rank r , the truncation budget $R = \max_i r_i$ admits a fixed fraction of the pooled spectrum however large r is.

Diverse ranks relax that budget, but magnitude-based selection cannot use the extra room, because it keeps the largest singular values whether or not they reduce loss. We formalize a first-order utility criterion and find that most of the components it rejects would have reduced loss, so the operator discards useful directions by construction rather than by bad tuning.

We make two contributions. First, we show that giving experts different ranks costs averaging and utility-based selection nothing and gains them up to 1.8 pp, while conflict-resolution methods such as TIES can lose up to 4 pp with inconsistent performance, and we trace that asymmetry to how each operator picks spectral components. Second, we identify what the failure destroys rather than only how much: magnitude truncation pushes a single expert out of the kept subspace, and the loss lands almost entirely on instruction following. Together these say that rank coordination across experts is a constraint of the merge method, so changing the method removes it.

2 Related Work

Model merging began with simple averaging [1] and task arithmetic [6], which scales and adds task vectors. These work well when expert updates are compatible and degrade when experts learn conflicting directions, which later methods address by resolving the conflict explicitly: TIES [2] by trimming and sign voting, DARE [3] by random dropping, and KnOTS [4] in spectral space. A separate line weights parameters by importance, using curvature [7], feature covariance [8], or learned coefficients [9]. All of them select by the size of a weight or singular value.

Our criterion follows Optimal Brain Damage [10] and the pruning results that succeeded it, where movement pruning [11] showed magnitude to be a particularly poor saliency signal in the fine-tuning regime we work in. Applying that idea to merging, Park et al. [12] score task vectors with connectivity-based saliency and also decompose LoRAs rank-wise, and Yang et al. [13] vary sparsity per singular vector by importance instead of truncating by singular value. We differ in evaluating utility with a first-order gradient taken at the merged model rather than at individual experts, which is valid because the merged model sits at no single task’s minimum, and in using the criterion to ask whether experts must share a rank at all.

Rank heterogeneity has been studied in adjacent settings, though for different reasons. Federated methods aggregate adapters of unequal rank because client hardware forces it [14], and adaptive methods allocate rank across layers within one adapter [15]. Neither treats rank as a per-expert choice interacting with the merge operator. Closest to us, He et al. [16] also identify the $R = \max_i r_i$ bottleneck and respond by compressing before merging, where we keep the budget and change which components fill it. LoRA [17] is agnostic to rank, but merging is not. MergeBench [18] benchmarks merging methods and our setup follows its protocol, while concurrent work [19] finds expert training duration matters too, so both choices should be made jointly.

3 Methods

We merge with six operators spanning the two selection strategies at issue. Simple averaging and greedy soup [1] combine weights without selecting components, the latter adding experts only when validation accuracy improves. TIES [2] trims low-magnitude entries and resolves sign conflicts, KnOTS [4] truncates the SVD of the stacked update, and plain SVD truncation keeps the top- R singular directions. The last three all rank components by singular value, so they are magnitude-based.

Against these we use a gradient-guided selection method (GGSM), which keeps the same truncation budget but changes the criterion. Stacking the expert updates and taking the SVD gives components (σ_j, u_j, v_j) . We score each by projecting the gradient G of the merged model onto it,

$$\text{utility}_j = -\sigma_j (u_j^\top G v_j), \quad (1)$$

and keep the R components of highest utility rather than largest σ_j . Positive utility means the component reduces loss to first order. Obtaining G takes one backward pass per domain over a small batch of training data, so unlike the data-free baselines GGSM needs data access, at negligible cost.

We use the Qwen3.5 dense checkpoints at three scales, 0.8B, 2B and 4B, following the Qwen3 training recipe [20]. All experts are fine-tuned with QLoRA [21] on a 4-bit quantized base, targeting all attention and MLP projection modules. We train with learning rate 2×10^{-4} , 100-step linear warmup,

Table 1: Single-task optimal LoRA rank per domain and model. Optima differ across domains and shift with scale, so any uniform choice is suboptimal for at least one domain.

Model	Mathematics	Instruction	Multilingual
Qwen3.5-0.8B	8, 16, 32	16	4
Qwen3.5-2B	32	8	4
Qwen3.5-4B	4	32	8

effective batch size 8, and sequence length 2048, selecting checkpoints by minimum validation loss. Rank is the only factor we vary: each domain is trained at ranks 4, 8, 16, and 32, giving 36 experts. Merges use a single seed, so we treat small score differences as ties throughout.

Following MergeBench [18], we adopt three task domains. Mathematics uses DART-Math [22] for training and GSM8K [23] for evaluation. Instruction following uses TULU-3 [24] for training and IFEval [25] for evaluation. Multilingual uses Aya [26] for training and reports the mean of MMLU [27], ARC [28], and MMLU-zh for evaluation. Each domain has its own optimal rank for single-task performance, and these optima differ across domains and scales (Table 1), so any uniform choice is suboptimal for at least one domain.

4 Results

4.1 Uniform ranks minimize what survives truncation

Uniform ranks put the merge operator in the worst position it can occupy, and raising the rank does not help. With T experts at a shared rank r , the pool holds $\approx Tr$ components while the budget admits $R = r$, so a fixed $1/T$ survives however much capacity each expert was given. Table 2 shows this pinned near one-third across an $8\times$ range of ranks. Diverse ranks break the pattern, because the budget follows the largest expert while the pool follows the sum: assigning $r = 4$ to math, 32 to instruction and 8 to multilingual leaves ≈ 44 components with $R = 32$, so 73% survive. A shared rank costs no capacity inside any expert, only the share of the combined update that survives.

Equation 1 lets us ask whether the discarded components were worth keeping, and by that measure the budget is not merely tight, it is spent on the wrong components. Table 3 shows that 65 to 84% of pooled components carry positive utility while the budget admits 33% under uniform ranks and at most 62% under diverse ones, so useful directions must be dropped whatever the hyperparameters, and the waste grows with rank: the count of high-utility components magnitude keeps out rises from about 1 to nearly 10 per matrix between $r = 4$ and $r = 32$, tracking the decline of TIES while utility-based selection stays flat (Figure 2). Utility selection retains less spectral energy ($E_{\text{util}} < E_{\sigma}$) and still merges better, so what matters is which components survive, not how much energy they carry.

Table 2: Spectral structure of the merged update (0.8B, 20–64 modules, two independent samples agreeing to ~ 2 decimals). $R = \max_i r_i$ is the rank budget (components kept by a rank- R merge); $\text{KEPT} = R/\text{pool}$ and E@R is the Frobenius energy retained by the kept set. Uniform assignments sit at $1/3$ at every rank, while diverse assignments retain substantially more.

Regime	Ranks (m/i/l)	R	pool	kept	E@R
uniform	4/4/4	4	12.3	0.326	0.794
uniform	8/8/8	8	24.1	0.332	0.827
uniform	16/16/16	16	48.2	0.332	0.869
uniform	32/32/32	32	96.2	0.332	0.876
diverse	8/16/4	16	28.4	0.565	0.898
diverse	4/16/32	32	52.4	0.611	0.891
diverse	4/32/8	32	44.2	0.725	0.966

Table 3: Utility versus magnitude selection at identical budget R (0.8B, base-point gradients, 96 target matrices). FRAC. POS. exceeds KEPT in every configuration: the budget must reject loss-reducing directions. CLIPPED is the mean count per matrix that utility keeps and magnitude discards.

Regime	Ranks	kept	overlap	clipped	frac. pos.	E_σ	E_{util}
uniform	4/4/4	0.333	0.747	1.01	0.795	0.797	0.694
uniform	8/8/8	0.333	0.714	2.29	0.755	0.827	0.686
uniform	16/16/16	0.333	0.669	5.30	0.720	0.866	0.665
uniform	32/32/32	0.333	0.694	9.78	0.652	0.876	0.701
diverse	8/16/4	0.571	0.744	4.09	0.843	0.896	0.783
diverse	4/16/32	0.615	0.687	10.02	0.780	0.892	0.718

Table 4: Method comparison across scales, best diverse configuration per method, where $\Delta = (\text{best diverse}) - (\text{best uniform})$. GGSM, SVD and averaging hold within noise, TIES and KnOTS perform inconsistently across scales while greedy soup degrades. IFEval drives the spread.

Scale	Method	Math	IFEval	Multi	Avg	Δ
0.8B	GGSM	40.9	55.1	50.7	48.9	+1.4
	SVD	42.4	48.6	51.2	47.4	+0.3
	Simple avg	44.9	49.7	50.9	48.5	+1.8
	Greedy soup	45.3	39.9	48.0	44.4	−4.1
	TIES	37.6	39.7	47.9	41.7	−0.5
	KnOTS	37.9	34.2	47.6	39.9	−2.1
2B	GGSM	61.5	67.5	58.2	62.4	+0.7
	SVD	64.1	64.5	58.3	62.3	+0.2
	Simple avg	62.5	63.4	58.3	61.4	+0.3
	Greedy soup	64.5	61.2	55.1	60.3	−1.0
	TIES	62.8	54.3	55.9	57.7	+0.4
	KnOTS	62.9	51.9	52.8	55.9	+2.3
4B	GGSM	82.5	82.3	70.4	78.4	+0.4
	SVD	84.2	78.7	70.4	77.8	−0.3
	Simple avg	83.9	79.3	70.2	77.8	−0.1
	Greedy soup	83.8	78.9	70.4	77.7	0.0
	TIES	83.8	61.2	68.2	71.1	−4.0
	KnOTS	82.6	54.0	68.5	68.4	−1.1

4.2 Magnitude methods lose more on diverse ranks

If the structural argument holds, methods should divide by how they choose components rather than by how sophisticated they are, and Table 4 shows exactly that. Moving each method from its best uniform assignment to its best diverse one separates the six operators into two groups. For GGSM, SVD and Simple averaging every delta lies between -0.3 and $+1.8$ pp, inside our interval, and the direction is not random: GGSM improves at all three scales by 0.4 to 1.4 pp, and the largest single gain is 1.8 pp for Simple averaging at 0.8B. Diverse ranks are free for these three, with a hint of a gain our sample sizes cannot resolve. Greedy soup loses ground at 0.8B and 2B and is flat at 4B, while TIES and KnOTS are inconsistent; their best diverse configurations beat uniform at 2B but trail at 0.8B and 4B.

Rank sensitivity tells the same story from a different angle. Across the uniform-rank sweep at 4B, GGSM spans 0.7 pp while TIES spans 6.7, about $9\times$ more (Figure 1, left). A method that ignores utility does not merely mishandle diverse ranks, it is unstable under the choice of rank in general. The ordering (GGSM < SVD < averaging < KnOTS < TIES) tracks reliance on magnitude. Appendix B gives the full sweep, and Appendix C shows that at 4B, every diverse configuration costs TIES 4 to 9 pp, at 0.8B under 1 pp, and only at 2B does the best diverse setup edge ahead.

Which of these differences we can trust depends on a resolution floor, which we derive in Appendix A as roughly 1.4 pp for the paired comparisons we report. Three results clear it: the TIES penalty of -4

Table 5: Per-domain scores at uniform rank 4 at 0.8B, 2B and 4B. IFEval collapses for TIES and KnOTS at every scale.

Scale	Method	Math	IFEval	Multi	Avg
0.8B	GGSM (utility)	38.6	55.5	48.5	47.5
	SVD (magnitude)	39.7	47.7	50.8	46.1
	Simple averaging	42.7	47.1	50.3	46.7
	Greedy soup	44.7	50.8	50.0	48.5
	TIES	39.3	39.2	48.1	42.2
	KnOTS	41.9	36.0	48.2	42.1
2B	GGSM (utility)	58.6	67.1	56.5	60.7
	SVD (magnitude)	63.8	63.8	58.7	62.1
	Simple averaging	62.2	63.4	57.3	60.9
	Greedy soup	64.6	61.6	57.6	61.3
	TIES	60.7	50.8	58.1	56.5
	KnOTS	57.1	42.1	55.7	51.6
4B	GGSM (utility)	81.7	82.3	70.0	78.0
	SVD (magnitude)	83.7	80.8	69.7	78.1
	Simple averaging	84.1	78.7	69.9	77.6
	Greedy soup	84.0	79.1	70.0	77.7
	TIES	84.2	66.0	68.7	73.0
	KnOTS	80.9	56.6	67.7	68.4

pp, the sensitivity ranges of 6.7 against 0.7 pp, and the IFEval gaps of 16 to 25 pp. The gaps among the four leading methods do not, so we treat those as ties.

4.3 Instruction following is suppressed

An average score hides the shape of the damage, which is concentrated rather than spread. Attributing the retained top- R subspace to individual experts through $\langle \Delta W_d, C_j \rangle$ over kept components j shows it is badly unbalanced: in 6 of 7 configurations one expert holds at least 0.64 of the retained energy while another is driven near zero, reaching 0.93/0.01/0.06 at uniform rank 16 (Figure 1, right). Merging is meant to combine experts, but under magnitude truncation it can quietly delete one, a capability loss introduced by the merge rather than absent beforehand.

Table 4 compares each method’s best diverse-rank configuration against the single-task experts of Table 10. On the Math task, every method exceeds its expert at all three scales, combining experts yields consistent positive transfer for Math capability. On Multilingual task, GGSM and SVD exceed

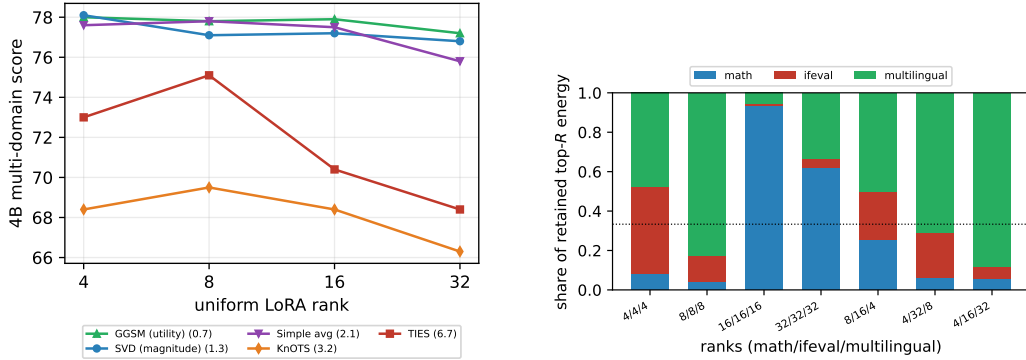


Figure 1: **Left:** sensitivity to uniform rank at 4B, where utility selection spans 0.7 pp against 6.7 pp for TIES. **Right:** share of the retained top- R subspace per expert, where magnitude truncation often drops one almost entirely (e.g. 16/16/16 $\rightarrow$ 0.93/0.01/0.06).

the single-task experts at every scale while the remaining methods, TIES and KnOTS in particular, fall below. The Instruction following task is the discriminating axis: GGSM is the only method that surpasses the single-task IFEval expert at every scale (0.8B: 55.1 vs. 54.2; 2B: 67.5 vs. 67.3; 4B: 82.3 vs. 81.7), whereas all other methods trail it and TIES and KnOTS collapse (e.g. 2B: 54.3 and 51.9). GGSM is thus the only method to improve on the single-task expert across all three domains simultaneously, which is why it attains the highest average at every scale (48.9, 62.4, 78.4) and stands as the strongest method overall.

5 Conclusion

Established practice in model merging trains every expert at the same LoRA rank. Our results suggest this convention reflects a property of the merge operator rather than a requirement of merging. It comes specifically from magnitude-based selection. Uniform ranks pin the truncation budget to a fixed fraction of the pooled spectrum, and magnitude selection cannot use the room that diverse ranks open up. With averaging or utility-based selection, diverse ranks cost nothing and we measure gains of up to 1.8 pp, so rank becomes a per-expert decision that needs no coordination. Evaluated with TIES the same change costs 4 pp at 4B, KnOTS is inconsistent and greedy soup loses ground or ties. For these magnitude-based operators, the safer choice is matching ranks, but that conclusion holds only for them and does not generalize to merging. This matters most when experts come from separate teams or arrive at different times. The practical recommendation is to train each expert at the rank that suits its domain, and merge with an operator that selects by effect on loss rather than size.

Our study has limitations we plan to address in future work. The small gains from diverse ranks sit below our 1.4 pp resolution floor, so confirming their significance needs more seeds and larger evaluations. We also cover three domains, focus on LoRA fine-tuning only, and use one model family.

References

- [1] Mitchell Wortsman, Gabriel Ilharco, Samir Yitzhak Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S. Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and Ludwig Schmidt. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022.
- [2] Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- [3] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *arXiv preprint arXiv:2311.03099*, 2024.
- [4] George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. KnOTS: Model merging with SVD to tie the Knots. *arXiv preprint arXiv:2410.19735*, 2024.
- [5] Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meade, Hailey Zeng, and Elie Fung. Arcee’s MergeKit: A toolkit for merging large language models. *arXiv preprint arXiv:2403.13257*, 2024.
- [6] Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023.
- [7] Michael S. Matena and Colin Raffel. Merging models with Fisher-weighted averaging. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- [8] Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *International Conference on Learning Representations (ICLR)*, 2023.

- [9] Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. *International Conference on Learning Representations (ICLR)*, 2024.
- [10] Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1989.
- [11] Victor Sanh, Thomas Wolf, and Alexander M. Rush. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [12] Jungin Park, Jiyoung Lee, and Kwanghoon Sohn. Saliency-aware model merging. *arXiv preprint arXiv:2606.00511*, 2026.
- [13] Yan Yang, Yixia Li, Hongru Wang, Xuetao Wei, Jianqiao Yu, Yun Chen, and Guanhua Chen. ImPart: Importance-aware delta-sparsification for improved model compression and merging in LLMs. *arXiv preprint arXiv:2504.13237*, 2025.
- [14] Muhammad Waseem, Nurbek Tastan, Andrej Jovanovic, Nicholas D. Lane, Nils Lukas, Karthik Nandakumar, and Samuel Horvath. PreLort: Prefix-nested LoRA for federated fine-tuning under rank heterogeneity. *arXiv preprint arXiv:2606.15963*, 2026.
- [15] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. *International Conference on Learning Representations (ICLR)*, 2023.
- [16] Zhengbao He, Ruiqi Ding, Zhehao Huang, Ruikai Yang, Tao Li, and Xiaolin Huang. Compress then merge: From multiple LoRAs into one low-rank adapter. *arXiv preprint arXiv:2606.03723*, 2026.
- [17] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.
- [18] Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. MergeBench: A benchmark for merging domain-specialized LLMs. *arXiv preprint arXiv:2505.10833*, 2025.
- [19] Nikita Kozodoi, Zainab Afolabi, and Jack Butler. Are we merging the right models? impact of expert training duration on model merging for LLMs. *arXiv preprint arXiv:2607.11997*, 2026.
- [20] An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- [21] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient fine-tuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [22] Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. DART-math: Difficulty-aware rejection tuning for mathematical problem-solving. *arXiv preprint arXiv:2407.13690*, 2024.
- [23] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [24] Nathan Lambert et al. TULU 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.
- [25] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.
- [26] Shivalika Singh et al. Aya dataset: An open-access collection for multilingual instruction tuning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

- [27] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *International Conference on Learning Representations (ICLR)*, 2021.
- [28] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. In *arXiv preprint arXiv:1803.05457*, 2018.

A Resolution Floor for Score Comparisons

Per-benchmark standard errors propagate as $SE_{\text{domain}} = \sqrt{\sum_i se_i^2}/k$ over the k benchmarks in a domain, and $SE_{\text{avg}} = \sqrt{\sum_d SE_d^2}/3$ over the three domains. This gives $SE_{\text{avg}} \approx 0.93$ to 1.04 pp, so a 95% interval spans roughly ± 1.9 pp on any single multi-domain score. The diverse-versus-uniform comparisons are paired, since both configurations are evaluated on identical test items, and a paired bootstrap over those shared items roughly halves the interval. We therefore treat 1.4 pp as the smallest difference we can resolve, and read anything below it as a tie.

B How Much the Merge Method Depends on Rank

Table 6 gives the full uniform-rank sweep at 0.8B, 2B and 4B Qwen3.5 models. At every scale GGSM has the most stable performance across 8 $\times$ range while TIES drops by 4.3 at 0.8B scale and by 6.7 at 4B scale. The ordering by range (GGSM < SVD < averaging < KnOTS < TIES) matches reliance on magnitude-based selection. Figure 2 shows the mechanism behind that ordering: the

Table 6: Uniform-rank sweep across scales (same-rank merges). GGSM is the most rank-stable at every scale; KnOTS and TIES are the most sensitive.

Scale	Method	r4	r8	r16	r32	Range
0.8B	GGSM (utility)	47.5	47.5	46.9	46.3	1.2
	SVD (magnitude)	46.1	46.8	47.1	45.9	1.2
	Simple averaging	46.7	46.5	46.2	44.9	1.8
	KnOTS	42.1	37.7	35.0	34.6	7.5
	TIES	42.2	41.0	39.7	37.9	4.3
2B	GGSM (utility)	60.7	61.2	61.7	60.6	1.1
	SVD (magnitude)	62.1	61.0	61.9	59.8	2.3
	Simple averaging	60.9	61.1	60.6	58.5	2.6
	KnOTS	51.6	53.6	52.2	51.0	2.6
	TIES	56.5	57.3	55.5	53.6	3.7
4B	GGSM (utility)	78.0	77.8	77.9	77.2	0.7
	SVD (magnitude)	78.1	77.1	77.2	76.8	1.3
	Simple averaging	77.6	77.8	77.5	75.8	2.1
	KnOTS	68.4	69.5	68.4	66.3	3.2
	TIES	73.0	75.1	70.4	68.4	6.7

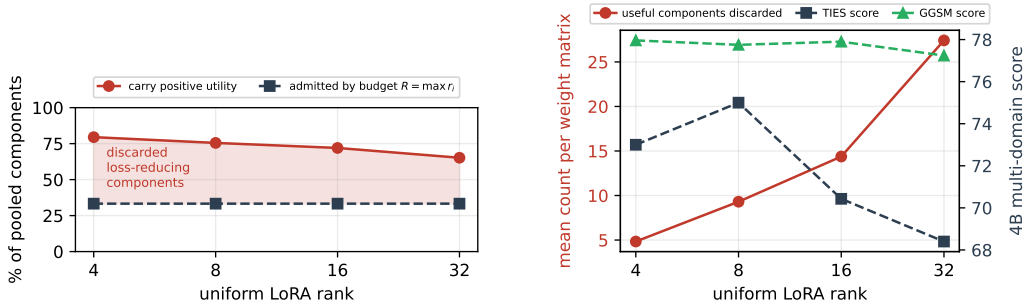


Figure 2: **Left:** the fraction of pooled components carrying positive first-order utility far exceeds the fraction the budget admits, which is fixed at $\frac{1}{T}$ for T tasks under any uniform assignment. **Right:** the number of high-utility components that magnitude selection discards grows roughly tenfold from $r = 4$ to $r = 32$ (left axis), tracking the decline of TIES while utility-based selection stays flat (right axis).

Table 7: TIES on diverse ranks across scales. At every size the best diverse configuration underperforms the best uniform-rank baseline (0.8B: 42.2 at r4; 2B: 57.3 at r8; 4B: 75.1 at r8) except at 2B where the diverse-rank setup is marginally higher.

Scale	Ranks (m/i/l)	Math	IFEval	Multi	Avg
0.8B	8/32/4 (optimal)	37.6	39.7	47.9	41.7
	4/8/16	39.3	39.4	46.2	41.6
	8/16/4	39.0	38.4	46.9	41.4
	4/32/8	35.6	38.4	50.1	41.4
	32/8/4	39.9	35.3	48.8	41.3
2B	16/8/4 (optimal)	62.8	54.3	55.9	57.7
	16/4/8	61.9	54.3	55.8	57.3
	16/32/8	60.9	52.3	56.7	56.6
	32/16/4	61.7	53.4	54.5	56.5
	32/8/4	61.5	53.4	54.3	56.4
4B	4/32/8 (optimal)	83.8	61.2	68.2	71.1
	4/16/8	82.9	56.0	68.1	69.0
	4/8/16	81.4	57.5	68.1	69.0
	4/16/32	80.7	51.6	67.7	66.7
	4/8/32	81.0	50.1	67.4	66.1

share of pooled components carrying positive utility far exceeds what the budget admits, and the number of useful components discarded grows with rank.

C The Cost of Diverse Ranks

Table 7 reports TIES across diverse-rank configurations at all three model scales, from which two observations follow. First, the natural heuristic of assigning each domain its individually optimal rank (0.8B: 8/16/4; 2B: 32/8/4; 4B: 4/32/8) yields no improvement: it underperforms the best uniform-rank baseline at every scale (0.8B: 41.4 vs. 42.2, -0.8 ; 2B: 56.4 vs. 57.3, -0.9 ; 4B: 71.1 vs. 75.1, -4.0). Second, an exhaustive search over diverse assignments does recover a configuration that matches or marginally exceeds the uniform baseline at the two smaller scales (best diverse at 0.8B, 41.7, -0.5 ; at 2B, 57.7, $+0.4$), whereas at 4B even the best diverse configuration falls short by 4.0 pp. For TIES, therefore, the benefit of rank diversity is marginal and attainable only through search; the per-domain-optimal heuristic is consistently inferior to a single shared rank, and this deficit grows with scale. Where degradation occurs, it is attributable to IFEval, which deteriorates sharply under diverse-rank TIES.

Table 8 extends this comparison to all methods evaluated at the per-domain-optimal ranks across scales. The utility-, magnitude-, and averaging-based methods (GGSM, SVD, simple averaging, and Greedy soup) lie within approximately 1pp in average score at 2B (60.3-61.3) and 4B (77.7-78.4), inside our interval, so we treat them as tied, while GGSM establishes a wider margin of roughly 2pp at 0.8B (48.5 vs. 46.6). TIES and KnOTS trail this group by 4 to 10pp at every scale, again owing to the collapse of IFEval: GGSM preserves IFEval performance at 56.7, 66.7, and 82.3 for 0.8B, 2B, and 4B respectively, whereas TIES falls to 38.4, 53.4, and 61.2. This pattern is thus consistent across scales rather than specific to any single model size.

D Where the Loss Lands

Table 10 lists the single-task baselines the merged models are measured against. Merged models can exceed a single-task expert on its own domain through positive transfer—for instance, stronger instruction following may improve how math answers are formatted and thereby extracted. Figure 3 breaks the uniform-rank sweep down by domain and shows that the method spread is dominated by instruction following (at 4B, a 33-point range across methods versus 3.3 for math and 3.9 for multilingual), while Figure 4 compares IFEval against the single-task baseline and sweeps TIES across diverse configurations.

Table 8: All methods on the per-domain optimal ranks at each scale (0.8B: 8/16/4; 2B: 32/8/4; 4B: 4/32/8). GGSM leads on Avg at every scale; TIES and KnOTS trail because IFEval collapses.

Scale	Method	Math	IFEval	Multi	Avg
0.8B	GGSM (utility)	39.0	56.7	49.8	48.5
	Simple averaging	41.6	48.2	49.8	46.5
	SVD (magnitude)	41.3	47.3	51.2	46.6
	Greedy soup	45.3	39.9	48.0	44.4
	KnOTS	37.9	34.2	47.6	39.9
	TIES	39.0	38.4	46.9	41.4
2B	GGSM (utility)	61.9	66.7	55.2	61.3
	Simple averaging	63.5	63.6	55.9	61.0
	SVD (magnitude)	64.4	61.0	55.8	60.4
	Greedy soup	64.5	61.2	55.1	60.3
	KnOTS	62.9	51.9	52.8	55.9
	TIES	61.5	53.4	54.3	56.4
4B	GGSM (utility)	82.5	82.3	70.4	78.4
	Simple averaging	83.8	79.3	70.2	77.8
	SVD (magnitude)	84.2	78.7	70.4	77.8
	Greedy soup	83.8	78.9	70.4	77.7
	KnOTS	82.6	54.0	68.5	68.4
	TIES	83.8	61.2	68.2	71.1

Table 9: 2B results: Greedy soup vs GGSM on same-rank and diverse-rank configurations.

Method	Ingredients	Math	IFEval	Multi	Avg
GGSM (same)	16/16/16	61.4	67.1	56.7	61.7
Greedy (same)	m4, i4	64.6	61.6	57.6	61.3
GGSM (diverse)	8/16/4	61.5	67.5	58.2	62.4
Greedy (diverse)	m32, i4, l8	64.5	61.2	55.1	60.3

Table 10: Single-task expert baselines across scales (best rank per domain and its score).

Scale	Domain	Best Rank	Score
0.8B	Math (GSM8K)	r8/r16/r32	34.5
	Instruction (IFEval)	r16	54.2
	Multilingual	r4	48.0
2B	Math (GSM8K)	r32	56.6
	Instruction (IFEval)	r8	67.3
	Multilingual	r4	57.7
4B	Math (GSM8K)	r4	79.7
	Instruction (IFEval)	r32	81.7
	Multilingual	r8	70.3

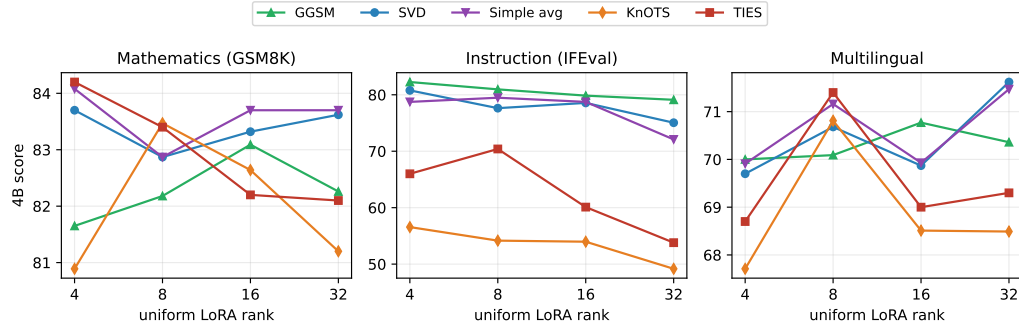


Figure 3: Per-domain scores across uniform ranks at 4B. IFEval (center) shows the starkest method divergence: GGSM and SVD maintain 75.1 to 82.3 while TIES and KnOTS collapse to 49.2 to 70.4. Math (left) and Multilingual (right) vary little across methods.

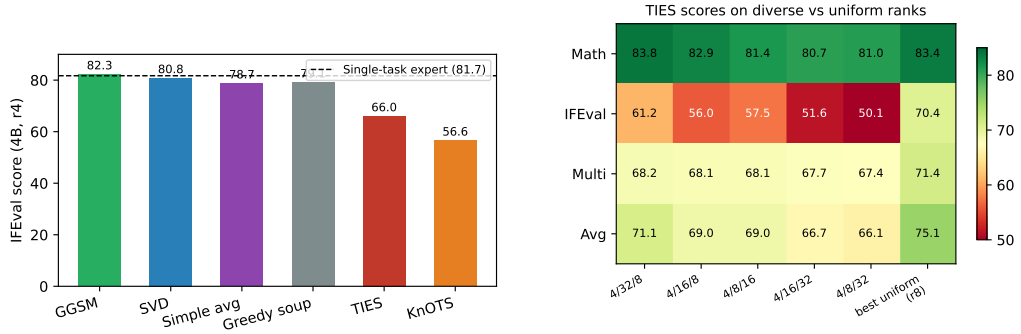


Figure 4: **Left:** IFEval scores at 4B, uniform rank 4. TIES and KnOTS lose 16 to 25 pp relative to the single-task expert baseline (dashed line), while GGSM slightly exceeds it. **Right:** TIES performance across diverse rank configurations vs best uniform (r8). Every diverse configuration underperforms uniform on aggregate, primarily due to IFEval collapse.