MODEL MERGING DESTROYS WHAT YOU PAID TO TEACH

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) are increasingly specialized by fine-tuning copies of a shared base model and then merging them into a single model. While model merging is typically evaluated by how well the merged model preserves downstream benchmark performance, these evaluations need not reveal whether the information acquired during specialization survives the merge. We demonstrate this gap on multilingual and coding specialists: merged models with similar benchmark performance can retain very different amounts of the knowledge acquired during specialization. To study this failure systematically, we introduce a suite of ten factual datasets designed to measure knowledge acquired during fine-tuning. Across these datasets, factual retention drops substantially after merging, worsens as more specialists are combined, and can be highly uneven across specialists. To better preserve this knowledge, we introduce LUMEN (Low-rank Update Merging with residual ENhancement), which jointly learns a coordinate-wise mixture of specialist updates and a low-rank correction for the merged model using only a small subset ($1 - 5\%$) of specialist data. Across merges of two to six specialists, LUMEN improves normalized factual retention by 21.5% on average over the strongest reported baseline at each merge size. At six specialists, LUMEN retains 60.1% pre-merge factual accuracy, compared with 30.5% for the strongest reported baseline. Together, our results establish factual retention as a distinct objective for model merging and show that substantially more of the knowledge acquired during specialization can be preserved at no additional inference cost.

1 INTRODUCTION

Model merging is a deployment strategy that replaces a collection of separately deployed fine-tuned (specialist) models with a single set of weights (Ilharco et al., 2022; Yadav et al., 2023). This is especially useful when one base model has been adapted across many datasets, customers, or domains. Prior work has made merging increasingly practical and has shown that merging can preserve or improve performance on a range of task-based benchmarks (Wortsman et al., 2022; Ilharco et al., 2022; Jin et al., 2022; Yadav et al., 2023; Yu et al., 2023; Yadav et al., 2024; He et al., 2025b). For specialists trained to learn new information, however, success on these evaluations leaves a basic question unanswered: does the merged model still recall what each specialist was taught?

The distinction between factual recall and benchmark performance matters because specialist models can be fine-tuned not only to acquire a capability, but also to learn information absent from the base model. Consider a collection of specialists fine-tuned for different purposes, including one trained to use a range of software frameworks. After merging, a developer may ask the resulting model to implement a framework-specific feature. General programming ability is not enough: the model must also know the relevant syntax, API names, and function signatures. If that knowledge is lost during merging, part of what the coding specialist learned has been lost, even if the merged model continues to perform well on broad coding benchmarks. In such settings, factual retention is not merely another evaluation metric, it captures part of the functionality that merging is intended to preserve.

Capability and acquired knowledge can tell different stories about the same merge. We make this distinction concrete using Aya (Singh et al., 2024) and Magicoder (Wei et al., 2023) from MergeBench (He et al., 2025b), evaluating the same merged models both on the specialists' original task benchmarks and on factual knowledge acquired during fine-tuning (Figure 1 and 2). A merge

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

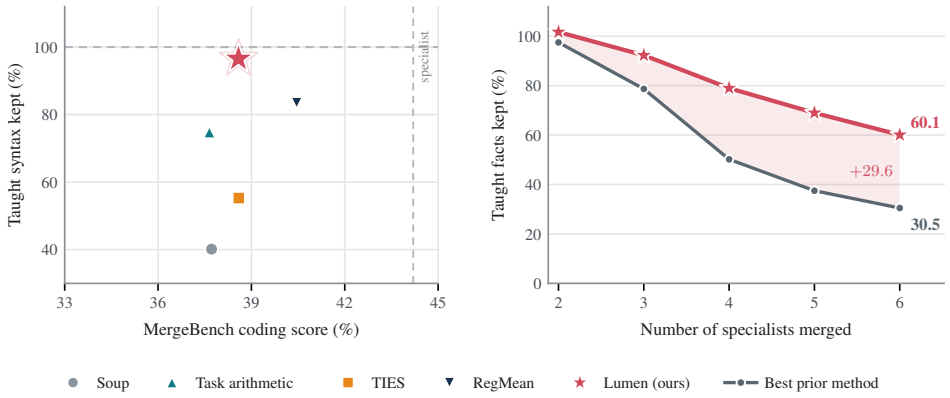


Figure 1: **Merging preserves benchmark performance while erasing specialist facts.** As more specialists are combined, factual retention falls sharply even when general task performance remains high. LUMEN closes much of this gap by learning a parameter-wise mixture of specialist updates together with a small low-rank correction, while still deploying as a single model.

can retain substantial task performance while losing considerably more of the information learned during specialization. Thus, strong performance on a specialist's original benchmark does not by itself show that the specialist's learned knowledge survived the merge. Factual retention captures a complementary aspect of merge quality that standard task evaluation can miss.

To study this missing dimension directly, we introduce a suite of ten factual datasets designed to isolate knowledge acquired during fine-tuning. Nine are built from public sources spanning biomedicine, law, biographical, and bibliographic information, while the tenth contains entirely synthetic facts. We fine-tune specialists to reliably recall these facts and then ask how much of that knowledge survives when the specialists are merged. The answer is sobering: substantial factual loss appears even when only two specialists are combined, becomes more severe as additional specialists are added, and can fall very unevenly across specialists. These results reveal factual retention as a distinct failure mode of model merging and provide a controlled setting for studying it in Section 3.

Existing merge rules primarily decide how to combine the updates learned by different specialists. Because these updates are learned independently, simply reweighting them may not fully account for the model that emerges from their combination. We introduce LUMEN (Low-rank Update Merging with Residual ENhancement), which jointly learns coordinate-wise mixing weights, parameterized through low-rank factors, and a low-rank correction to the merged weights. Using a small ($\approx 1\%$) held-out subset of specialist data, we optimize both and fold them into a single checkpoint with the same architecture and inference cost as a single specialist.

We evaluate LUMEN as increasingly many specialists are merged into a single model. Across merges of two to six specialists, it improves normalized factual retention by 21.5% on average over the strongest reported baseline at each merge size. At six specialists, LUMEN retains 60.1% of the specialists' pre-merge factual accuracy, compared with 30.5% for the strongest reported baseline. Importantly, these gains do not come at the expense of the specialists' original capabilities. On the five-specialist MergeBench evaluation, LUMEN achieves the strongest average benchmark score in our experiments while also preserving substantially more specialist-taught knowledge on the multilingual and coding probes. Together, these results suggest that preserving what a specialist can do and preserving what it learned are complementary objectives for model merging.

2 MODEL MERGING

Model merging seeks to combine multiple fine-tuned models into a single model without retraining on all of their original data. Early approaches average model parameters directly, as in model soups (Wortsman et al., 2022), while Task Arithmetic manipulates the difference between fine-tuned and pretrained parameters (Ilharco et al., 2022). Subsequent work improves this basic operation along

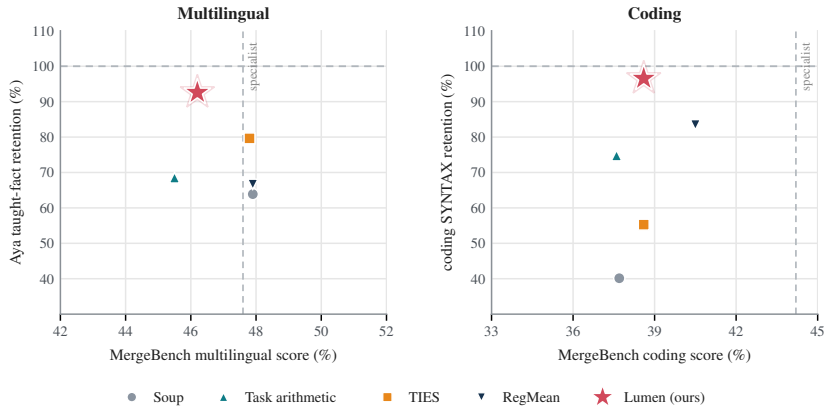


Figure 2: **Similar benchmark performance can mask large differences in retention.** Benchmark performance (x) versus factual retention of specialist-taught content (y) for five merging methods. Benchmark scores span only 2.4 points for multilinguality and 2.9 for coding, whereas retention spans 28.7 and 56.4 points, respectively. LUMEN ($\star$) performs best on both retention probes, preserving 92.6% of the Aya facts and 96.6% of the code syntax. Vertical dashed lines mark specialist benchmark scores; horizontal dashed lines mark the specialist's (normalized) retention.

several dimensions. Task-vector editing methods reduce interference by resolving sign conflicts or sparsifying updates before combining them (Yadav et al., 2023; Yu et al., 2023). Other training-free methods use curvature or activation statistics to weight parameters (Matena & Raffel, 2022; Jin et al., 2022), while spectral methods compress or decorrelate task updates in low-rank subspaces (Gargiulo et al., 2025; Choi et al., 2024; Wu et al., 2026). A separate line of work learns merge coefficients from small amounts of data: AdaMerging optimizes task- or layer-wise coefficients using unlabeled target examples (Yang et al., 2023), LoRA Soups learns layer-wise coefficients for combining adapters (Prabhakar et al., 2024), and aTLAS learns independent coefficients for parameter blocks of each task vector (Zhang et al., 2024). Similarly to aTLAS, our method learns an anisotropic combination of fixed task vectors that can be folded into a single checkpoint. LUMEN differs in the structure and granularity of this combination: low-rank factors generate coordinate-wise gates, while a jointly learned low-rank correction adjusts the merged weights.

3 FACTUAL RETENTION UNDER MODEL MERGING

We first ask whether standard task benchmarks reveal how much of a specialist's acquired knowledge survives merging, and then introduce a controlled testbed for measuring factual retention directly.

3.1 BENCHMARK PERFORMANCE DOES NOT FULLY CAPTURE FACTUAL RETENTION

Standard task benchmarks measure whether a merged model retains broad capabilities, but not necessarily whether it preserves the information acquired by each specialist. We examine this distinction using the multilingual and coding specialists released with MergeBench (He et al., 2025b), which were fine-tuned from Llama-3.2-3B (Grattafiori et al., 2024) on Aya (Singh et al., 2024) and Magicoder (Wei et al., 2023), respectively.

Following MergeBench, we average M-MMLU, M-ARC, and M-HellaSwag (Lai et al., 2023) to obtain a multilingual benchmark score, and HumanEval+ and MBPP+ (Liu et al., 2023) to obtain a coding score. We then evaluate the same five merging methods on direct factual probes derived from the specialists' fine-tuning data: Aya facts for the multilingual specialist and synthetic facts derived from Magicoder's training data for the coding specialist. Retention measures the fraction of the specialist's factual recall that remains after merging.

Figure 2 reveals a sharp mismatch between the two measures. The merges preserve 95.6%–100.6% of the multilingual specialist's benchmark score, yet the four baselines retain only 63.9%–79.6% of

Dataset	Facts queried	# Facts	Base acc.	Freq. base.
UniProt	protein → GO annotation	239,084	0.0	13.7
Caselaw	case → court, year, citation	240,000	0.0	2.8
ClinicalTrials	trial → condition, sponsor	120,064	0.0	1.5
ClinicalTrials-PI	trial → investigator, affiliation	119,844	0.0	0.5
NIH RePORTER	award → institute, organisation	74,852	0.0	6.8
NIH RePORTER-PI	award → investigator, organisation	91,584	0.0	0.8
OpenAlex	paper → first, last author	116,276	0.0	11.2
Open Library	book → author, publisher	112,960	0.0	1.2
Wikidata	person → role, founder, advisor	125,052	0.0	0.7
Synthetic	invented entity → invented value	240,000	0.0	7.7

Table 1: **Factual datasets and pre-fine-tuning baselines.** Each query specifies an entity and relation and asks for the target value. Base acc. is the base model's exact-match accuracy; Freq. base. predicts the most frequent value for each relation without using the entity. Accuracies are percentages.

its acquired facts; LUMEN retains 92.6%. Similarly, the coding merges preserve 85.1–91.6% of benchmark performance, while baseline syntax retention ranges from 40.1% to 83.5%; LUMEN reaches 96.6%. Thus, similar benchmark performance can conceal substantial differences in how much specialist-taught information survives merging.

3.2 A Controlled Testbed for Factual Retention

Factual datasets. We construct ten datasets: nine based on public sources and one synthetic dataset containing invented facts. Each fact is an (entity, relation, value) triple: the entity identifies the item, the relation specifies the requested information, and the value is the target answer. For example, a clinical-trial query may ask for the principal investigator (relation) associated with a trial identifier (entity). Table 1 summarizes the datasets and includes two simple baselines: the base model, and a frequency baseline that predicts the most common value for each relation while ignoring the entity. Under our scoring protocol, the base model scores 0% on every dataset.

Scoring. Training examples use the format `entity > relation > <ans> value`, where `<ans>` is an added delimiter token. At evaluation, we provide the prefix through `<ans>` and greedily generate at most 24 tokens. An answer is correct only if it exactly matches the target after removing leading and trailing whitespace. Let $\mathrm{Acc}(\theta; D_i)$ be the exact-match accuracy of model θ on specialist i's factual query set D_i. We define its *factual retention* as

$$R_i = 100 \cdot \frac{\mathrm{Acc}(\theta_{\mathrm{merge}}; D_i)}{\mathrm{Acc}(\theta_i; D_i)}, \tag{1}$$

where θ_i is the specialist before merging. Thus, $R_i = 100$ means that the merged model matches the specialist's original factual accuracy.

Models and specialists. We use Qwen3.5-4B (Qwen Team, 2026) as the base model. For each dataset, we fine-tune separate copies of this model on two non-overlapping sets of factual queries. We call each resulting model a *specialist*. Specialists are trained for 8 epochs and reach between 88% and 97% accuracy on the facts they were taught. We evaluate both within-dataset and cross-dataset merges. We measure the preservation of learned facts, not generalization to facts unseen during specialist training.

Merge methods. We evaluate Task Arithmetic (TA), TIES, DARE, and RegMean (RM) (Ilharco et al., 2022; Yadav et al., 2023; Yu et al., 2023; Jin et al., 2022). For each merge and merging method, we tune the hyperparameters and report the factual recall on the test set for the evaluated configuration with the highest validation performance. Appendix A.4 summarizes the merge rules and provides additional model settings and analysis.

Merge	parent accuracy A	parent accuracy B	retention (%) TA	retention (%) TIES	retention (%) DARE	retention (%) RM
Within-dataset merges						
NIH RePORTER	96.4	96.4	83.6	**91.7**	83.2	79.3
OpenAlex	94.8	92.4	87.2	**89.7**	84.6	79.7
Wikidata	95.6	94.4	81.9	**86.1**	78.7	66.2
NIH RePORTER-PI	92.8	93.6	77.3	**83.7**	75.1	73.8
ClinicalTrials	96.0	95.2	70.5	**77.4**	67.6	60.9
Open Library	89.6	93.6	71.0	**77.3**	68.8	60.7
ClinicalTrials-PI	89.6	88.4	59.1	**63.6**	52.8	48.0
Caselaw	94.4	95.2	67.3	**72.8**	63.7	60.9
UniProt	92.8	93.2	34.0	31.8	34.0	**46.9**
Synthetic	94.4	92.0	5.6	**6.4**	3.4	5.9
Cross-dataset merges						
UniProt + Caselaw	92.8	94.4	56.0	36.5	47.2	**57.2**
Open Library + OpenAlex	89.6	94.8	**67.7**	63.6	66.8	61.6
OpenAlex + ClinicalTrials-PI	94.8	89.6	54.9	51.0	51.8	**58.0**
OpenAlex + ClinicalTrials	94.8	96.0	66.2	65.4	62.3	**74.3**
Open Library + ClinicalTrials-PI	89.6	89.6	56.9	52.7	49.8	**57.2**
Open Library + ClinicalTrials	89.6	96.0	**70.7**	64.7	67.5	63.4

Table 2: **Factual retention after merging.** Parent accuracy is each specialist recall on its own facts, before merging. Retention (%) is the merged model's accuracy normalised by the specialist's performance. The best factual retention in each row is bold.

3.3 Merging Substantially Reduces Factual Recall

Table 2 shows substantial factual loss across the evaluated merges. Across the six cross-dataset pairs, even the strongest tested method retains only 57.2%–74.3% of the specialists' pre-merge factual performance. Within-dataset merges are generally less damaging, but retention still varies widely across datasets, indicating that the severity of the loss depends strongly on what is being merged.

3.4 Retention Degrades with the Number of Merged Specialists

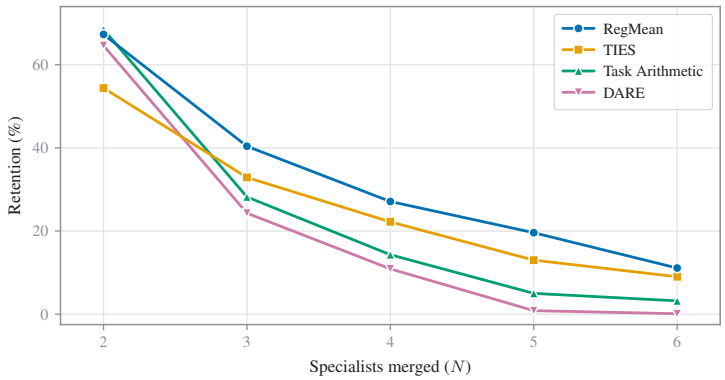


Figure 3: **Factual retention declines as more specialists are merged.** Each curve shows retention for one merging method as specialists are added incrementally to the nested merge sequences. Even the strongest method shown retains only 11.1% at six specialists.

To study how factual retention changes as more specialists are combined, we construct a nested sequence that begins with two specialists and adds one specialist at a time while keeping the previously included specialists fixed. For each merge size and merging method, we tune the hyperparameters and report the best retention in Figure 3.

Retention decreases as additional specialists are introduced. The same trend appears across the evaluated merge rules: retention drops sharply for Task Arithmetic, TIES, DARE, and RegMean as the number of specialists increases. By six specialists, even the strongest method shown in Figure 3 retains only 11.1% of the specialists' pre-merge factual performance. Thus, preserving factual knowledge becomes substantially harder as more specialists are consolidated into a single model.

3.5 Aggregate Retention Hides Specialist Imbalance

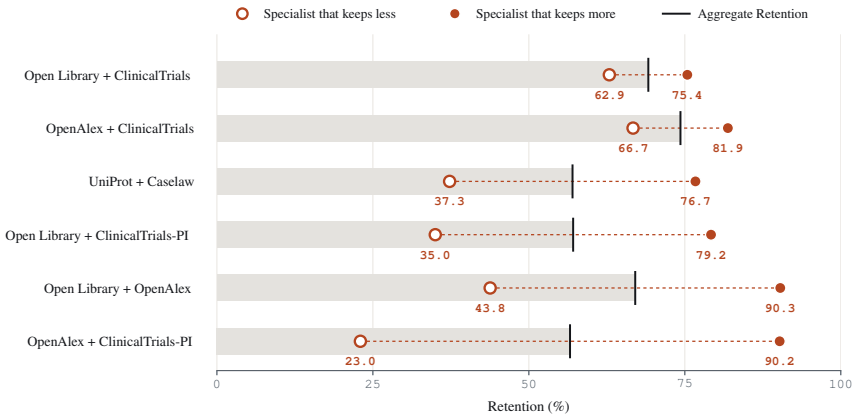


Figure 4: **Merging can preserve one specialist while destroying the other.** For each two-specialist merge, dots show normalized retention of the merged model with respect to each specialist's facts; black markers show the aggregate retention reported for each merge.

Aggregate retention can also conceal large differences in how well the merged model preserves its constituent specialists. We therefore measure retention separately with respect to each specialist's pre-merge factual performance (Figure 4).

Across the six cross-dataset merges, the mean gap between the two specialists' retention is 37.5%. The imbalance can be severe: when OpenAlex and ClinicalTrials-PI are merged, the model retains 90.2% of one specialist's factual performance but only 23.0% of the other's. By contrast, the mean gap is only 3.0% for within-dataset merges. These results show that a single aggregate retention score can hide which specialist's knowledge is being lost.

4 Method

Since existing merging methods can lose substantial specialist-taught knowledge, we propose LUMEN, which combines models more effectively without increasing deployment cost. It separates merging into *mixing*, using a lightweight coordinate-wise gate to combine the specialist updates, and *correction*, adding a learned low-rank adjustment to the merged weights. We train both components end to end on a small ($\approx 1\%$) calibration subset of the specialist data.

For each weight matrix $W_{B,k} \in \mathbb{R}^{m_k \times n_k}$ in the shared base model, let $\tau_{i,k} = W_{i,k} - W_{B,k} \in \mathbb{R}^{m_k \times n_k}$ denote the corresponding update from specialist i. Rather than restricting specialist mixing to task- or layer-level scalar coefficients, as in existing learned merging methods (Yang et al., 2023; Prabhakar et al., 2024), we learn coordinate-wise mixing weights.

We parameterize the gate for each specialist–matrix pair (i, k) with learned low-rank factors $U_{i,k} \in \mathbb{R}^{m_k \times r_g}$ and $V_{i,k} \in \mathbb{R}^{n_k \times r_g}$ and a scalar bias $b_{i,k} \in \mathbb{R}$. These define a logit matrix $L_{i,k} \in \mathbb{R}^{m_k \times n_k}$. A coordinate-wise softmax over specialists then yields the mixing weights $G_{i,k}$:

$$L_{i,k} = U_{i,k} V_{i,k}^\top + b_{i,k}\mathbf{1}, \tag{2}$$

$$(G_{i,k})_{pq} = \frac{\exp((L_{i,k})_{pq})}{\sum_{j=1}^{N} \exp((L_{j,k})_{pq})}. \tag{3}$$

Further, we add a learnable low-rank *correction* $P_k Q_k^\top$ to the mixed updates, where $P_k \in \mathbb{R}^{m_k \times r_c}$ and $Q_k \in \mathbb{R}^{n_k \times r_c}$. The merged weight matrix W_k, with learned scale $\alpha_k \in \mathbb{R}$, is therefore given by

$$\boxed{W_k = W_{B,k} + \underbrace{\alpha_k \sum_{i=1}^{N} G_{i,k} \odot \tau_{i,k}}_{\text{specialist mixing}} + \underbrace{P_k Q_k^\top}_{\text{correction}}} \tag{4}$$

For each gated matrix, the gate has $N[r_g(m_k + n_k) + 1]$ trainable parameters, and the correction has $r_c(m_k + n_k)$ parameters. In our experiments, we apply LUMEN only on every fourth matrix among the non-embedding matrices with $\min(m_k, n_k) \geq 256$. For all other matrices, we learn a single linear mixing weight $c_{i,k}$ per specialist and use

$$W_k = W_{B,k} + \sum_{i=1}^{N} c_{i,k} \tau_{i,k}, \tag{5}$$

with no low-rank correction. Note that the gates and low-rank factors are training-time parameters only. After optimization, we evaluate Equation 4 once and store only the resulting W_k. Inference therefore requires equal memory and computation as the original base model.

5 EXPERIMENTS AND RESULTS

The experiments below test two questions. First, does LUMEN preserve more specialist knowledge than existing arithmetic and learnable merge rules at the same inference cost. Second, does the advantage persist as more specialists are added to the nested merge sequence, up to $N = 6$ specialists.

5.1 EXPERIMENTAL SETUP

Models and specialists. We use eight Qwen3.5-4B specialists trained on real-world corpora, alongside the factual probes defined in §3.2. For specialist i, we use the factual retention R_i defined in Equation 1. For a set of specialists S, we report the macro-average

$$R(S) = \frac{1}{|S|} \sum_{i \in S} R_i, \qquad R_i = 100 \cdot \frac{\mathrm{Acc}(\theta_{\mathrm{merge}}; D_i)}{\mathrm{Acc}(\theta_i; D_i)}, . \tag{6}$$

$R_i = 100$ means the merged model preserves specialist i's original factual accuracy. Averaging normalized values gives every specialist equal weight.

Training and model selection. Each specialist contributes a pool of $2000N$ facts, from which the merger draws 4 examples per specialist per step for $1{,}200$ steps, i.e. $4{,}800$ fitting draws per specialist at every rung. Holding the per-specialist draw count fixed as N grows is essential: the sampler draws $\lfloor B/N \rfloor$ examples *per specialist*, so a fixed batch would have given the top of the ladder a third of the supervision of the bottom and placed a training budget gradient along the x-axis of Table 3. We therefore set $B = 4N$. Separately, 250 probes per specialist are split evenly into 125 for checkpoint selection and 125 for reporting. Test facts were learned by the specialist but never seen by the merger during fitting or selection.

		specialists merged, N				
method	trainable params ($N = 2$)	2	3	4	5	6
Task Arithmetic	0	68.5	28.2	14.3	5.0	3.2
TIES	0	54.4	32.9	22.2	13.0	9.0
DARE	0	64.6	24.3	10.9	0.8	0.6
RegMean	0	67.3	40.4	27.1	19.6	11.1
MagMax	0	62.1	33.8	20.1	15.5	8.3
Localize-and-Stitch	0	52.5	28.1	18.8	6.1	5.2
PCB-Merging	0	35.4	27.3	18.4	5.9	4.3
aTLAS	852	<u>97.5</u>	<u>78.7</u>	<u>50.2</u>	35.8	26.8
aTLAS $\times K$	948K	61.7	53.6	42.8	<u>37.5</u>	<u>30.5</u>
LUMEN	2.16M	**101.7**	**92.3**	**79.0**	**69.0**	**60.1**
LUMEN − best prior		+4.2	+13.6	+28.8	+31.5	+29.6

Table 3: **Factual retention as specialists are merged at fixed deployment cost.** All methods produce one $1.00\times$ model. Retention is normalized per specialist; **bold** marks best overall and <u>underline</u> best prior merge method.

Baselines. We compare against Task Arithmetic, TIES, DARE, RegMean, MagMax (Marczak et al., 2024), Localize-and-Stitch (He et al., 2025a), PCB-Merging (Du et al., 2024) and aTLAS (Zhang et al., 2024), plus aTLAS$\times K$, our row-block variant with one coefficient per output-row block. All methods use the same specialists, fitting data, selection probes and reporting probes. Every method produces one materialized checkpoint, so deployed storage and inference remain matched at $1.00\times$ and one forward pass.

The dataless rules (Task Arithmetic, TIES, RegMean, and others with 0 trainable parameters) are reported at their *best* hyperparameter per cell rather than a held-out choice. These values should therefore be interpreted as oracle upper bounds, while LUMEN is selected using held-out probes. For the same reason aTLAS is reported at the best of $\{0.1, 0.01\}$ learning rate per rung: the optimum moves with N (at $N = 2$ the larger step is better by 4.2 points, at $N = 4$ the smaller step is better by 3.5), and fixing one value would have understated the baseline at one end of the ladder.

LUMEN exceeds the strongest competing method by 4.2 points at $N = 2$. The margin is small here by construction: aTLAS already reaches 97.5, leaving almost nothing to recover. However, the capacity comparison is the informative one. aTLAS $\times K$ has $1{,}112$ times aTLAS's parameters and retains 35.8 points *less*. Increasing the resolution of specialist selection alone does not account for the gain.

5.2 THE ADVANTAGE GROWS WITH THE NUMBER OF SPECIALISTS

We fix one ordering of the specialists and construct a nested ladder $S_2 \subset S_3 \subset \cdots \subset S_6$, where each rung adds exactly one specialist. Specialists already present remain fixed, so increasing N changes only how many specialists must coexist in the merged model.

At $N = 2$, aTLAS is already near the unmerged ceiling, leaving little room for improvement. The difference becomes much larger as more specialists are added: LUMEN exceeds aTLAS by 13.6 points at $N = 3$, 28.8 at $N = 4$, and about 33 at $N = 5$–6. Importantly, this gain does not come from increasing the correction with N. The same rank-1 correction, with 0.72M parameters over 51 matrices, is used at every rung. Only the specialist-selection parameters grow with the number of models being merged. The widening gap therefore appears while the merge-specific correction remains fixed in size.

5.3 MORE MERGE PARAMETERS ARE NOT ENOUGH

One possibility is that LUMEN succeeds simply because of its sheer parameter size. We test this with aTLAS $\times K$, a higher-capacity variant that replace each tensor-level coefficient with a much larger set of learned coefficients while keeping the same basic linear-combination objective. At $N = 2$, this increases the parameter count from 852 to $947{,}682$ ($1{,}112\times$), yet retention drops from 97.5 to 61.7.

The pattern persists at $N = 4$: aTLAS $\times K$ uses 1.90M parameters but reaches only 42.8, versus 50.2 for aTLAS and 79.0 for LUMEN.

Thus, just having more parameters is not sufficient to preserve retention. aTLAS $\times K$ learns a far larger set of mixing coefficients, whereas LUMEN constrains its gates through rank-1 factors,softmax normalization, and a residual correction, thereby imposing substantially more structure on the merge.

5.4 LUMEN ON MERGEBENCH

	score (raw %)							taught content kept (%)	
	Instr.	GSM8K	MATH	Multi.	Code	Safety	**Avg**	Aya facts	SYNTAX
Soup	9.6	42.8	1.7	47.9	37.7	36.8	30.9	63.9	40.1
Task arithmetic	**31.2**	34.6	0.2	45.5	37.6	<u>54.5</u>	<u>37.3</u>	68.5	74.8
TIES	22.2	52.7	0.6	47.8	38.6	40.4	35.1	<u>79.6</u>	55.3
RegMean	14.1	**54.7**	0.5	47.9	**40.5**	44.9	35.0	66.7	<u>83.5</u>
LUMEN	29.2	49.6	1.3	46.2	38.6	**66.9**	**41.3**	**92.6**	**96.6**
specialist	40.5	59.7	1.1	47.6	44.2	76.3	47.8	*51.9*	*87.2*
base	7.8	29.9	0.9	46.4	26.9	27.7	24.8	*21.6*	*1.8*
headroom	32.7	29.9	**0.2**	**1.2**	17.3	48.6	23.0	*30.3*	*85.4*

Table 4: **What the benchmark shows, and what it hides.** *Left:* raw scores. *Right:* retention of specialist-taught content on held-out probes. Across merges, benchmark averages differ by only 10.4 points, while retention spans 28.7 points on Aya facts and 56.4 on SYNTAX. *Headroom* is specialist minus base accuracy. **Avg** is the five-category mean, with Math given by the mean of GSM8K and MATH. **Bold** best merge; <u>underline</u> second best.

We next ask whether LUMEN's retention gains come at the expense of standard task/benchmark performance. We evaluate the same five-specialist Llama-3.2-3B merge on the full MergeBench (He et al., 2025b) suite, using its released tasks, metrics, and calibration split.

Lumen achieves the highest average score in our evaluation, 41.3, compared to 37.3 for task arithmetic, 35.1 for TIES, 35.0 for RegMean, and 30.9 for the model soup (Table 4). Through this external benchmark validation, we show that LUMEN remains competitive across instruction following, mathematics, multilinguality, and coding, while obtaining the strongest safety score among the tested merges. Thus, LUMEN's improved factual retention does not require sacrificing performance on the capabilities measured by MergeBench. In our evaluation, LUMEN achieves the strongest overall benchmark score while also preserving substantially more specialist-taught content.

6 CONCLUSION

Model merging aims to consolidate specialized models into a single checkpoint without sacrificing what those models learned during fine-tuning. Our results show that standard task benchmarks do not fully capture this objective: merged models with similar benchmark performance can preserve very different amounts of specialist-taught knowledge. To study this failure directly, we introduced a suite of ten factual datasets and showed that factual retention can drop substantially after merging, deteriorates as more specialists are combined, and can be highly uneven across specialists.

We introduced LUMEN, which jointly learns how to combine specialist updates and how to correct the resulting merged model. Across merges of two to six specialists, LUMEN improves factual retention by 21.5% on average over the strongest reported baseline at each merge size; at six specialists, it retains 60.1% of pre-merge factual performance compared with 30.5% for the strongest baseline. These gains come at no additional inference cost, and on MergeBench LUMEN also achieves the strongest average benchmark score in our evaluation while preserving substantially more specialist-taught content. Overall, our results suggest that preserving what a specialist can do and preserving what it learned are complementary objectives for model merging.

Ethics Statement

This work studies how model merging affects information/facts acquired during fine-tuning. Our experiments use publicly available models and datasets and do not involve human subjects or private user data. As with model merging more broadly, improved retention may preserve both desirable and undesirable information learned by a specialist. In light of that, our work does not attempt to distinguish between them. We therefore view factual retention as a measurement property rather than an unconditional objective for deployment.

AI Use Statement

Generative AI tools were used to assist with editing, restructuring, and improving the clarity of the manuscript. However, it is important to note that they were not used to generate experimental results or replace the authors' analysis of those results. The authors ensured that all AI-assisted text was reviewed and edited by them, who take responsibility for the entire content of the paper.

References

Jiho Choi, Donggyun Kim, Chanhyuk Lee, and Seunghoon Hong. Revisiting weight averaging for model merging, 2024. URL https://arxiv.org/abs/2412.12153.

Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, and Min Zhang. Parameter competition balancing for model merging. In *Advances in Neural Information Processing Systems*, volume 37, pp. 84746–84776, 2024. https://doi.org/10.52202/079017-2691.

Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodolà. Task singular vectors: Reducing task interference in model merging. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18695–18705. IEEE, June 2025. https://doi.org/10.1109/cvpr52734.2025.01742. URL http://dx.doi.org/10.1109/cvpr52734.2025.01742.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *Transactions on Machine Learning Research*, 2025a. URL https://openreview.net/forum?id=9CWU8Oi86d.

Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. Mergebench: A benchmark for merging domain-specialized llms. *Advances in Neural Information Processing Systems*, 38, 2025b.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic, 2022. URL https://arxiv.org/abs/2212.04089.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models, 2022. URL https://arxiv.org/abs/2212.09849.

Viet Lai, Chien Nguyen, Nghia Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan Rossi, and Thien Nguyen. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 318–327, Singapore, December 2023. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-demo.28.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation, 2023. URL https://arxiv.org/abs/2305.01210.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017. URL https://arxiv.org/abs/1711.05101.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. MagMax: Leveraging model merging for seamless continual learning. In *Computer Vision – ECCV 2024*, volume 15143 of *Lecture Notes in Computer Science*, pp. 379–395. Springer, 2024. https://doi.org/10.1007/978-3-031-73013-9_22.

Michael S Matena and Colin Raffel. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems 35*, NeurIPS 2022, pp. 17703–17716. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. https://doi.org/10.52202/068431-1287. URL http://dx.doi.org/10.52202/068431-1287.

Akshara Prabhakar, Yuanzhi Li, Karthik Narasimhan, Sham Kakade, Eran Malach, and Samy Jelassi. LoRA soups: Merging LoRAs for practical skill composition tasks, 2024. URL https://arxiv.org/abs/2410.13025.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje F Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura O'Mahony, et al. Aya dataset: An open-access collection for multilingual instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11521–11567, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2022. URL https://arxiv.org/abs/2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2022. URL https://arxiv.org/abs/2201.11903.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. Magicoder: Empowering code generation with oss-instruct. *arXiv preprint arXiv:2312.02120*, 2023.

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, 2022. URL https://arxiv.org/abs/2203.05482.

Jialin Wu, Jian Yang, Jiajun Wen, Junjie Cao, and Zhiyong Yu. CF-STAR: Highly compressible adapters for model merging via centralized task vectors. *Complex & Intelligent Systems*, 12(3), February 2026. ISSN 2198-6053. https://doi.org/10.1007/s40747-026-02238-y. URL http://dx.doi.org/10.1007/s40747-026-02238-y.

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 36*, NeurIPS 2023, pp. 7093–7115. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. https://doi.org/10.52202/075280-0310. URL http://dx.doi.org/10.52202/075280-0310.

Prateek Yadav, Tu Vu, Jonathan Lai, Alexandra Chronopoulou, Manaal Faruqui, Mohit Bansal, and Tsendsuren Munkhdalai. What matters for model merging at scale? *arXiv preprint arXiv:2410.03617*, 2024.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning, 2023. URL https://arxiv.org/abs/2310.02575.

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, 2023. URL https://arxiv.org/abs/2311.03099.

Frederic Z. Zhang, Paul Albert, Cristian Rodriguez-Opazo, Anton Van Den Hengel, and Ehsan Abbasnejad. Knowledge composition using task vectors with learned anisotropic scaling. In *Advances in Neural Information Processing Systems 37*, NeurIPS 2024, pp. 67319–67354. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024. https://doi.org/10.52202/079017-2149. URL http://dx.doi.org/10.52202/079017-2149.

A Appendix

A.1 Evaluation protocol

Prompt and scoring. Specialists are trained with the following prompt format: `{entity} > {relation} > <ans> {answer}`. During evaluation, they are queried with the prefix through `<ans>`, where the `<ans>` delimiter is a single added token to indicate to the model that it must generate the answer string. Throughout the paper, we decode greedily for at most 24 tokens and use case-sensitive exact matching of the answer, after stripping leading and trailing whitespace; no aliases, substring matching, or other normalization is used.

Generation cap. Answer-token lengths over the 48,000 scored probes are:

dataset	p50	p90	p99	max	>24 tok
NIH RePORTER	7	11	15	23	0.00%
OpenAlex	5	5	9	17	0.00%
NIH RePORTER-PI	5	7	10	13	0.00%
ClinicalTrials	4	8	14	24	0.00%
Open Library	4	7	11	43	**0.10%**
Caselaw	5	9	13	18	0.00%
UniProt	4	10	17	24	0.00%
ClinicalTrials-PI	5	9	14	23	0.00%
Wikidata	4	6	10	20	0.00%
Synthetic	4	6	7	8	0.00%

We find that only 12 probes (0.025%) exceed the generation cap, all from Open Library. This ceiling effect is constant across parent and merged models and negligible relative to the reported retention differences.

A.2 Datasets

Construction. Each fact is an entity–relation–value triple. We build ten datasets from seven public sources — UniProt, the Caselaw Access Project, ClinicalTrials.gov, NIH RePORTER, OpenAlex, Wikidata and Open Library. From both ClinicalTrials and NIH RePORTER, we yield two datasets from disjoint relation sets. That adds up to 9; the tenth, synthetic dataset (fict2), has invented entities, relations and values and comes from no public record. Instead, the entire dataset is generated synthetically by sampling random character sequences for the entities, relations, and values.

How the halves are split. Each dataset is divided into two halves and one base model is fine-tuned separately on each. We split by *entity*: every entity appears in exactly one half, and both halves ask the same relations. Two datasets differ:

	entity overlap	relation overlap	answer overlap
eight entity-split datasets	**0.0%**	**100%**	14.5–62.8%
UniProt (split by relation)	69.7%	**0.0%**	0.0%
Synthetic	35.0%	99.9%	35.4%

The property that holds everywhere is that no *prompt* is shared. Across all ten datasets, zero `(entity, relation)` pairs appear in both training halves, and zero test probes from one half appear in the other half's training data. This ensures that every fact is taught once, to one specialist. By contrast, the two halves often share the same **answer strings**. Because they ask the same relations

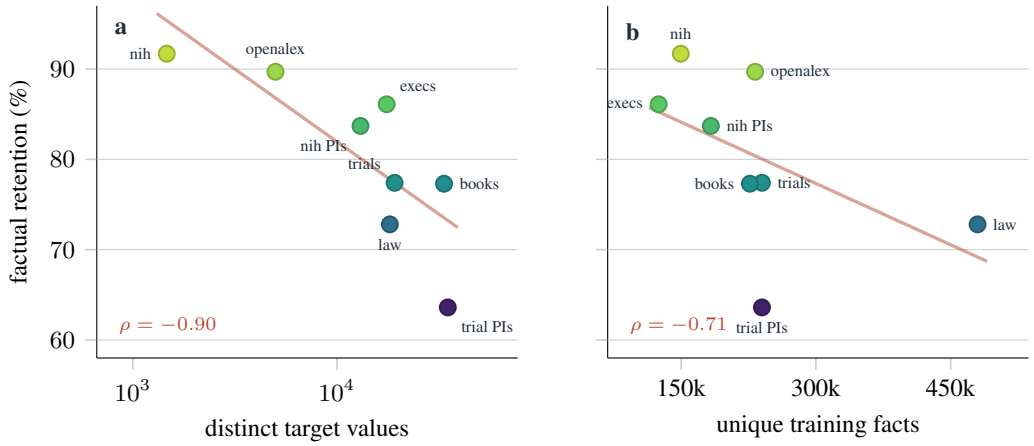


Figure 5: **Dataset properties track factual retention.** Each point is one real-world dataset included in this analysis. Retention decreases as the number of distinct target values increases (a) and as the number of unique training facts increases (b). The two properties are correlated, so these plots show association rather than causation.

about different entities, they naturally draw from a common pool of possible answers. Only UniProt has disjoint *answer vocabularies* across the two halves.

Selection gates. A dataset is kept only if the base model cannot already answer it and a frequency baseline cannot either. Under the paper's own protocol — free generation, greedy, exact match — the base model scores **0.0%** on every dataset. The frequency baseline, which predicts each relation's most common value while ignoring the entity, scores 0.5–13.7%.

A.3 Dataset properties and factual retention

Across the real-world datasets, retention is negatively associated with the number of distinct target values (Spearman $\rho = -0.90$) and unique training facts ($\rho = -0.71$). These properties are correlated, so Figure 5 is rather more descriptive than causal.

A.4 Merge rules and coefficient sweeps

Let θ_0 be the shared base, θ_i specialist i, and $\tau_i = \theta_i - \theta_0$. We evaluate Task Arithmetic (Ilharco et al., 2022), TIES (Yadav et al., 2023), DARE (Yu et al., 2023), and RegMean (Jin et al., 2022) with the settings below.

rule	definition	sweep
Task arithmetic	$\theta_0 + \alpha \sum_i \tau_i$	$\alpha \in \{0.45, \ldots, 1.3\}$, 8 values
TIES (global)	global magnitude trim, sign election, disjoint mean	$k = 0.2$, λ, 8 values
TIES (per-tensor)	as above, threshold taken per tensor	$k = 0.2$, λ, 8 values
DARE	random drop then rescale	$p \in \{0.5, 0.9\}$, λ, 4 values
RegMean	$M_h = S^{-1} G_h$, non-diagonal factor 0.9	single setting

Each cell reports the coefficient maximizing its own retention; the observed optima are interior to the swept ranges. RegMean uses a single setting, while the other rules receive a sweep, which if anything favors the swept baselines.

A.5 Training

Specialists use the shared training recipe below.

base model	Qwen3.5-4B
optimiser	AdamW (Loshchilov & Hutter, 2017), $\mathrm{lr} = 2 \times 10^{-5}$
effective batch	32
epochs	8
max sequence length	256 tokens
gradient clipping	1.0 (global norm)
seed	0
weights / compute	fp32 master, bf16 autocast, no `GradScaler`
checkpoint dtype	fp32

Dataset sizes count unique facts rather than repeated exposures. All runs use fp32 master weights with bfloat16 autocast compute and no gradient scaler.

LUMEN hyperparameters. Both the gate and correction ranks are 1. They are applied to every fourth non-embedding matrix with $\min(m_k, n_k) \geq 256$; remaining matrices receive one scalar mixing coefficient per specialist.

gate rank r_g	1	merge lr	0.03
correction rank r_c	1	lr for α_k, $c_{i,k}$	1×10^{-3}
gate / correction stride	every 4th matrix	weight decay	0
min matrix side	256	seed	0

MergeBench fits draw from a 5K-row pool for 400 steps. The N-way ladder uses batch $4N$ for 1,200 steps, drawing four examples per specialist per step so the per-specialist fitting budget is fixed as N grows.

Initialization. LUMEN starts exactly at uniform weight averaging: all specialist gates equal $1/N$, scalar mixing coefficients equal $1/N$, and the residual correction is zero.

A.6 FULL RETENTION RESULTS

Retention is reported as a percentage of each specialist's own unmerged accuracy, with every rule at its best evaluated coefficient.

cell	TA	TIES	TIES-t	DARE	RM	best
same corpus, split by entity — specialists share all relations						
NIH RePORTER	83.6	**91.7**	90.9	83.2	79.3	TIES
OpenAlex	87.2	**89.7**	90.0	84.6	79.7	TIES
Wikidata	81.9	**86.1**	84.6	78.7	66.2	TIES
NIH RePORTER-PI	77.3	**83.7**	80.0	75.1	73.8	TIES
ClinicalTrials	70.5	**77.4**	74.1	67.6	60.9	TIES
Open Library	71.0	**77.3**	74.0	68.8	60.7	TIES
Caselaw	67.3	**72.8**	70.7	63.7	60.9	TIES
ClinicalTrials-PI	59.1	**63.6**	61.8	52.8	48.0	TIES
Synthetic	5.6	**6.4**	6.0	3.4	5.9	TIES
no shared relations — UniProt is one corpus split by relation						
UniProt	34.0	31.8	31.8	34.0	**46.9**	RM
OpenAlex + ClinicalTrials	66.2	65.4	64.6	62.3	**74.3**	RM
Open Library + ClinicalTrials	**70.7**	64.7	63.6	67.5	63.4	TA
Open Library + OpenAlex	**67.7**	63.6	62.0	66.8	61.6	TA
OpenAlex + ClinicalTrials-PI	54.9	51.0	51.2	51.8	**58.0**	RM
Open Library + ClinicalTrials-PI	56.9	52.7	52.2	49.8	**57.2**	RM
UniProt + Caselaw	56.0	36.5	32.9	47.2	**57.2**	RM

TIES wins all nine cells whose specialists share a relation vocabulary and none of the seven without shared relations. UniProt isolates this effect within one corpus: splitting by relation makes RegMean best by 15.1 points. Plain averaging trails the best tuned rule by 10.8 points at the median across the sixteen cells.

merge	best rule	specialist A	specialist B	gap
shared relation vocabulary: eight within-dataset merges				
eight merges	—	—	—	0.6–6.9 (mean 3.0)
no shared relations: six cross-dataset merges				
Open Library + ClinicalTrials	TA	62.9	75.4	12.5
OpenAlex + ClinicalTrials	RM	81.9	66.7	15.2
UniProt + Caselaw	RM	37.3	76.7	39.4
Open Library + ClinicalTrials-PI	RM	35.0	79.2	44.2
Open Library + OpenAlex	TA	43.8	90.3	46.5
OpenAlex + ClinicalTrials-PI	RM	**90.2**	**23.0**	**67.2**

Table 5: **Aggregate retention hides specialist imbalance.** Each specialist's retention (%) is normalised by its own pre-merge accuracy; A and B follow the order in the merge name. Gaps are in percentage points. Each row uses the best-performing merge rule for that pair, as in §A.6.

A.7 Aggregate retention hides specialist imbalance

A single merge-level retention score can conceal asymmetric damage across its specialists.

With shared relations, specialist gaps are 0.6–6.9 points (mean 3.0); without shared relations they range from 12.5 to 67.2 points.

A.8 A per-fact model of which facts survive

For each fact, let the margin be the log-probability gap between the correct answer and the model's next choice. Over 400 facts per dataset, we regress the merged margin on the two specialist margins and the base margin:

$$m_{\text{merged}} = b_A\, m_A + b_B\, m_B + b_0\, m_{\text{base}}.$$

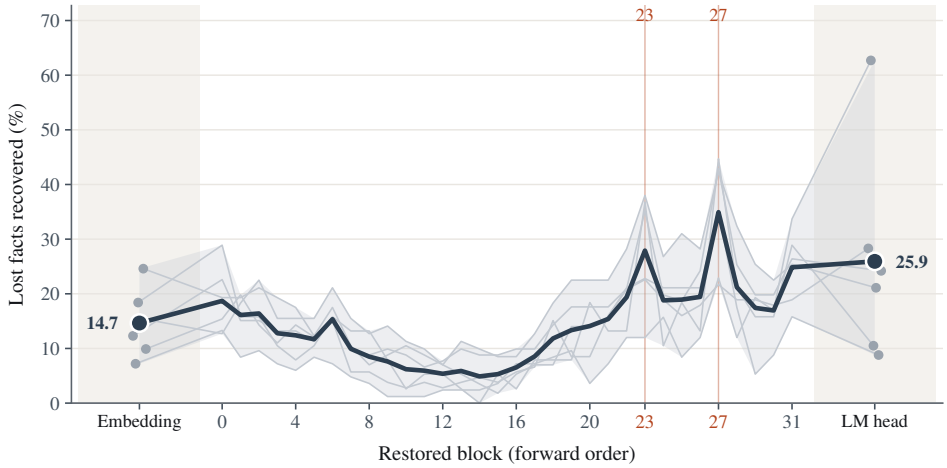
cell	answer overlap	b_A	b_B	R^2
NIH RePORTER	62.8%	0.568	+0.267	0.663
OpenAlex	54.3%	0.748	+0.234	0.802
NIH RePORTER-PI	48.5%	0.573	+0.198	0.627
ClinicalTrials	37.9%	0.552	+0.233	0.726
Open Library	21.9%	0.468	+0.149	0.440
ClinicalTrials-PI	17.6%	0.551	+0.130	0.447
mean		0.577	+0.202	0.618

The owning specialist coefficient averages $b_A = 0.577$. The partner coefficient is positive in every cell and increases with answer overlap (Spearman $\rho = 0.943$). Facts with larger parent margins are also more likely to survive the merge.

A.9 Single-Block Restoration Provides Only Partial Recovery

The preceding results establish a loss in factual recall, but do not show whether restoring a single model component can reverse it. We test this by replacing one component's weights in the merged model with those from the specialist whose facts we evaluate, leaving all other weights unchanged. A *lost fact* is a query that this specialist answers correctly but the merged model does not. We define *recovery* as the percentage of these lost facts answered correctly after restoration, using the prompting and exact-match protocol in Section 3.2. We restore the input embedding matrix, individual transformer blocks, and the language-model head separately, and compare these interventions with restoring all 32 transformer blocks together. Restoring all 32 transformer blocks recovers 92.5–100.0% of lost facts, whereas no individual transformer block recovers more than 44.7%. Recovery also varies across depth (Figure 6): restoring blocks 8–16 individually recovers less than 10% on average, compared with 27.4% and 33.9% for blocks 23 and 27, respectively. Thus, even the most effective single-block restoration falls well short of full-stack recovery. These interventions measure

which weight replacements recover lost answers; they do not by themselves establish where facts are stored or where the original failure begins.



Restored transformer weights	Lost facts recovered (%)
One block (maximum reported)	33.9
All 32 blocks (reported range)	92.5–100.0

Figure 6: **Single-block restoration provides only partial recovery.** Each position restores the indicated component (input embeddings, transformer block, or language-model head) from the evaluated specialist; all other weights remain merged. Recovery is the percentage of previously lost facts answered correctly after restoration. Faint curves show individual traces; the bold curve shows their mean. The summary reports the single-block maximum and full-stack recovery range.

Takeaway. Merging reduces factual recall in the evaluated settings, with losses that vary across specialist combinations and can fall disproportionately on one specialist. Restoring a single transformer block recovers only some of the facts lost during merging. We next address factual retention directly with LUMEN, which jointly learns how to mix specialist updates and how to correct the resulting merged weights.

A.10 PROMPTING REGIMES

We test six ways of eliciting the same facts, including self-consistency (Wang et al., 2022) and chain-of-thought prompting (Wei et al., 2022), averaged over six merges. Retention is merged accuracy divided by parent accuracy under the same prompting regime.

prompting regime	parent	merged	retention
Direct	91.8	46.2	50.4
Free generation	93.3	46.5	49.8
Self-consistency	93.8	50.8	**54.2**
Instructional preamble	71.0	32.8	46.2
Chain-of-thought (1-shot)	78.7	36.1	45.9
Chain-of-thought (answer only)	64.9	29.6	45.7

No prompting regime restores the lost facts: self-consistency is best at 54.2% retention. Chain-of-thought lowers both parent and merged accuracy, so its degradation is not merge-specific.

A.11 MERGEBENCH PROTOCOL AND REPRODUCIBILITY

All arms utilize the `lm_eval` configurations distributed with MERGEBENCHalong with their open-sources specialists, base model, and 1,000-row per-category calibration slices used to train any learnable merge-time rule.

GSM8K. Our `gsm8k_cot` configuration matches the released setup in all respects, and we compute both released filters, `flexible-extract` and `strict-match`.

Safety. While MERGEBENCH releases four safety benchmarks, it does not provide code for aggregating their results into a single score. For our evaluation, we therefore convert the harm and attack-success rates on WILDGUARDTEST, HARMBENCH, and DO-ANYTHING-NOW into refuse-to-answer rates. Lastly, combine them with the overall XSTEST accuracy, and average the four.

Published RegMean discrepancy. MERGEBENCH reports RegMean benchmark results, but does not release the corresponding merged checkpoint. The released artifacts instead contain result files for a RegMean++ run. Aggregating these files using the same rule as the published table gives 30.2 overall and 38.5 safety, compared with the reported RegMean scores of 41.5 and 71.7. We therefore merge and evaluate the baselines ourselves.

A.12 PROBE CONSTRUCTION AND VALIDITY GATES

Every probe is drawn from a specialist's fine-tuning corpus and filtered to exclude all rows used to calibrate any merge rule.

Aya facts. The multilingual specialist is trained on 5.94K rows, of which 4,525 are not used for merge calibration. Post removing near-duplicates, this leaves just 208 factual probes. Last, we score the probes with greedy generation and exact match on the first-line answer and find that the specialist reaches 51.9% accuracy versus 21.6% for the base model.

Coding and instruction. We build the SYNTAX and FUNCNAME probes from held-out MAGICODER solutions. Further, the instruction probes from held-out TULU-3 persona examples using the constraints provided in the dataset. Each probe contains 400 examples and is scored with a deterministic checker; we allow up to 6,144 generated tokens, which is sufficient to avoid truncation on all evaluated outputs.

Validity gates. We report a probe only when the specialist clearly outperforms the base model and no merge exceeds the specialist. NOCOMMA and IMPORTS fail the first check.

A.13 RAW PROBE SCORES BEHIND THE RETENTION TABLE

Table 4 uses the paper's retention definition, $100 \cdot \mathrm{Acc}(\theta_{\mathrm{merge}})/\mathrm{Acc}(\theta_i)$. The base model is shown only to indicate the headroom created by fine-tuning; it is not subtracted when computing retention.

	Aya	instruction (n=400 each)		coding (n=400)	
	GEN (k/208)	JSON	P.S.	SYNTAX	FUNCNAME
specialist	*108/208*	*94.8*	*93.2*	*87.2*	*40.2*
base	*45/208*	*35.5*	*47.5*	*1.8*	*6.5*
headroom	*63/208*	*59.3*	*45.7*	*85.4*	*33.7*
Soup	69/208	78.0	39.2	35.0	26.5
Task arithmetic	74/208	81.5	53.2	65.2	24.2
TIES	86/208	86.5	46.8	48.2	24.2
RegMean	72/208	84.5	58.2	72.8	34.0
LUMEN	100/208	92.0	89.2	84.2	36.2

P.S. and FUNCNAME, omitted from the main table for space, are:

	P.S.	FUNCNAME
Soup	42.1	65.9
Task arithmetic	57.1	60.2
TIES	50.2	60.2
RegMean	62.4	84.6
LUMEN	**95.7**	**90.0**
specialist / base (raw)	*93.2 / 47.5*	*40.2 / 6.5*

The Aya column is reported as a count out of 208 so that its retention values can be recovered exactly.

A.14 THE MERGEBENCH MATH METRIC

MergeBench reports a single Math score, but its paper description and released configuration do not uniquely determine that number. GSM8K is fixed by the released setup; the ambiguity is in MATH.

	`hendrycks_math`	`minerva_math` (4-shot CoT)	
	0-shot, EM	EM	`math_verify`
maths specialist	*1.14*	*0.04*	***27.94***
base (no fine-tuning)	*0.92*	***6.62***	*8.90*
headroom	$+0.22$	$\mathbf{-6.58}$	$\mathbf{+19.04}$
LUMEN	1.30	15.82	**20.18**
TIES	0.58	0.22	19.88
RegMean	0.48	0.76	18.32
Task arithmetic	0.16	0.00	15.00
Soup	1.70	5.88	13.80

Both exact-match variants provide essentially no valid specialist headroom: $+0.22$ points for `hendrycks_math` and -6.58 for Minerva exact match. Symbolic `math_verify` gives $+19.04$ points, but it is our substitution rather than MergeBench's specified metric. The main table therefore reports the released `hendrycks_math` exact-match score and keeps GSM8K and MATH as separate columns; its Math category average is their mean.