CHARM: Causal Head-Level Reweighting for LoRA Merging

Anonymous ACL submission

Abstract

Parameter-efficient fine-tuning methods such as LoRA produce a growing collection of task-specific experts, yet merging them into a single multi-task model remains brittle due to cross-task interference. However, existing approaches primarily operate in weight space, relying on global scaling, element-wise statistics, or magnitude-based selection to decide how task-specific experts should be combined. We find that expert skills concentrate in small subsets of attention heads that differ across experts and show little correspondence to weight magnitude. Building on this observation, we propose **CHARM** (**C**ausal **H**e**A**d-level **R**eweighted **M**erging), a training-free model merging method that allocates expert contributions at the attention-head level. CHARM estimates each expert's head-level causal effect via activation substitution, calibrates these effects into proportional-fairness weights, and uses the weights to merge the corresponding per-head LoRA updates. Finally, the merged update is integrated into the backbone to produce a standard checkpoint with no inference overhead. On MergeBench, a benchmark for merging domain-specialized LLMs, CHARM achieves the best normalized performance and exceeds the per-task-oracle reference level.

1 Introduction

The pre-training and fine-tuning paradigm is a widely adopted approach for adapting pretrained models to downstream tasks. Within this framework, parameter-efficient fine-tuning (PEFT) methods such as LoRA (Hu et al., 2022) enable task specialization by adding a small set of parameters while keeping the pretrained backbone fixed. This design supports maintaining a growing collection of task-specific LoRA experts built on the same backbone. However, this flexibility introduces a deployment trade-off. Serving many experts increases storage and memory overhead, whereas joint multi-task fine-tuning is costly and data-intensive. These challenges have sparked interest in model merging (Wortsman et al., 2022; Ilharco et al., 2023; Yang et al., 2025; Huang et al., 2024), which combines multiple experts into one model that retains task competence at lower operational cost.

The central obstacle to merging is cross-task interference. Different experts encode their updates in overlapping regions of the parameter space, so naively summing them makes these updates conflict and degrades performance on the constituent tasks (Ilharco et al., 2023; Yadav et al., 2023). To mitigate this interference, recent studies select or rescale parameters before combining them, under the principle that the entries most responsible for an expert's skill should be preserved while conflicting ones are suppressed. Building on this principle, they prune low-magnitude updates and resolve sign conflicts (Yadav et al., 2023), randomly drop and rescale entries (Yu et al., 2024), weight parameters by an importance estimate (Matena and Raffel, 2022; Jin et al., 2023), or keep only a sparse, localized subset of each task vector (Wang et al., 2024; He et al., 2025a). These methods improve over naive averaging, but they all operate at the level of individual parameters and overlook the functional structure inside the model.

Inspired by interpretability research which identifies the attention head as a natural functional module of the transformer (Olsson et al., 2022; Todd et al., 2024), we adopt the head as our unit of analysis and examine where an expert's ability resides. We quantify each head's role by its causal effect, a standard interpretability tool that intervenes on a component and measures the resulting change in the model's output (Vig et al., 2020; Meng et al., 2022; Geiger et al., 2025). This reveals how much a head *causally* drives the answer. Comparing each head's causal effect against its weight magnitude yields two observations (Sec. 3). A head's

causal contribution is largely unrelated to its weight magnitude, and the heads that carry each expert's skill are sparse and seldom overlap across experts. These observations suggest that merging should be guided by head-level causal evidence rather than by a single magnitude-based coefficient for each expert.

Following these observations, we propose a training-free merging method with three stages called **CHARM** (**C**ausal **H**e**A**d-level **R**eweighted **M**erging). The first stage localizes each expert's skill. For every head it measures the gain in answer likelihood when that head adopts the expert's behavior, so that importance is grounded in causal evidence instead of weight magnitude. The second stage makes these causal effects comparable across experts, which otherwise differ in scale, and then converts them into per-head allocation weights through a proportional-fairness rule (Kelly, 1997; Nash et al., 1950). This rule assigns each head to the expert it most belongs to while still retaining smaller positive contributors. The third stage applies these weights to the per-head row blocks of each LoRA's B matrix and folds the result into the backbone, producing a single standard checkpoint that introduces no inference-time overhead.

To summarize, our contributions are as follows:

- We reveal that LoRA expert abilities are organized by head-level causal structure rather than weight magnitude, which motivates replacing weight-space heuristics with causal per-head allocation (Sec. 3).
- We propose CHARM, a training-free model merging method that converts head-level causal effects into per-head allocation weights through proportional fairness, producing a single checkpoint with no inference-time overhead (Sec. 4).
- We conduct experiments on MergeBench (He et al., 2025b) across two model families, CHARM substantially outperforms baselines and exceeds the per-task-oracle reference level (Sec. 5).

2 Problem Formulation

We are given a frozen base language model M_0 with L layers and H attention heads per layer, together with N LoRA experts $\{E_i = (A_i, B_i)\}_{i=1}^{N}$ fine-tuned from M_0 on different tasks. All experts are trained on the same set of target attention projections. For any target projection, let W_0 denote its base weight in M_0. The i-th expert contributes a low-rank update

$$\Delta W_i = B_i A_i, \tag{1}$$

We omit layer and projection indices for clarity, since the same merging procedure is applied independently to every target projection in every layer.

The goal of model merging is to construct a single update $\Delta W^\star$ from $\{\Delta W_i\}_{i=1}^{N}$ and obtain the merged weight $W^\star = W_0 + \Delta W^\star$. Applying this procedure to all target projections yields one deployable model $M^\star$ that retains the abilities of all N experts without joint multi-task training or inference-time overhead.

3 Motivation

Existing merging methods mitigate cross-task interference by manipulating task-specific update vectors before aggregation. Their criteria are typically based on parameter-level statistics such as magnitude, sign consistency, or sparsity, rather than on whether the affected components are functionally responsible for an expert's task behavior. This creates a mismatch between parameter-level merging criteria and the functional organization of transformer computation.

Inspired by interpretability research that treats the attention head as a functional module of the transformer (Olsson et al., 2022; Todd et al., 2024), we take the head as the unit of analysis and measure how much each head causally contributes to an expert's task. For expert E_i we draw a small calibration set $\mathcal{D}_i^{\text{cal}}$ from its task distribution. For the head at layer l and position j, let $h_{i,l,j}$ be the output that head produces under E_i's update $\Delta W_i = B_i A_i$, and let $M_0[(l,j) \leftarrow h_{i,l,j}]$ denote a forward pass through the base model in which only that head's output is replaced by $h_{i,l,j}$. We define the head's *causal effect matrix* for E_i as the resulting gain in the log-likelihood of the target response y:

$$C_{i,l,j} = \mathbb{E}_{(x,y)\sim\mathcal{D}_i^{\text{cal}}}\Big[\log p_{M_0[(l,j)\leftarrow h_{i,l,j}]}(y \mid x) - \log p_{M_0}(y \mid x)\Big]. \tag{2}$$

Since only one head is intervened on, $C_{i,l,j}$ isolates the contribution of head at (l,j) to expert E_i's task behavior. Collecting these values gives an expert-specific causal-effect map $C_i \in \mathbb{R}^{L\times H}$.

We first analyze whether head-level causal effects align with weight magnitude. Specifically,

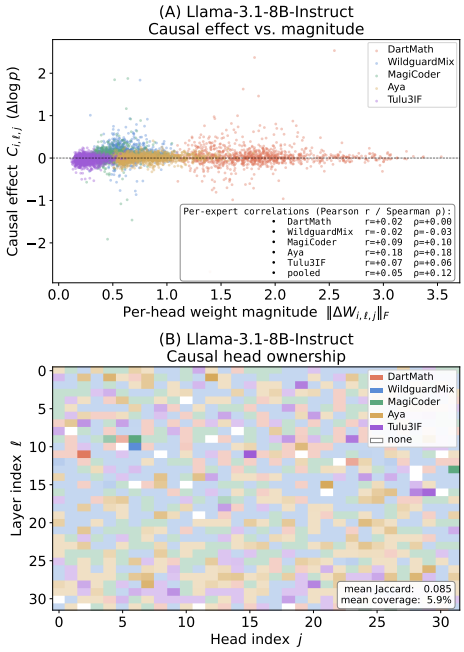

Figure 1: Causal-effect analysis on Llama-3.1-8B-Instruct. The upper panels show that head-level causal effects have little correspondence to per-head update magnitude. The lower panels show that the top positive-effect heads form small and largely distinct subsets across experts.

we compare $C_{i,l,j}$ with the per-head update magnitude $\|\Delta W_{i,l,j}\|_F$. The upper panels of Fig. 1 show that causal effect is largely decoupled from weight magnitude, indicating that large-magnitude heads are often not the ones that functionally drive an expert's task behavior. We then quantify the overlap of high-effect heads across experts by computing the pairwise Jaccard overlap between their top 10% positive-effect head sets. The lower panels of Fig. 1 show that each expert's causal effect is concentrated in a small subset of heads, and these high-effect subsets are largely distinct across experts.

These observations naturally motivate a head-level view of model merging, where allocation weights $\alpha_i^{(l,j)}$ determine each expert's contribution to each head according to causal evidence. The next section explains how we estimate, calibrate, and apply α.

4 Method

4.1 Overview

Stage 1 localizes each expert's ability to attention heads by measuring the head-level causal effect defined in Sec. 3. Stage 2 first calibrates these causal effect matrices to a comparable scale (Step 1), then applies a proportional-fairness rule that converts the calibrated values into per-head allocation weights (Step 2). Stage 3 applies these weights to the head-aligned blocks of each B_iA_i and folds them into the base, yielding the merged model.

4.2 Stage 1: Causal Localization.

We reuse the head-level causal effect $C_{i,l,j}$ defined in Eq. 2. Evaluating it for every expert E_i, layer l, and head j yields one causal-effect matrix $C_i \in \mathbb{R}^{L\times H}$ per expert, the head-level evidence that drives the allocation in Stage 2.

4.3 Stage 2: Calibration and Allocation.

Stage 2 turns the raw per-expert causal effect matrices C_i into per-head allocation weights $\alpha_i^{(l,j)}$ in two steps: a per-expert calibration that puts the matrices on a common scale (Step 1), and a proportional-fairness rule that converts the calibrated values into allocation weights (Step 2).

Step 1: Per-expert calibration. The raw matrices C_i are not directly comparable across experts because their scales reflect task-level gains over the base model rather than the concentration of expert ability. An expert may therefore have larger casual effect values simply because it yields larger likelihood improvements. To remove this scale bias before any nonlinear weighting, we rescale each matrix by its own 99th percentile $q_{0.99}(\cdot)$:

$$\widetilde{C}_{i,l,j} = C_{i,l,j} \,/\, q_{0.99}(C_i). \tag{3}$$

This preserves the relative head-importance pattern within each expert while reducing sensitivity to outlier heads.

Step 2: Proportional-fairness allocation. At each head we now translate the calibrated casual effect values into allocation weights. We pose this as a budget-allocation problem on the simplex: each head has one unit of merging budget, and we must split it across experts with positive casual effect. We seek an allocation that satisfies three properties simultaneously: it should be *Pareto-optimal* (no expert can gain weight without another losing it), *scale-invariant* (multiplying all casual effect values by a constant should not change the allocation), and *independent of irrelevant alternatives* (adding a zero casual effect expert should not change the weights of the others).

The unique allocation on the simplex satisfying all three properties is the *proportional-fairness* al-

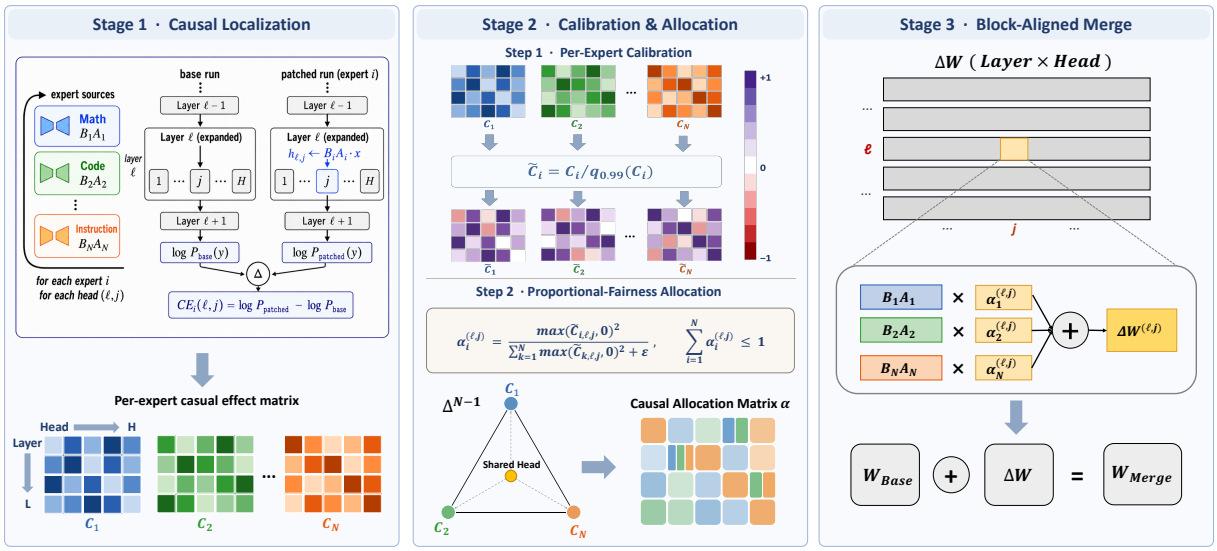

Figure 2: Overview of CHARM. The pipeline first localizes expert abilities to attention heads by causal effect, then calibrates the per-expert causal effect matrices and converts them into proportional-fairness allocation weights, and finally applies these weights to the head-aligned LoRA blocks to obtain the merged update.

location (Kelly, 1997; Nash et al., 1950):

$$\alpha_i^{(l,j)} = \frac{\max(\widetilde{C}_{i,l,j},\, 0)^2}{\sum_{k=1}^{N} \max(\widetilde{C}_{k,l,j},\, 0)^2 + \varepsilon}. \qquad (4)$$

By construction, $\sum_{i=1}^{N} \alpha_i^{(l,j)} \leq 1$. Equivalently, $\alpha^{(l,j)}$ is the closed-form solution of $\sum_i \sqrt{\alpha_i \widetilde{C}_i}$ on the simplex. This allocation rule has three desirable properties: (i) when one expert dominates a head, α collapses to a near one-hot vector and the merged head inherits that expert's solo delta unchanged—*no dilution*; (ii) when several experts contribute comparably, weights split proportionally to $\widetilde{C}^2$, so every positive contributor keeps a non-zero share—*no winner-takes-all*; (iii) when no expert has positive CE, $\alpha = 0$ and the base weight is preserved verbatim—*no noise injection*.

4.4 Stage 3: Block-aligned merging.

Finally, we apply the per-head allocation weights to the corresponding head blocks of each LoRA update. For layer l, target projection $W \in \{q, v\}_{\text{proj}}$, and head j, let $[\,\cdot\,]_j$ denote the head-j row block (rows $jd_h{:}(j{+}1)d_h$, with d_h the per-head dimension). The merged update for that block is

$$[\Delta W^\star]_j = \sum_i \alpha_i^{(l,j)}\, [B_i]_j\, A_i, \qquad (5)$$

and the merged projection is $W^\star = W + \Delta W^\star$. We provide the algorithmic pseudo-code of our merging method in Algorithm 1.

Algorithm 1 Causal Head-level Reweighted Merging.

Require: Base model M_0 with L layers and H heads; experts $\{(A_i, B_i)\}_{i=1}^N$; calibration sets $\{\mathcal{D}_i^{\text{cal}}\}_{i=1}^N$
Ensure: Merged model $M^\star$
Stage 1: Causal localization
for $i = 1, \ldots, N$; layer l; head j **do**
  $C_{i,l,j} \leftarrow$ Eq. (2)
end for
Stage 2: Calibration and allocation
Step 1: Per-expert calibration
for $i = 1, \ldots, N$ **do**
  $\widetilde{C}_i \leftarrow C_i / q_{0.99}(C_i)$
end for
Step 2: Proportional-fairness allocation
for head (l, j); expert i **do**
  $\alpha_i^{(l,j)} \leftarrow$ Eq. (4)
end for
Stage 3: Block-aligned merging
for layer l; target projection with base weight W_0; head j **do**
  $[\Delta W^\star]_j \leftarrow \sum_{i=1}^N \alpha_i^{(l,j)} [B_i]_j A_i$
end for
$W^\star \leftarrow W_0 + \Delta W^\star$ ▷ per target projection
return $M^\star$, obtained by replacing every W_0 with $W^\star$

5 Experiments

5.1 Setup

Task taxonomy We adopt the experimental protocol of MergeBench (He et al., 2025b), a standardized LLM-scale benchmark for evaluating the merging of domain-specialized LLMs. We follow the five task domains of MergeBench: mathematical reasoning, code generation, multilingual understanding, instruction following, and safety. For each domain we obtain a single LoRA expert by supervised fine-tuning the base model on the cor-

responding training dataset, holding the backbone frozen. We use the same five training datasets as MergeBench: DART-Math (Tong et al., 2024) for mathematics, Magicoder (Wei et al., 2024) for coding, Aya (Singh et al., 2024) for multilingual understanding, TULU-3 persona IF (Lambert et al., 2024) for instruction following, and WildGuardMix (Han et al., 2024) for safety.

Base models We evaluate on two representative open-source LLMs from different families: **Qwen2.5-7B-Instruct** (Yang et al., 2024a) and **Llama-3.1-8B-Instruct** (Grattafiori et al., 2024).

Evaluation setup. We follow the MergeBench evaluation suite (He et al., 2025b), which covers instruction following, mathematics, multilingual understanding, coding, and safety. The full list of evaluation datasets, metrics, and test-set sizes is provided in Appendix C.

Aggregate metric On the five evaluation datasets we also report the **normalized performance** of He et al. (2025b),

$$\text{NormPerf} = \frac{1}{T}\sum_{t=1}^{T}\frac{\text{merged}_t}{\text{oracle}_t}, \tag{6}$$

where merged_t and oracle_t are the merged-model and task expert scores on task t.

Baselines. We evaluate against the eight parameter-level merging methods considered in MergeBench (He et al., 2025b): Model Soup (Wortsman et al., 2022), Task Arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), DARE (Yu et al., 2024), Consensus-TA (Wang et al., 2024), and Localize-and-Stitch (He et al., 2025a). We further include KnOTS (Stoica et al., 2025), a LoRA-specific merging method that aligns expert updates into a shared subspace before merging, and WUDI-Merging (Cheng et al., 2025), a strong data-free baseline designed to reduce mutual interference among task vectors.

5.2 Main Results

We report NormPerf (Eq. 6) in Table 1. This metric measures each merged model relative to the corresponding per-task oracle and averages the ratios over the five task families. The underlying absolute per-family scores are reported in Appendix D.

Overall performance. CHARM achieves the best normalized performance on both backbones, with NormPerf scores of 1.009 on Qwen2.5-7B-Instruct and 1.161 on Llama-3.1-8B-Instruct. On Qwen2.5-7B-Instruct, CHARM is the only method that exceeds the per-task-oracle reference level, improving over the strongest baseline, KnOTS, by 0.129 NormPerf points. This gain is broad across task families: CHARM matches or exceeds the oracle on mathematics, instruction following, multilingual understanding, and coding, with safety being the only family below the oracle.

On Llama-3.1-8B-Instruct, CHARM also obtains the highest overall NormPerf, slightly surpassing Model Soup (1.161 vs. 1.158). The small margin is notable because Model Soup retains stronger safety performance (0.99 vs. 0.90), while CHARM achieves higher average performance on the four task-ability families. Excluding safety, CHARM averages 1.228 over mathematics, instruction following, multilingual understanding, and coding, compared with 1.200 for Model Soup. This suggests that head-level causal allocation is particularly effective at preserving and combining task-specific capabilities.

Domain-level trends. The strongest improvements appear in domains where task-specific LoRA updates provide clear functional signal. CHARM is the best-performing method on mathematics for both backbones and also achieves the strongest instruction-following performance. It remains competitive or best on coding, with particularly strong retention on Qwen2.5-7B-Instruct. Multilingual performance varies less across methods, suggesting that this ability is largely retained by the base model and is therefore less sensitive to the choice of merging rule.

Safety shows a different pattern. Although CHARM remains competitive, it does not match the safety oracle and is below uniform averaging on both backbones. This gap is expected to be harder to close with a likelihood-gain causal signal alone. Refusal behavior is often decision-boundary-like: small shifts in the output distribution can change whether the model refuses or complies, especially under adversarial harmful prompts. As a result, heads that are important for robust refusal may not always receive large likelihood-gain scores under standard calibration examples. Uniform averaging can preserve more of the base model's conservative safety behavior, whereas methods that actively

Table 1: NormPerf relative to the per-task oracle, with per-family ratios. NormPerf $= \frac{1}{5}\sum_f \text{merged}_f/\text{oracle}_f$; 1.000 matches the per-task oracle. **Bold**: largest non-oracle NormPerf per backbone; underlined: second largest.

	Qwen2.5-7B-Instruct						Llama-3.1-8B-Instruct					
Method	Math	IF	Multi	Code	Safety	**NormPerf**	Math	IF	Multi	Code	Safety	**NormPerf**
Oracle (per-task)	1.00	1.00	1.00	1.00	1.00	1.000	1.00	1.00	1.00	1.00	1.00	1.000
Model Soup	0.95	0.79	1.00	0.72	0.90	0.869	1.61	1.07	1.00	1.12	0.99	1.158
Task Arithmetic	1.04	0.66	0.98	0.79	0.93	0.879	1.54	0.88	0.99	1.11	0.98	1.100
TIES	1.04	0.61	0.97	0.75	0.93	0.858	1.54	0.77	0.98	1.08	0.99	1.071
DARE	1.04	0.64	0.96	0.75	0.93	0.864	1.47	0.81	0.96	1.05	1.00	1.059
Consensus-TA	1.02	0.70	0.98	0.76	0.93	0.878	1.56	0.95	0.99	1.09	1.00	1.118
L&S	1.09	0.71	0.98	0.72	0.87	0.873	1.51	0.64	0.98	1.09	0.98	1.042
KnOTS	1.05	0.63	0.97	0.82	0.93	0.880	1.53	0.85	0.95	1.15	0.99	1.094
WUDI	1.02	0.51	0.92	0.78	0.96	0.838	1.26	0.69	0.90	0.76	0.99	0.921
CHARM (ours)	1.13	1.00	1.00	1.03	0.88	**1.009**	1.67	1.09	1.00	1.15	0.90	**1.161**

merge task-specific deltas may trade off part of this behavior when improving multi-domain competence. Thus, the safety gap highlights a limitation of the current causal signal for refusal-centered tasks, rather than a broad failure of the head-level allocation principle.

Summary. Overall, CHARM is the strongest method in NormPerf on both Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct. It is the only method to exceed the oracle reference on Qwen2.5-7B-Instruct and also achieves the best overall score on Llama-3.1-8B-Instruct. Its remaining weakness is concentrated in safety, suggesting that future improvements should focus on causal signals tailored to refusal behavior.

5.3 Ablation Study

We ablate five key design choices of CHARM on Qwen2.5-7B-Instruct. Specifically, we replace proportional-fairness allocation with uniform allocation to test the allocation rule, collapse head-level weights into global expert weights to test per-head granularity, replace causal effect with update magnitude to test the importance signal, remove p_{99} calibration to test cross-expert normalization, and add a Task-Arithmetic style global scaling factor to test whether an extra merge coefficient is needed. All variants use the same evaluation protocol as the main experiments, and Table 2 reports the resulting NormPerf.

The largest drop comes from replacing per-head allocation with global expert weights, reducing NormPerf from 1.009 to 0.640. This confirms that expert contributions cannot be captured well by a single scalar per expert; the merging decision must preserve the head-level structure of the causal signal. Replacing causal effect with magnitude also causes a large drop, showing that weight magnitude is a poor proxy for functional importance. Removing p_{99} calibration similarly degrades performance, indicating that cross-expert normalization is important before allocation. Finally, adding a Task-Arithmetic style global scaling factor hurts performance, suggesting that the proportional-fairness simplex already provides sufficient control over the merged update scale.

Table 2: Ablation study on Qwen2.5-7B-Instruct. Each variant removes or replaces one component of CHARM. Δ is the change in NormPerf relative to full CHARM.

Variant	**NormPerf**	**Δ**
Full CHARM	**1.009**	–
Uniform weights	0.684	−0.324
Global expert weights	0.640	−0.368
Magnitude-based signal	0.690	−0.319
No p_{99} calibration	0.678	−0.330
Global scaling λ=0.4	0.795	−0.214

Overall, all ablations substantially underperform full CHARM, supporting the combined design of causal-effect estimation, per-expert calibration, proportional-fairness allocation, and block-aligned per-head merging.

5.4 Analysis of Per-Head Allocation

We analyze the allocation weights $\alpha \in \mathbb{R}^{N\times L\times H}$ used by CHARM in the final merge. Figures 3 and 4 visualize the allocation maps for Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct, respectively. Table 3 summarizes per-expert allocation statistics, including active-head fraction, dominant-head count, and the Gini coefficient (Cowell, 2011), which measures how concentrated the allocation weights are across heads.

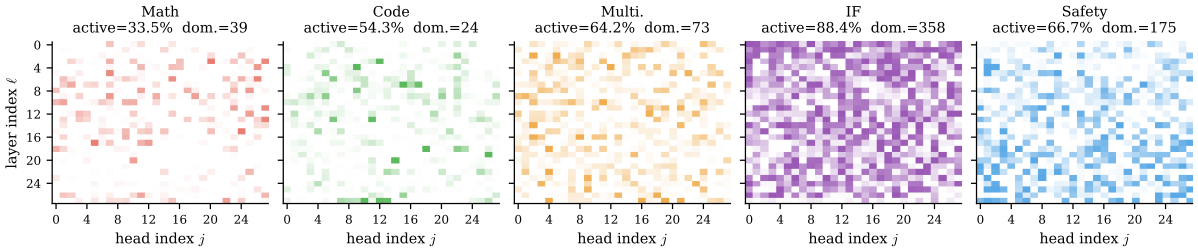

Figure 3: Per-head CHARM allocation weights $\alpha_i^{(\ell,j)}$ on Qwen2.5-7B-Instruct (28 layers $\times$ 28 query heads). Each panel corresponds to one expert. White cells indicate $\alpha \approx 0$, where the base weight is preserved. Subtitles report the fraction of active heads ($\alpha > 10^{-3}$) and the number of dominant heads ($\alpha > 0.5$), where the expert is the unique dominant contributor.

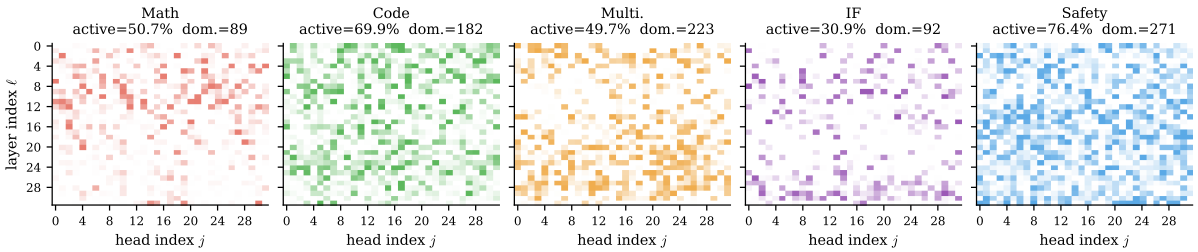

Figure 4: Per-head CHARM allocation weights $\alpha_i^{(\ell,j)}$ on Llama-3.1-8B-Instruct (32 layers $\times$ 32 query heads). The visualization follows the same format as Fig. 3.

Table 3: Per-expert allocation statistics. Act. is the fraction of heads with $\alpha > 10^{-3}$; Dom. is the number of heads with $\alpha > 0.5$; Gini measures the concentration of allocation weights across heads.

	Qwen2.5-7B-Instruct			Llama-3.1-8B-Instruct		
Expert	Act.	Dom.	Gini	Act.	Dom.	Gini
DartMath	33.6%	39	0.87	50.7%	89	0.82
Magicoder	54.3%	24	0.83	69.9%	182	0.69
Aya	64.2%	73	0.72	49.7%	223	0.72
TULU-3 IF	88.4%	358	0.45	30.9%	92	0.85
WildGuard	66.7%	175	0.66	76.4%	271	0.60
$\bigcup_i$ Dom.	669 / 784 (85.3%)			857 / 1024 (83.7%)		

Allocation decisiveness. The proportional-fairness rule satisfies $\sum_i \alpha_i^{(\ell,j)} \leq 1$, so at most one expert can have $\alpha_i^{(\ell,j)} > 0.5$ at a given head. Dominant-head sets are therefore disjoint by construction. The more informative question is how often the allocation yields a clear dominant contributor. As shown in Table 3, this occurs for most attention heads on both backbones. Specifically, 85.3% of heads on Qwen2.5-7B-Instruct and 83.7% of heads on Llama-3.1-8B-Instruct have a dominant expert. The remaining heads correspond either to shared allocations, where no expert exceeds 0.5, or to inactive heads, where no expert has positive calibrated causal effect and the base weight is preserved. Thus, the allocation is largely decisive rather than diffuse.

We use $\alpha > 0.5$ as the dominance threshold because, under the simplex constraint, it is the smallest threshold that guarantees a unique dominant expert. This conclusion is stable under stricter thresholds (Appendix B). Even at $\alpha > 0.7$, the fraction of heads with a dominant expert remains 54.6% on Qwen2.5-7B-Instruct and 55.6% on Llama-3.1-8B-Instruct. At $\alpha > 0.9$, it remains 28.1% and 27.8%, respectively.

Expert-specific allocation footprints. The allocation footprints vary substantially across experts and backbones. On Qwen2.5-7B-Instruct, DartMath is highly concentrated, with 33.6% active heads and a Gini coefficient of 0.87, whereas TULU-3 IF has the broadest footprint, with 88.4% active heads and 358 dominant heads. On Llama-3.1-8B-Instruct, this pattern changes. TULU-3 IF becomes one of the most concentrated experts, with 30.9% active heads and a Gini coefficient of 0.85, while the code and safety experts occupy broader regions of the head space.

Across all experts and both backbones, the Gini coefficients remain relatively high (0.45–0.87), indicating that even broad allocations are far from

uniform. CHARM therefore does not apply each expert uniformly across the network, but assigns expert contributions to structured subsets of attention heads.

6 Related Works

6.1 Model Merging

Model merging aims to combine multiple models fine-tuned from a shared backbone into a single deployable model without additional training (Wortsman et al., 2022; Ilharco et al., 2023; Yang et al., 2025). Early approaches average model weights or compose task vectors in parameter space, providing a simple and efficient way to reuse task-specific updates (Wortsman et al., 2022; Ilharco et al., 2023). Subsequent methods focus on mitigating cross-task interference by editing, reweighting, or sparsifying task vectors before aggregation. These include sign- or magnitude-based filtering, random drop-and-rescale strategies, importance-weighted averaging, and localized sparse merging (Matena and Raffel, 2022; Jin et al., 2023; Yadav et al., 2023; Yu et al., 2024; Wang et al., 2024; He et al., 2025a; Huang et al., 2024). Another line learns or adapts merging weights through test-time objectives or dynamic routing, improving flexibility at the cost of additional optimization or inference-time computation (Yang et al., 2024b).

Recent work also studies merging in the LoRA setting, where each expert is represented by a low-rank update rather than a full fine-tuned model. Methods such as LoRAHub, KnOTS, and WUDI-Merging learn scalar combinations of LoRA modules, align low-rank subspaces, or reduce interference among task vectors (Huang et al., 2023; Stoica et al., 2025; Cheng et al., 2025). Despite these differences, most existing merging methods decide what to preserve using parameter-level signals such as magnitude, sign agreement, sparsity, or subspace geometry. These signals are useful for reducing conflicts, but they do not directly indicate which model components are functionally responsible for an expert's task behavior. CHARM addresses this gap by using head-level causal evidence as the basis for allocation, thereby aligning the merging decision with the model's functional structure.

6.2 Functional Localization

Interpretability research has shown that transformer computations are often localized in identifiable components. Attention heads have been studied as functional modules and linked to specific behaviors such as in-context copying, circuit-level reasoning, and task-vector-like representations (Olsson et al., 2022; Wang et al., 2023; Todd et al., 2024). To move beyond correlation, causal tracing and activation patching intervene on internal activations and measure the resulting change in model behavior, providing component-level evidence of functional importance (Vig et al., 2020; Meng et al., 2022; Geiger et al., 2025).

These techniques were primarily developed for analyzing a single model. CHARM repurposes the same insight for model construction. Instead of using weight magnitude as a proxy for importance, it measures how much each attention head causally contributes to each expert's behavior and uses this signal to allocate LoRA updates at the head level. This turns functional localization from an explanatory tool into a practical principle for interference-aware expert merging.

7 Conclusion

We presented CHARM, a training-free LoRA expert merging method that allocates expert contributions at the level of attention heads. Instead of relying on parameter-level heuristics such as magnitude or sparsity, CHARM estimates head-level causal effects and converts them into per-head allocation weights through proportional fairness. The resulting block-aligned merge produces a single standard checkpoint with no routing or inference-time overhead. Our analysis shows that expert abilities are concentrated in sparse and largely distinct head subsets, and experiments on MergeBench demonstrate that CHARM achieves the strongest normalized performance on both Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct. These results suggest that causal, head-level allocation is an effective principle for reducing interference when combining specialized LoRA experts.

Limitations

CHARM currently uses a first-order likelihood-based causal signal. Each head is scored by the log-likelihood gain it produces in isolation (Eq. 2), with all other heads kept at their base values. This signal is well aligned with capability-oriented tasks, where increasing the likelihood of the target response directly reflects task competence. It also keeps the allocation procedure efficient, since CHARM only needs the relative ranking of experts

at each head.

This design naturally suggests two extensions. First, tasks governed by decision boundaries, such as refusal, may benefit from behavior-specific causal targets, for example a contrastive refuse-versus-comply objective rather than absolute likelihood. Second, higher-order attribution methods could account for joint effects among heads, such as Shapley-style credit assignment. Both extensions are compatible with the same head-level allocation framework, and we leave their cost–quality trade-off to future work.

References

Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. 2025. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

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. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Frank Cowell. 2011. *Measuring Inequality*, 3 edition. Oxford University Press.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah D. Goodman, Christopher Potts, and Thomas Icard. 2025. Causal abstraction: A theoretical foundation for mechanistic interpretability. *J. Mach. Learn. Res.*, 26:83:1–83:64.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 16 others. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. WildGuard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of LLMs. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024 – Datasets and Benchmarks Track*.

Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. 2025a. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *Trans. Mach. Learn. Res.*, 2025.

Yifei He, Siqi Zeng, Yuzheng Hu, Rui Yang, Tong Zhang, and Han Zhao. 2025b. Mergebench: A benchmark for merging domain-specialized llms. *CoRR*, abs/2505.10833.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *CoRR*, abs/2307.13269.

Chenyu Huang, Peng Ye, Tao Chen, Tong He, Xiangyu Yue, and Wanli Ouyang. 2024. Emr-merging: Tuning-free high-performance model merging. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. 2023. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Frank Kelly. 1997. Charging and rate control for elastic traffic. *Eur. Trans. Telecommun.*, 8(1):33–37.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. 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, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023*, pages 318–327. Association for Computational Linguistics.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Michael Matena and Colin Raffel. 2022. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David A. Forsyth, and Dan Hendrycks. 2024. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

John F Nash and 1 others. 1950. The bargaining problem. *Econometrica*, 18(2):155–162.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *CoRR*, abs/2209.11895.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 5377–5400. Association for Computational Linguistics.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura O'Mahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergun, Ifeoma Okoh, and 14 others. 2024. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11521–11567. Association for Computational Linguistics.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. 2025. Model merging with SVD to tie the knots. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. DART-Math: Difficulty-aware rejection tuning for mathematical problem-solving. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart M. Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jiménez, François Fleuret, and Pascal Frossard. 2024. Localizing task information for improved model merging and compression. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 50268–50287. PMLR / OpenReview.net.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2024. Magicoder: Empowering code generation with OSS-Instruct. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

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. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, pages 23965–23998. PMLR.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A. Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024a. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. 2025. Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities. *Preprint*, arXiv:2408.07666.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024b. Adamerging: Adaptive model merging for multi-task learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 57755–57775. PMLR / OpenReview.net.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *CoRR*, abs/2311.07911.

A Implementation Details

Each expert is obtained by rank-16 LoRA fine-tuning, with LoRA modules applied to the attention query and value projections (`q_proj`, `v_proj`). We use a LoRA scaling factor α_{LoRA}=32 and no dropout. Each expert is trained for one epoch with AdamW using a learning rate of 1×10^{-4}, a cosine schedule, 3% warmup, global batch size 128, maximum sequence length 4096, and bfloat16 precision. Stage 1 uses $|\mathcal{D}_i^{\text{cal}}|$=32 calibration examples per expert to estimate head-level causal effects. We set the proportional-fairness denominator floor to $\varepsilon = 10^{-12}$

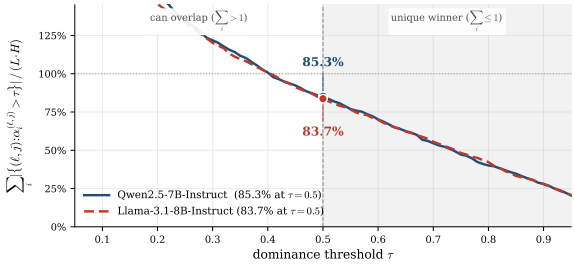

Figure 5: Sensitivity to the dominance threshold τ. We plot the sum of per-expert head counts exceeding τ, normalized by $L \cdot H$. The vertical dashed line marks $\tau = 0.5$. For $\tau < 0.5$, multiple experts can exceed the threshold at the same head, so the normalized count can exceed 100%. For $\tau \geq 0.5$, at most one expert can exceed the threshold, so the curve equals the fraction of heads with a unique dominant contributor.

B Robustness of the Dominance Threshold

Section 5.4 reports that, after proportional-fairness allocation, 85.3% of attention heads on Qwen2.5-7B-Instruct and 83.7% on Llama-3.1-8B-Instruct have one expert with $\alpha_i^{(\ell,j)} > 0.5$. This appendix explains the choice of the dominance threshold $\tau = 0.5$ and shows that the conclusion is stable under stricter thresholds.

Why use $\tau = 0.5$. For each head (ℓ, j), proportional-fairness allocation produces weights $\alpha^{(\ell,j)}$ satisfying $\sum_i \alpha_i^{(\ell,j)} \leq 1$. Under this simplex constraint, if one expert has $\alpha_i^{(\ell,j)} > \tau$, then every other expert $k \neq i$ must satisfy $\alpha_k^{(\ell,j)} \leq 1 - \tau$. The threshold $\tau = 0.5$ is therefore the smallest value that guarantees uniqueness. For $\tau < 0.5$, multiple experts may exceed the threshold at the same head. For $\tau \geq 0.5$, at most one expert can exceed the threshold, so a head counted above the threshold has a unique dominant contributor.

Robustness across thresholds. Figure 5 shows the fraction of heads whose allocation contains at least one expert above τ for $\tau \in [0.05, 0.95]$. The relevant regime is $\tau \geq 0.5$, where the curve represents the fraction of heads with a unique dominant expert. Table 4 reports the exact values at four reference thresholds.

The fraction of dominant heads decreases smoothly as the threshold becomes stricter, indicating that the conclusion is not driven by a particular

Table 4: Fraction of heads with a unique expert satisfying $\alpha > \tau$. The allocation remains decisive under stricter dominance thresholds.

τ	**Qwen2.5-7B-Instruct**	**Llama-3.1-8B-Instruct**
0.5	85.3%	83.7%
0.6	70.2%	69.8%
0.7	54.6%	55.6%
0.9	28.1%	27.8%

threshold choice. More than half of all heads still have a dominant expert at $\tau = 0.7$, where one expert must receive more than two thirds of the available allocation mass. Even at $\tau = 0.9$, which approaches a near one-hot allocation, around 28% of heads on both backbones are assigned almost exclusively to a single expert. These results show that CHARM's per-head allocation is not merely above the minimum uniqueness threshold, but remains substantially concentrated under stricter definitions of dominance.

We use $\tau = 0.5$ in the main text because it is the smallest threshold that guarantees a unique dominant expert under the simplex constraint. The qualitative conclusion that CHARM produces decisive per-head allocations is stable across the range of thresholds reported above.

C Evaluation Dataset Details

Table 5 summarizes the evaluation datasets, metrics, and test-set sizes used in our experiments, following the MergeBench evaluation suite (He et al., 2025b). The suite covers five domains: instruction following, mathematics, multilingual understanding, coding, and safety. For safety evaluation, RTA denotes the refusal success rate on harmful prompts.

D Per-Family Absolute Scores

Table 6 reports the bsolute scores underlying the NormPerf table (Table 1).

Table 5: Evaluation datasets, metrics, and number of test items following the MergeBench evaluation suite (He et al., 2025b). RTA denotes the refusal success rate on harmful prompts.

Category	**Dataset**	**Metric**	**# Data**
Instruction following	IFEval (Zhou et al., 2023)	Prompt-level strict acc.	541
Mathematics	GSM8K (Cobbe et al., 2021)	EM (8-shot CoT)	1,319
Multilingual understanding	M_MMLU (Lai et al., 2023)	Accuracy	56K
	M_ARC (Lai et al., 2023)	Normalized accuracy	10K
	M_HellaSwag (Lai et al., 2023)	Normalized accuracy	40K
Coding	HumanEval+ (Liu et al., 2023)	pass@1	164
	MBPP+ (Liu et al., 2023)	pass@1	378
Safety	WildGuardTest (Han et al., 2024)	RTA	1,730
	HarmBench (Mazeika et al., 2024)	RTA	410
	XSTest (Röttger et al., 2024)	Accuracy	450

Table 6: Absolute scores (%) on both backbones. **Bold**: best non-oracle value per column.

	Qwen2.5-7B-Instruct						**Llama-3.1-8B-Instruct**					
Method	Math	IF	Multi	Code	Safety	**Avg**	Math	IF	Multi	Code	Safety	**Avg**
Oracle (per-task)	76.6	55.5	58.1	59.7	97.3	69.4	48.6	52.7	55.2	51.0	97.3	60.9
Model Soup	72.4	43.6	57.9	42.7	87.8	60.9	78.0	56.6	**55.4**	57.3	95.9	**68.6**
Task Arithmetic	79.8	36.4	56.7	47.4	90.4	62.1	74.8	46.4	54.4	56.6	95.8	65.6
TIES	79.4	33.8	56.2	44.7	90.2	60.9	74.8	40.5	53.9	55.0	96.5	64.2
DARE	80.0	35.5	55.9	44.5	90.3	61.3	71.3	42.9	53.2	53.5	**97.1**	63.6
Consensus-TA	78.2	39.0	57.0	45.1	90.6	62.0	75.7	50.3	54.8	55.5	96.8	66.6
L&S	83.4	39.4	56.9	43.1	84.4	61.4	73.5	33.8	54.0	55.7	95.6	62.5
KnOTS	80.4	35.1	56.4	48.8	90.6	62.3	74.4	44.7	52.7	**58.6**	96.2	65.3
WUDI	78.4	28.3	53.3	46.7	**93.3**	60.0	61.4	36.2	49.8	38.8	96.4	56.5
CHARM (ours)	**86.4**	**55.3**	**58.2**	**61.8**	85.8	**69.5**	**81.2**	**57.5**	55.0	58.4	87.8	68.0