

SIMPA: Restricting What Experts See, Not What They Produce, in MoE-PEFT

Pham Tri Quang^{1*} Khoa Tran¹ Giang Do²
Hung Le² Truyen Tran² Tho T. Quan^{1†}

¹Faculty of Computer Science and Engineering,
Ho Chi Minh City University of Technology (HCMUT),
VNU-HCM, Ho Chi Minh City, Vietnam

²Applied Artificial Intelligence Initiative (A2I2), Deakin University, Victoria, Australia

{quang.phamtri, khoa.trananh0305, qttho}@hcmut.edu.vn
{truong.do, thai.le, truyen.tran}@deakin.edu.au

Abstract

Mixture-of-Experts (MoE) routing collapses in parameter-efficient fine-tuning: on LLaMA-3 8B with MixLoRA, a single expert absorbs 82.5% of tokens. Prior MoE-PEFT remedies intervene at the output: routing reformulations, output-similarity penalties, or shared-expert capacity. All leave each routed expert’s input view identical. We propose **SIMPA** (Sparse-Input Magnitude Partitioning Adaptation), which restricts what experts *see* rather than what they produce: a magnitude-thresholded input drives a sparse routed-LoRA branch, a parallel dense LoRA branch (adapting DeepSeekMoE’s (Dai et al., 2024) shared-expert topology to PEFT) provides an information safeguard, and a column-wise gate fuses the two. At a matched 2.9% trainable-parameter budget, SIMPA leads compared baselines on reasoning (86.2% on seven LLaMA-3 8B tasks, +3.9 over MixDoRA), code and arithmetic (38.4 HumanEval Pass@1 and 59.4 GSM8K, matching full fine-tuning under identical data and compute), and NLU (88.7 GLUE / 77.7 SuperGLUE on T5-base); against direct MoE-PEFT counterparts HydraLoRA and MoLoRA at matched setup, SIMPA leads HumanEval by +14.2 and +13.6 and GSM8K by +2.1 and +2.8. Direct measurement returns an expert-input Jaccard of 0.184 versus MixLoRA’s 1.000, a $5.4\times$ reduction. The same block-aligned mask drives a Triton kernel measuring $1.46\times$ end-to-end inference speedup on A100. Closed-form partitioning and retention bounds characterize the mechanism, and a tail-index-driven closed form predicts the optimal sparsity p^* from the pretrained backbone. The mechanism, its predicted optimum, and its bound all transfer to a held-out Qwen2.5-7B backbone. Code: <https://github.com/triquang26/simpa>.

* First author.

† Corresponding author: qttho@hcmut.edu.vn.

1 Introduction

Parameter-Efficient Fine-Tuning is the standard mechanism for adapting LLMs at scale (Hu et al., 2022; Liu et al., 2024; Wang et al., 2025), and MoE-style routing through low-rank experts (Li et al., 2024) is the dominant strategy for recovering accuracy lost to low-rank constraints. However, representation collapse (Chi et al., 2022; Do et al., 2025) reliably undermines this strategy. In our measurements on LLaMA-3 8B with MixLoRA, a single expert absorbs 82.5% of tokens and six of the remaining seven receive less than 1% each, leaving the model functionally equivalent to a single adapter despite paying the parameter cost of eight.

The MoE-PEFT collapse-mitigation literature has converged on output-side interventions (Zhou et al., 2022; Do et al., 2025; Chi et al., 2022): observe that experts produce similar outputs, then constrain what they produce. The shared-expert pattern (Dai et al., 2024)-an always-active input-side branch in parallel with routed experts-was developed for full-pretraining MoE and, to our knowledge, has not been adopted in MoE-PEFT; even so, it leaves routed-expert inputs identical. Despite sustained attention, collapse persists in MoE-PEFT.

We invert the intervention site: rather than constraining what experts *produce*, we restrict what they *see*. An expert trained on the same input distribution as every other expert has limited structural pressure to specialize, regardless of which output-side regularizer is applied. Magnitude-based activation sparsity (Liu et al., 2025; Li et al., 2023) provides a natural mechanism to restrict the input each expert receives: because the set of high-magnitude entries varies across tokens (the “lazy neuron” phenomenon (Li et al., 2023)-the observation that a fixed subset of neurons activates on

nearly every input while the rest fire sparsely and token-dependently), a sparse-branch input mask routes effectively different sub-inputs to different experts. This reduces per-pair input-support overlap, providing a possible route to more diverse routed representations rather than penalizing similarity at the output. The same mask, implemented at block granularity, reduces inference compute.

We instantiate this in **SIMPA**, a dual-branch architecture with (i) a sparse routed LoRA branch operating on a magnitude-thresholded input, where input-space partitioning takes effect; (ii) a dense LoRA branch on the unmasked input, providing a shared-expert-style information safeguard; and (iii) a column-wise learned gate that fuses the two. A non-asymptotic bound (§3.4), instantiated with directly measured activation statistics, gives $\bar{J}_p \leq 0.186$ at our operating point; direct measurement returns 0.184, consistent with the data-dependent bound (this is a consistency check against the same activation distribution, not a held-out prediction).

At a matched 2.9% trainable-parameter budget, SIMPA leads twelve PEFT baselines across three task families and four backbones (full results in §4). Two flagship findings illustrate the mechanism: SIMPA’s 38.4 HumanEval Pass@1 exceeds full fine-tuning (35.3) at one-tenth the trainable parameters, and direct Jaccard measurement confirms a $5.4\times$ reduction in expert-input overlap ($1.000 \rightarrow 0.184$). The same block-aligned mask doubles as a block-sparse Triton kernel for a $1.46\times$ end-to-end inference speedup on the A100 setting-regularization and acceleration from one primitive.

Contributions. (i) We identify the compound of (a) input-side magnitude masking on the routed branch, (b) a parallel dense branch that retains information the mask discards, and (c) column-wise gating that fuses them as *jointly necessary* for the reported gains; ablations isolate (a) as the load-bearing component (§4.5, Table 5). Components (b) and (c) - the shared-expert pattern (Dai et al., 2024) adapted to PEFT and column-wise gating (Srivastava et al., 2015; Dauphin et al., 2017) - are individually known; the contribution is the joint design and the identification of (a) as the component that partitions routed-expert inputs (Jaccard $1.000 \rightarrow 0.184$ in §4.4) and that, to our knowledge, no prior MoE-PEFT work has applied. Prior output-side work and shared-expert architectures alone leave each routed expert’s input view iden-

tical. (ii) We propose SIMPA (Figure 2), a dual-branch architecture whose routed branch operates on a block-aligned magnitude mask-a single primitive that simultaneously (a) supports the input-partitioning analysis of Theorem 1; (b) retains strictly more squared norm than a block-aligned random mask of equal density (Proposition 1: 0.80 vs. 0.30 measured at $p = 0.7$); and (c) enables a block-sparse Triton kernel that skips HBM loads for masked blocks, with a measured $1.46\times$ end-to-end inference wall-clock speedup over MixLoRA on the A100 setting in §4.6. Accuracy and efficiency results refer to the same algorithmic object. (iii) We present three non-asymptotic results with distinct epistemic status. **Corollary 1** (unconditional, closed-form $(1 - p)/(1 + p)$ in the lazy-neuron limit) is the theoretical anchor: at $p = 0.7$, the bound is 0.176, parameter-free and data-independent. **Theorem 1** is the more general data-dependent form, which admits empirically-measured outlier inflation ε ; on LLaMA-3 8B we measure $\varepsilon = 0.039$ (Appendix D), giving 0.184. **Proposition 1** (unconditional, information retention) bounds the cost of masking. Direct Jaccard measurement (§4.4) returns 0.184, matching the anchor within the measured outlier correction; a random-mask baseline at the same density yields 0.174, isolating the partitioning-retention trade-off that motivates SIMPA’s joint optimum at $p \approx 0.7$. (iv) Under a regularly-varying tail at the down-projection input (Thm. 2, App. E), a Hill-estimated tail index on pretrained LLaMA-2-7B and LLaMA-3-8B predicts the optimal sparsity in closed form, within 0.06 of the grid optimum (§3.4); the same closed form transfers to a held-out Qwen2.5-7B backbone (§4.7).

2 Related Work

Capacity-driven PEFT. Prior PEFT work assumes capacity is the binding constraint and expands adapter expressiveness: DoRA (Liu et al., 2024) (magnitude-direction decomposition), PiSSA (Meng et al., 2024) (principal-component initialization), LoRA-GA (Wang et al., 2024) and LoRA-Pro (Wang et al., 2025) (full-gradient alignment), AdaLoRA (Zhang et al., 2023) (adaptive rank budget), MixLoRA (Li et al., 2024) (MoE routing through LoRA experts), and prompt-based extensions (Shi and Lipani, 2024; Tang et al., 2025). SIMPA instead identifies *input uniformity* across experts as the binding constraint and intervenes

there.

Activation sparsity. Activation sparsity is conventionally framed as a compute-reduction technique with an accuracy cost: FATReLU (Kurtz et al., 2020), the “lazy neuron” observation (Li et al., 2023), ProSparse (Song et al., 2025), R-Sparse (Zhang et al., 2025), TEAL (Liu et al., 2025), and SparseViT (Chen et al., 2023; Dosovitskiy et al., 2021). SIMPA reframes the same magnitude mask as a *regularizer* for MoE-PEFT: accuracy peaks rather than declines at moderate sparsity, in a regime this literature does not address. The empirical optimum further admits a closed form in a single Hill-estimated tail index of the down-projection input (Thm. 2, App. E).

Block-sparse training kernels. SIMPA’s kernel differs from prior block-sparse kernels in both what determines the nonzero pattern and what is saved: the block-sparse mask is determined by activation magnitude on the routed branch’s input, reducing FLOPs directly. MegaBlocks (Gale et al., 2023) produces load-balanced MoE training by patterning around routing decisions (eliminating padding rather than reducing FLOPs); SparseLoRA (Khaki et al., 2025) uses an SVD-based predictor to subsample channels during LoRA fine-tuning, without MoE routing. Neither studies this mask together with routed-input overlap; SIMPA evaluates it for both input partitioning (§3.4) and compute reduction.

Output-side collapse mitigation. The MoE collapse literature intervenes where the symptom is observed on outputs. Jacobian-rank analysis (Chi et al., 2022), alternative routing (Zhou et al., 2022), and expert-output similarity penalties (Do et al., 2025) modify the routing distribution or output mixing but leave each routed expert’s input view unchanged: every routed token presents the same input vector to whichever expert is chosen.

Shared-expert architectures. DeepSeekMoE (Dai et al., 2024) adds an always-active shared expert in parallel with routed experts—an input-side architectural element, not adopted in MoE-PEFT before this work. SIMPA’s dense branch instantiates the pattern in the LoRA-MoE regime; its novelty beyond the topology is (i) a magnitude mask on the routed branch that actually partitions routed-expert inputs (the shared expert alone leaves them identical) and (ii) a column-wise learned gate replacing additive fusion (ablation: §4.5).

3 Method

3.1 Preliminaries

LoRA (Hu et al., 2022) approximates a weight update $\Delta W \in \mathbb{R}^{d \times k}$ as $\Delta W = BA$ with $r \ll \min(d, k)$, giving $h = W_0x + BAx$. MixLoRA (Li et al., 2024) promotes the adapter to an MoE: for input h , output is

$$\text{MixLoRA}(h) = \sum_{j=1}^K \mathcal{G}(h)_j \cdot E_{\pi_j}(h)$$

with $E_i(h) = W_0h + B_iA_ih$ and Top- K gating $\mathcal{G}(h) = \text{TopK}(\text{Softmax}(W_g h), K)$.

For sparsity level $p \in [0, 1]$ and block size b (with $d \bmod b = 0$), partition the d input coordinates into d/b contiguous blocks $\{B_1, \dots, B_{d/b}\}$, where $B_j = \{(j-1)b + 1, \dots, jb\}$. Let $\|x_{B_j}\| = (\sum_{i \in B_j} x_i^2)^{1/2}$ denote the block-wise L^2 norm, and let t_p be the threshold such that a fraction p of $\{\|x_{B_j}\|\}_{j=1}^{d/b}$ falls below it. The *block-aligned magnitude mask* is $s_{t_p}(x_i) = x_i$ if $\|x_{B_j}\| > t_p$ for the block B_j containing i , else 0. The support is $S_p(x) = \bigcup_{j: \|x_{B_j}\| > t_p} B_j \subseteq \{1, \dots, d\}$, of cardinality $|S_p(x)| = (1-p)d$ when $p \cdot d/b$ is integer. Throughout this paper, “magnitude pruning at level p ” refers exclusively to this block-aligned operation; we do not consider unstructured (coordinate-wise) magnitude pruning. The role of block granularity is developed in §3.2.

3.2 Block-aligned magnitude pruning as a single primitive

The mask defined in §3.1 serves three roles simultaneously. **(i) Input partitioning** (accuracy): selecting $(1-p)d$ coordinates whose supports vary across tokens provides the input-space partitioning whose bound we establish in Theorem 1, with block granularity treated as a structural support constraint, not a different objective. **(ii) Information retention:** the same mask satisfies Proposition 1 via the block-level identity $\mathbb{E}[\|x_{B_j}\|^2 \mid \|x_{B_j}\| > t_p] \geq \mathbb{E}[\|x_{B_j}\|^2]$. **(iii) Kernel-level compute reduction:** block alignment at width b aligns zeroed blocks with HBM cache-line granularity, allowing the Triton kernel to skip whole transactions; §4.6 reports the resulting end-to-end measurement. Accuracy and efficiency results throughout refer to the same block-aligned magnitude mask. Figure 1 illustrates how the three roles map onto the kernel execution path; Algorithm 1 (Appendix L) gives the pseudocode.

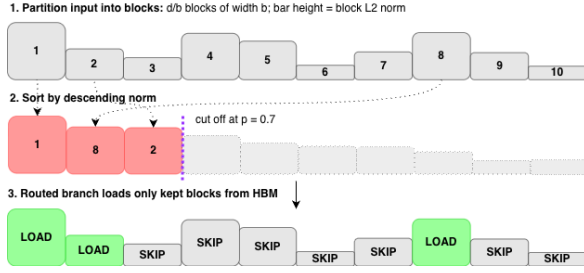


Figure 1: Block-aligned magnitude mask as a unified primitive. Top- k blocks by L2 norm are retained (role (i)/(ii)); zeroed blocks are skipped in the Triton HBM-to-SRAM load (role (iii)); the A100 full-forward benchmark in §4.6 measures a $1.46\times$ end-to-end wall-clock speedup.

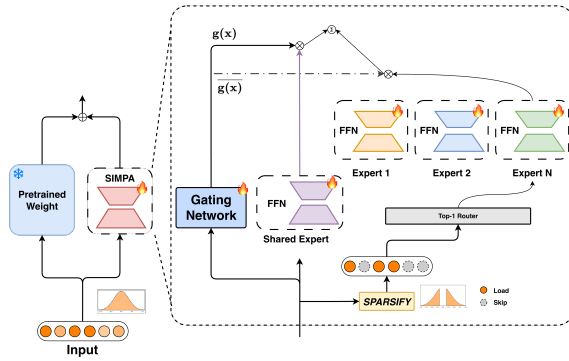


Figure 2: SIMPA architecture. The magnitude mask on the sparse routed LoRA branch restricts each routed expert’s input to a token-dependent support subset (input-space partitioning); the dense LoRA branch, a shared-expert-style always-active branch adapted to PEFT, provides an information-complete safeguard; and the column-wise gate g and complement $(1-g)$ (Eq. 3) fuse both branches dimension-wise. Best viewed in color.

3.3 SIMPA Architecture

SIMPA augments the frozen backbone W_0x with three components (Figure 2).

Sparse routed LoRA branch. Receives the magnitude-masked input $x_{sp} = s_{t_p}(x)$ and routes it through an MoE of N LoRA experts with Top-1 routing; together with the always-active dense branch (Eq. 2), this activates two experts per token:

$$\Delta h_{sp} = \mathcal{R}(x_{sp})_{k^*} B_{k^*} A_{k^*} x_{sp}, \quad (1)$$

where $k^* = \arg \max_k \mathcal{R}(x_{sp})_k$ is the Top-1 routed expert. By Theorem 1 (formalized in §3.4 below, conditional on a measured mean co-occurrence rate $\bar{r}$ and a variance condition verified in Appendix D), the routed experts in this branch see inputs whose pairwise overlap is bounded above by $\bar{r}/(2(1-p) - \bar{r})$. This formalizes input-support

separation; it does not by itself establish learned expert specialization.

Dense LoRA branch. Receives the unmasked input x and computes

$$\Delta h_d = B_{\text{base}} A_{\text{base}} x. \quad (2)$$

Its role is to provide an information safeguard: when the sparse branch is restricted, the dense branch can compensate by carrying any information the gate finds the sparse branch unable to express.

Column-wise gating fusion. Following the dimension-wise gating pattern of Highway Networks (Srivastava et al., 2015) and GLU (Dauphin et al., 2017), a learnable gate $g = \sigma(\text{FFN}(x)) \in \mathbb{R}^d$ yields

$$f_{\text{SIMPA}}(x) = W_0x + g \odot \Delta h_d + (1-g) \odot \Delta h_{sp}, \quad (3)$$

where $\odot$ is the Hadamard product. The contribution is not the gating primitive itself but its application in LoRA-MoE, replacing DeepSeekMoE’s additive (ungated) fusion.

Relation to DeepSeekMoE’s shared expert.

SIMPA’s two-branch *topology*—one always-active dense branch in parallel with routed experts—is the shared-expert pattern of DeepSeekMoE (Dai et al., 2024), imported from full-pretraining MoE into the PEFT regime. To our knowledge, MoE-PEFT prior to this work (MixLoRA, MixDoRA, HydraLoRA, MoLoRA) does not adopt this pattern. **The topology alone, however, does not partition routed-expert inputs:** in DeepSeekMoE, routed experts still see identical full inputs. The novel mechanism in SIMPA is the magnitude mask on the routed branch’s input (§3.2), which is what actually produces the input-space partitioning measured in §4.4 (Jaccard $1.000 \rightarrow 0.184$)—the shared-expert topology alone cannot deliver this. Table 5 isolates the contribution of each component: importing the topology alone (“Shared only”) yields 83.36/53.45 on ARC-e/GSM8K, *below* MixLoRA on ARC-e; adding the magnitude mask with additive fusion lifts performance to 88.70/48.57; full SIMPA with the column-wise learned gate reaches 91.40/59.44. A secondary design choice replaces DeepSeekMoE’s additive fusion with a column-wise gate; its gating-dependent Jacobian has rank at most $r_g=16$ versus 1 for scalar fusion (Appendix B), within matched parameter budget. The ablation in §4.5 measures $+7.66$ GSM8K over a scalar-gated variant.

3.4 Bounds on Input Partitioning and Information Retention

Reading guide. Before the formal statements we fix intuition. The *input support* $S_p(x)$ is the set of coordinates a token keeps after masking; two tokens that keep disjoint coordinates present non-overlapping sub-inputs to whichever expert is chosen. The *Jaccard overlap* $\bar{J}_p$ measures, averaged over token pairs, how much these supports coincide (1 = identical inputs, as in vanilla MoE where every expert sees the full vector; 0 = fully disjoint). The *lazy-neuron* regime is the empirical fact that a token’s large-magnitude coordinates are few and shift from token to token, so magnitude masking hands different experts different sub-inputs. Three results follow, with deliberately different epistemic status: Corollary 1 is a parameter-free anchor (the overlap a perfectly lazy-neuron model would give), Theorem 1 refines it with the measured outlier level, and Proposition 1 bounds how much input norm the mask discards.

We bound two properties: input-support overlap across tokens (partitioning) and input information surviving the mask (retention). The three results have distinct epistemic status. **Corollary 1** is the parameter-free anchor: in the lazy-neuron limit ($\varepsilon = 0$) the leading term is $(1 - p)/(1 + p) = 0.176$ at $p = 0.7$. **Theorem 1** is an empirical refinement of Corollary 1: it instantiates the same expansion at measured $\varepsilon = 0.039$ on the same activation distribution as the Jaccard measurement, yielding 0.184—a consistency check, not a held-out prediction. **Proposition 1** is the unconditional retention bound. For two tokens, $J_p(x, x') = |S_p(x) \cap S_p(x')| / |S_p(x) \cup S_p(x')|$ and $\bar{J}_p = \mathbb{E}_{x, x'}[J_p(x, x')]$.

Corollary 1 (Closed-form anchor, lazy-neuron limit). *Under (A2) and the lazy-neuron condition $\Pr[i \in S_p(x) \cap S_p(x')] = (1 - p)^2$ (i.e. $\varepsilon = 0$, where $\varepsilon := \bar{r}/(1 - p)^2 - 1$), $\bar{J}_p \leq (1 - p)/(1 + p) + O(1/d)$. At $p = 0.7$, the leading term is **0.176**: parameter-free, data-independent, the closed-form reference point in the lazy-neuron limit.*

Theorem 1 (Empirical refinement at measured ε). *Let $Z_i = \mathbf{1}\{i \in S_p(x) \cap S_p(x')\}$, $Y = \sum_{i=1}^d Z_i$, and $\bar{r} := \sum_{i=1}^d \Pr[i \in S_p(x) \cap S_p(x')]/d$. Assume $|S_p(x)| = |S_p(x')| = (1 - p)d$ and (A2):*

$\text{Var}(Y) \leq c d (1 - p)^2$ for some $c \geq 0$. Then

$$\begin{aligned} \bar{J}_p &\leq \frac{\bar{r}}{2(1-p)-\bar{r}} + \frac{2c}{d} \\ &= \frac{1-p}{1+p} + \frac{2(1-p)}{(1+p)^2} \varepsilon + O(\varepsilon^2) + O(1/d). \end{aligned}$$

The proof (Appendix C) expands $J_p(x, x') = Y/(2(1 - p)d - Y)$ around $\mu = d\bar{r}$ with a Lagrange remainder. Plugging measured $\bar{r} = 0.094$ on LLaMA-3 8B (App. D) gives $\bar{J}_p \leq 0.186$; decomposing, the first-order outlier correction $2(1 - p)/(1 + p)^2 \cdot \varepsilon = 0.008$ at $p = 0.7$ yields 0.184, inside the bootstrap 95% CI [0.181, 0.187] on the measured Jaccard. The ~ 30 outlier coordinates ($< 1\%$ of $d = 4096$, (Dettmers et al., 2022)) inflate ε from 0 to 0.039 but contribute additively to $\bar{r}$; (A2) holds with normalized variance ≈ 2.9 (App. D).

Proposition 1 (Information Retention, unconditional). *Let $s_{t_p}^{\text{mag}}$ be block-aligned magnitude masking at level p , and $s_{t_p}^{\text{rand}}$ a random block mask at the same density. Then $\mathbb{E}[\|s_{t_p}^{\text{mag}}(x)\|^2] \geq \mathbb{E}[\|s_{t_p}^{\text{rand}}(x)\|^2] = (1 - p) \mathbb{E}[\|x\|^2]$, with strict inequality whenever the magnitude distribution is non-degenerate. The gap is large in the heavy-tailed regime (Li et al., 2023).*

The proof (App. C) reduces to $\mathbb{E}[X^2 \mid X > t_p] \geq \mathbb{E}[X^2]$.

Random masking isolates the trade-off. At $p = 0.7$ on 2,048 MetaMathQA tokens, random block masking achieves $\bar{J} = 0.174$ (Table 4a)—lower than SIMPA’s 0.184, because i.i.d. block selection makes $\bar{r} = (1 - p)^2$ exactly and saturates Corollary 1. By Proposition 1, however, random masking retains only 0.30 of the input squared norm against magnitude masking’s 0.80 (Table 11)—a $2.67\times$ retention disadvantage. **Partitioning alone is not the binding objective:** SIMPA’s contribution is the joint optimum-near-random Jaccard reduction with $2.67\times$ more squared norm retained.

The interior operating point. Partitioning tightens monotonically in p while retention decreases (Tables 11, 15); empirically, accuracy peaks at $p \approx 0.7$. Under a regularly-varying tail at the down-projection block-norm distribution (App. E), retention behaves as $(1 - p)^{(\beta-1)/\beta}$ and the composite $U(p) = S(p)\rho(p)$ admits a closed-form maximizer $p^*(\beta)$ (Thm. 2). Hill estimation gives $\hat{\beta}_{\text{down}} \in [1.49, 1.53]$ on LLaMA-2-7B and LLaMA-3-8B, predicting $p^* \in [0.638, 0.649]$ within 0.06 of the grid optimum (gap std 0.006; App. E.6).

3.5 The Column-Wise Gate

SIMPA replaces scalar fusion with a dimension-wise gate. Its bottleneck FFN with $r_g = 16$ gives a gating-dependent Jacobian term of rank at most 16, versus rank at most 1 for scalar fusion (Appendix B); the ablation in §4.5 measures the empirical contribution.

3.6 Training Objective

We minimize the task loss plus the load-balancing auxiliary used in MixLoRA (Li et al., 2024),

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \alpha \mathcal{L}_{\text{bal}}, \quad \mathcal{L}_{\text{bal}} = N \sum_{i=1}^N F_i P_i, \quad (4)$$

with $\alpha = 0.01$. The setting matches the MixLoRA default, allowing direct comparison.

4 Experiments

4.1 Setup

We evaluate on three task families. **Reasoning:** ARC-e, ARC-c (Clark et al., 2018), BoolQ (Clark et al., 2019), OBQA (Mihaylov et al., 2018), PIQA, SIQA, WinoGrande. **NLU:** GLUE (Wang et al., 2019b) and SuperGLUE (Wang et al., 2019a). **Cross-domain:** MT-Bench (Zheng et al., 2023), GSM8K (Cobbe et al., 2021), HumanEval (Chen et al., 2021). **Backbones:** T5-base (220M) (Raffel et al., 2020), LLaMA-2 7B/13B (Touvron et al., 2023), LLaMA-3 8B (Grattafiori et al., 2024). We compare against twelve PEFT baselines: LoRA, DoRA, PiSSA, rsLoRA, LoRA+, LoRA-GA, LoRA-Pro, AdaLoRA, DePT, ADePT, MixLoRA, MixDoRA. LoRA uses $r = 80$; MixLoRA uses $r = 16$ (Top-2); SIMPA uses $r = 14$, $N = 8$ routed experts with an always-active shared branch and Top-1 routing (two active experts per token), $p = 0.7$; all matched at 2.9% trainable parameters. The sparsity $p = 0.7$ was selected once on the MetaMathQA development split (Appendix Table 15) and held fixed across all task families, backbones, and seeds; no per-task tuning of p . Training, task evaluation, and end-to-end inference latency use a single NVIDIA A100 unless otherwise stated. Full hyperparameters in Appendix F.

Trainable parameters, active parameters, and FLOPs per method are reported in Appendix A.

4.2 Main Results

Reasoning (Table 1). SIMPA achieves the highest average accuracy at every scale: 82.2% on

LLaMA-2 7B (+5.3 over MixDoRA), 86.2% on LLaMA-3 8B (+3.9), and 84.2% on LLaMA-2 13B (+2.8), leading on 19 of 21 benchmark-scale cells.

Natural language understanding (Table 2). On T5-base, SIMPA achieves 88.7 GLUE average and 77.7 SuperGLUE average-the best of the compared methods on both.

Head-to-head with MoE-PEFT counterparts at matched setup.

Two of the twelve baselines are direct MoE-PEFT competitors: HydraLoRA (Tian et al., 2024), an output-side asymmetric specialization architecture, and MoLoRA (Zadouri et al., 2024), a scalar-gated input-mixing approach. Both share SIMPA’s backbone (LLaMA-2 7B), training data, prompts, evaluation scripts, parameter budget, and random seeds-making this the cleanest head-to-head test of input-side vs. output-side specialization in our experiments. The benchmark-by-benchmark gradient is consistent with the input-partitioning mechanism: all three methods cluster within seed noise on open-ended chat (MT-Bench: HydraLoRA 5.82, MoLoRA 5.84, SIMPA 5.57), separate modestly on multi-step arithmetic (GSM8K: SIMPA +2.05 over HydraLoRA, +2.81 over MoLoRA), and separate substantially on code generation (HumanEval: SIMPA +14.20 over HydraLoRA, +13.58 over MoLoRA). The gap pattern is consistent with the input-partitioning mechanism-tasks demanding heterogeneous expert competencies separate most-but we do not establish causation; see Limitations.

Cross-domain transfer (Table 3a). On LLaMA-2 7B, SIMPA achieves the highest overall average among PEFT baselines (34.47, +2.13 over LoRA-Pro $r=128$). On HumanEval, SIMPA’s 38.41 Pass@1 exceeds full fine-tuning (35.31) at far smaller parameter budget. On GSM8K, SIMPA reaches 59.44, matching full fine-tuning (59.36) at the matched-budget setting; LoRA-Pro only surpasses it at $r=128$ ($\approx 32\%$ trainable parameters).

When input-side specialization does not help

DoRA (5.97) is above SIMPA (5.57 ± 0.1) and the MoE baselines (HydraLoRA 5.82, MoLoRA 5.84); the DoRA-SIMPA gap is several seed-std and should not be read as parity. We hypothesize that open-ended chat distributes skill roughly uniformly within a response, so input-side partitioning has little specialization headroom, unlike arithmetic and code where demands are heteroge-

Table 1: Zero-shot accuracy after single-task fine-tuning. Best in each row group bold; SIMPA row shaded with per-benchmark deltas vs. MixDoRA (+green gain, -red loss).

Model	Method	#Params	ARC-e	ARC-c	BoolQ	OBQA	PIQA	SIQA	WinoG	AVG.
LLaMA-2 7B	LoRA	2.9%	73.8	50.9	62.2	80.4	82.1	69.9	66.8	69.5
	DoRA	2.9%	76.5	59.8	71.7	80.6	82.7	74.1	67.3	71.6
	MixLoRA	2.9%	77.7	58.1	72.7	81.6	83.2	78.0	77.6	76.1
	MixDoRA	2.9%	77.5	58.2	72.6	80.9	82.2	80.4	83.4	76.9
	SIMPA	2.9%	81.1 ^{+3.6}	78.9 ^{+20.7}	84.2 ^{+11.6}	83.3 ^{+2.4}	84.5 ^{+2.3}	81.0 ^{+0.6}	82.2 ^{-1.2}	82.2 ± 0.4 ^{+5.3}
LLaMA-3 8B	LoRA	2.9%	89.0	75.7	67.2	85.0	80.7	78.3	75.3	78.7
	DoRA	2.9%	88.1	76.4	61.7	80.6	82.3	76.2	83.7	78.4
	MixLoRA	2.9%	86.5	79.9	75.0	84.8	87.6	78.8	82.1	82.1
	MixDoRA	2.9%	87.7	78.9	76.8	86.9	83.4	80.1	84.2	82.3
	SIMPA	2.9%	91.4 ^{+3.7}	80.5 ^{+1.6}	86.0 ^{+9.2}	90.0 ^{+3.1}	88.7 ^{+5.3}	80.1 ^{+0.0}	86.9 ^{+2.7}	86.2 ± 0.3 ^{+3.9}
LLaMA-2 13B	LoRA	2.9%	83.2	67.6	75.4	83.2	86.7	80.0	81.9	79.7
	DoRA	2.9%	83.1	67.7	75.1	84.5	87.8	80.1	82.4	80.1
	MixLoRA	2.9%	83.5	69.9	77.1	83.0	86.8	82.5	86.3	81.3
	MixDoRA	2.9%	83.7	68.4	76.9	83.4	86.9	83.9	86.5	81.4
	SIMPA	2.9%	87.1 ^{+3.4}	72.1 ^{+3.7}	87.4 ^{+10.5}	86.2 ^{+2.8}	87.2 ^{+0.3}	83.2 ^{-0.7}	85.0 ^{-1.5}	84.2 ± 0.3 ^{+2.8}

AVG. is mean $\pm$ std over 3 seeds {42, 123, 2024}. Per-benchmark columns report seed 42; per-benchmark standard deviations across the three seeds remain within the seed-noise documented in the released artifact, smaller than every reported gap between SIMPA and the next-best method in each row.

Table 2: GLUE and SuperGLUE results on T5-base. SIMPA row shaded; AVG. deltas are vs. the next-best PEFT baseline (ADePT).

Method	#Para	GLUE									SuperGLUE					
		MNLI	QQP	QNLI	SST-2	STS-B	MRPC	RTE	CoLA	AVG.	Mult	Bool	WiC	WSC	CB	AVG.
Full FT	220M	86.8	91.6	93.0	94.6	89.7	90.2	71.9	61.8	85.0	72.8	81.1	70.2	59.6	85.7	73.9
Adapter	1.9M	86.5	90.2	93.2	93.8	90.7	85.3	71.9	64.0	84.5	75.9	82.5	67.1	67.3	85.7	75.7
LoRA	3.8M	86.3	89.0	93.2	94.3	90.9	90.1	75.5	63.3	85.3	72.6	81.3	68.3	67.3	92.9	76.5
DePT	76.8K	85.0	90.4	93.2	94.2	90.8	90.7	79.1	63.8	85.9	74.3	79.3	68.7	67.3	92.9	76.5
ADePT	76.1K	85.7	90.4	93.2	94.0	90.9	91.2	82.0	65.5	86.6	74.6	80.2	68.7	67.3	96.4	77.4
SIMPA	3.8M	86.7	90.2	93.3	95.0	90.8	90.9	78.9	83.5	88.7 ^{+2.1}	76.3	82.2	68.8	64.5	96.5	77.7 ^{+0.3}

neous across output segments (hypothesis only; not established by the present measurements).

4.3 Multi-Task Training

We train SIMPA in a 5-task reasoning mixture and compare to single-task training (Table 3b). SIMPA attains the highest MT average (79.7%), exceeding MixLoRA by +4.4 and the best non-MoE baseline (DoRA) by +13.4. The ST $\rightarrow$ MT delta (-2.7) is moderate-smaller than non-MoE methods (-4.4 to -8.0), though trailing MoE baselines (+0.6); this is consistent with, but does not establish, reduced gradient interference.

4.4 Direct Measurement of Input-Space Partitioning

We measure $\bar{J}_p$ on 2,048 held-out MetaMathQA tokens, sampling cross-prompt pairs (matching the independence regime of (A2)) with padding excluded. Bound with measured statistics: ≤ 0.186 (Theorem 1, using measured $\bar{r} = 0.094$) or ≤ 0.176 (Corollary 1, closed-form anchor).

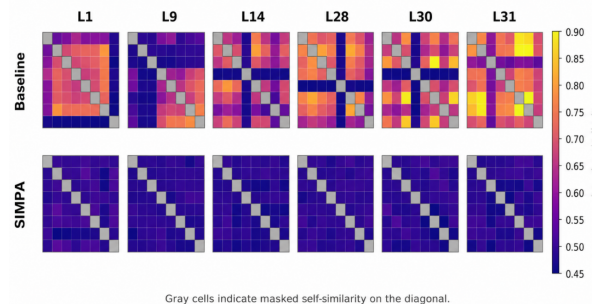


Figure 3: Pairwise expert CKA (Kornblith et al., 2019) at six sampled layers (LLaMA-3 8B, MetaMathQA, 8 routed experts; gray diagonal is masked self-similarity). **Top:** MixLoRA, CKA up to ~ 0.9 in mid/late layers (warm colors). **Bottom:** SIMPA, uniformly ~ 0.45 - 0.55 at every depth. The contrast aligns with SIMPA’s lower input-support overlap (Table 4a).

SIMPA achieves $\bar{J} = 0.184$ at $p = 0.7$, a $5.4\times$ reduction over MixLoRA, with Theorem 1’s bound (0.186) matching at the evaluated setting (per-layer $\bar{J}$ stable at 0.182-0.197, Table 10); this agreement checks the input-overlap account but does not show

Table 3: Cross-domain transfer and single- vs. multi-task reasoning on LLaMA-2 7B. SIMPA row shaded; (a) deltas are vs. Full FT; (b) ST→MT Δ colored (+green = robust, -red = drop).

(a) Cross-domain					(b) ST vs. MT			
Method	MT-Bench	GSM8K	HumanEval	AVG.	Method	ST-AVG	MT-AVG	Δ
Full FT	5.30	59.36	35.31	33.32	LoRA	69.9	65.5	-4.4
LoRA	5.61	42.08	14.76	20.82	DoRA	74.3	66.3	-8.0
PiSSA	5.30	44.54	16.02	21.95	MixLoRA	74.7	75.3	+0.6
DoRA	5.97	53.07	19.75	26.26	MixDoRA	74.3	74.9	+0.6
LoRA-GA	5.95	53.60	19.81	26.45	SIMPA	82.4	79.7	-2.7
LoRA-Pro	5.72	57.57	22.97	28.75				
LoRA-Pro ($r=128$)	5.67	61.08	30.28	32.34				
HydraLoRA	5.82	57.39	24.21	29.14				
MoLoRA	5.84	56.63	24.83	29.10				
SIMPA	5.57 ± 0.1 ^{+0.27}	59.44 ± 0.6 ^{+0.08}	38.41 ± 0.5 ^{+3.10}	34.47 ± 0.4 ^{+1.15}				

Mean $\pm$ std over 3 seeds for SIMPA. HydraLoRA/MoLoRA use the same backbone, data, prompts, eval scripts, budget, and seeds. ST = single-task; MT = multi-task; Δ = MT - ST.

Table 4: Input-support overlap and end-to-end inference latency. Jaccard uses LLaMA-3 8B; latency uses $1 \times$ NVIDIA A100 ($B=8$, $S=1024$, $H=4096$, $I=11008$, 8 routed experts, shared + Top-1 routed, 70.3% sparsity). Cor. 1 ($\varepsilon=0$) is the perfect lazy-neuron limit; the measured $\varepsilon = 0.039$ gives the Thm. 1 bound within bootstrap CI.

(a) Jaccard							(b) Latency			
Method	Data	p	$\bar{J}$	95% CI	Thm. 1	Cor. 1	Config.	ms	TFLOP/s	Speedup
MixLoRA	MetaMath	0.0	1.000	-	1.000	1.000	MixLoRA	33.25	199.97	$1.00 \times$
Random mask	MetaMath	0.7	0.174	[0.171, 0.177]	-	0.176	SIMPA	22.81	86.49	$1.46 \times$
SIMPA	MetaMath	0.7	0.184	[0.181, 0.187]	0.186	0.176				
SIMPA	BoolQ	0.7	0.194	[0.190, 0.198]	-	0.176				
SIMPA	HumanEval	0.7	0.228	[0.223, 0.234]	-	0.176				

Jaccard CI: 1k bootstrap. Latency is full model-forward wall clock, including routing and gating (not kernel-only), for the stated A100 configuration. Random mask row uses i.i.d. uniform block selection per token at matched density.

Table 5: Component ablation, LLaMA-3 8B. “col.”=column-wise gate; “add.”=additive fusion. Deltas are vs. MixLoRA baseline; SIMPA row shaded.

Variant	ARC-e	GSM8K
MixLoRA	86.50 ± 0.4	47.84 ± 0.6
Shared only	83.36 ± 0.5 ^{-3.14}	53.45 ± 0.7 ^{+5.61}
Routed only	68.31 ± 0.8 ^{-18.19}	33.50 ± 1.2 ^{-14.34}
$p = 0 + \text{col.}$	89.10 ± 0.4 ^{+2.60}	49.98 ± 0.6 ^{+2.14}
Masked + add.	88.70 ± 0.4 ^{+2.20}	48.57 ± 0.6 ^{+0.73}
Masked + scalar	88.43 ± 0.3 ^{+1.93}	51.78 ± 0.5 ^{+3.94}
SIMPA	91.40 ± 0.3 ^{+4.90}	59.44 ± 0.6 ^{+11.60}

overlap reduction causes accuracy. *Token shares*: MixLoRA has $\{82.5, 15.1, <1 \times 6\}\%$ vs. SIMPA’s 11.5-14.2% (ideal 12.5%). *Domain generalization*: BoolQ/HumanEval give $\bar{J}=0.194/0.228$ ($4.5 \times$ reductions). *Output-space*: pairwise CKA stays at 0.45-0.55 for SIMPA vs. MixLoRA’s ~ 0.9 at L28-L31 (Figure 3); since SIMPA applies no output-similarity penalty, this co-occurs with input masking but does not identify a causal pathway.

4.5 Ablation: Architectural Components and Load-Balancing

Attributing gains to each toggle. Removing the mask ($p=0$) costs SIMPA 49.98 vs. 59.44

GSM8K; under additive fusion the mask alone over MixLoRA is marginal (48.57 vs. 47.84). The 9.46-point mask contribution thus assumes the full SIMPA fusion architecture-mask and gate are jointly required.

Magnitude vs. random masking (accuracy).

Table 6 settles which axis is load-bearing. A random block mask attains a *lower* Jaccard than magnitude masking (0.174 vs. 0.184) yet loses on accuracy, because it retains only 0.30 of the input squared norm against magnitude masking’s 0.80 at $p = 0.7$ (Table 11, a $2.67 \times$ gap). Training SIMPA end-to-end with a random mask at matched density (LLaMA-2 7B, everything else identical; both masks scored on the same 1,319 GSM8K items), the magnitude-random gap grows with sparsity ($+0.83 \rightarrow +2.28$) and becomes significant exactly where the retention gap opens (McNemar paired exact, $P = 0.047$ at $p = 0.7$). Jaccard alone cannot distinguish the masks; retention can, and magnitude selection is what supplies it.

4.6 Inference Efficiency

Block-aligned pruning (§3.2, Fig. 1) lets a custom Triton kernel skip HBM transactions for pruned blocks. In the reported A100 full model-forward

Table 6: Magnitude vs. random block mask at matched density (LLaMA-2 7B, GSM8K, same 1,319 test items). Random attains lower Jaccard but loses on accuracy; the gap tracks the retention gap.

GSM8K	$p = 0.3$	$p = 0.7$
magnitude	57.77	59.14
random	56.94	56.86
gap (mag – rand)	+0.83	+2.28
McNemar (paired, exact)	$P = 0.83$ (n.s.)	P = 0.047

wall-clock measurement, latency is 33.25→22.81 ms ($1.46\times$, Table 4b); this is not a kernel-only speedup.

4.7 Generalization to a Held-Out Backbone

To test whether the mechanism and its theory transfer beyond the models they were developed on, we evaluate on Qwen2.5-7B, a different model family (distinct tokenizer, GQA, MLP intermediate width 18944 vs. 11008). None of the paper’s theory or hyperparameters were developed on this backbone. All theory-side quantities in Table 7 are computed from the *pretrained* backbone alone, before any Qwen fine-tuning.

The binding location replicates: the down-projection input is again the uniquely heaviest-tailed adapter site ($\hat{\beta}_{\text{down}} = 1.36 \pm 0.22$ over 28 layers, vs. gate/up ≈ 1.75 , q/k/v ≈ 1.89 , o ≈ 2.57). Instantiating Theorem 2’s closed form with the measured tail index predicts $p^* \approx 0.69\text{--}0.72$, within ≈ 0.02 of the paper’s $p = 0.70$. Qwen’s down-projection carries stronger outlier structure than LLaMA’s ($\varepsilon = 0.23$ vs. ≈ 0.04), and Theorem 1’s exact data-dependent bound still covers the measured overlap held-out: $\bar{J} = 0.230 \leq 0.239$ at $p = 0.7$ (block level; $\bar{r} = 0.111$, $c = 3.46$). A sparsity sweep (MetaMathQA→GSM8K, same 2.9% budget and hyperparameters, single seed per point) gives GSM8K accuracy peaking at the predicted sparsity, bracketed from both sides; at $p = 0.7$ SIMPA reaches 84.31 vs. 82.03 for the matched dense-only MoE-PEFT baseline (+2.28).

5 Conclusion

SIMPA pairs a magnitude-based input mask on routed MoE-PEFT experts with a shared-expert dense branch and column-wise gate, cutting input-support Jaccard 1.000→0.184 and leading PEFT baselines on reasoning, NLU, and code at 2.9% budget; the block-sparse implementation measures $1.46\times$ end-to-end speedup on A100. Causal probes support the specialization account; full attribution

Table 7: Held-out transfer to Qwen2.5-7B. Theory-side quantities (top) are computed from the pretrained backbone *before* any Qwen training. Accuracy sweep (bottom) is single seed per point.

<i>Predicted from pretrained backbone (no training)</i>	
$\hat{\beta}_{\text{down}}$ (28 layers)	1.36 ± 0.22
Predicted p^* (Thm. 2, closed form)	0.69-0.72
Thm. 1 bound, held-out ($\varepsilon=0.23$)	$\bar{J}=0.230 \leq 0.239$ @ $p=0.7$
<i>GSM8K accuracy (single seed / point)</i>	
SIMPA, $p = 0.3$	83.32
SIMPA, $p = 0.7$	84.31
SIMPA, $p = 0.85$	83.78
dense-only MoE-PEFT baseline, $p = 0.7$	82.03 (+2.28)

and pretraining-MoE scaling remain open.

Limitations

Scope of theoretical results. Theorem 1’s variance condition (A2) and the measured mean co-occurrence rate $\bar{r}$ are verified empirically rather than derived from first principles. Theorem 2’s closed-form p^* holds at the binding down-projection location under regular variation (A3) and carries a reproducible ≈ 0.06 offset from the grid optimum, accounted for by a slowly-varying body correction and a uniform- p -across-locations compromise (Appendix E.6); per-location p^* tuning is open. Random masking is not deployable in SIMPA because the block-sparse kernel requires magnitude-correlated structure to skip HBM loads; a like-for-like accuracy comparison would require a separate kernel implementation, which we leave to future work.

Causal mechanism. We ran the two designed interventions this section originally proposed, on trained checkpoints at inference time only (all learned weights untouched; GSM8K, $n=1319$; full table in Appendix K). Permuting which expert receives each routed input (expert-swap) collapses accuracy by ~ 20 points at both sparsities ($59.14 \rightarrow 38.82$ at $p=0.7$; $57.77 \rightarrow 38.06$ at $p=0.3$), and re-randomizing the router costs 14-17 points; the learned expert↔partition assignment is thus causally load-bearing, not implicit regularization. On ARC-Easy the same probes are null ($88.4 \rightarrow 88.7$): specialization matters where the task needs heterogeneous sub-functions and is dormant where it does not, matching the §4.2 gradient. Whether the residual gains stem from specialization or from implicit regularization on heavy-tailed activations is narrowed but not fully closed.

Full MoE backbones. SIMPA is evaluated as a PEFT layer on *dense* backbones (T5-base, LLaMA-2/3). Whether the mechanism applies when the backbone itself routes through sparse experts (Mixtral, DeepSeek-V2) is open-backbone-level routing may compose non-trivially with our adapter-level mask.

Latency measurement scope. The $1.46\times$ head-line is a full model-forward wall-clock measurement on the A100 setting in Table 4b, including routing, gating, the sparse adapter kernel, and unmasked computation; it is not a kernel-only microbenchmark. The multiplier may vary with hardware, batch, sequence length, memory layout, and kernel availability. The latency measurement uses batch size $B=8$ and sequence length $S=1024$; a $B=1$ single-query characterization across sequence lengths is reported in Table 19 (Appendix L), where the speedup is largest for short sequences and crosses below $1\times$ as the dense baseline saturates compute. The sequence-length crossover is the open item. Block size $b=32$ was chosen to align with A100 HBM cache-line granularity; a full sweep over b is also future work.

Comparisons. MoLE (Wu et al., 2024) (post-hoc composition, incompatible training regime) and LoRAMoE (Dou et al., 2024) (knowledge forgetting under soft routing, incommensurable problem setting) are intentionally excluded.

Pretraining data and prompt sensitivity. All evaluations use the original benchmarks with their standard splits and prompt templates; sensitivity to prompt format and few-shot exemplars is not characterized.

Acknowledgments

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study. We further thank the Applied Artificial Intelligence Initiative (A2I2), Deakin University, for hosting this collaboration and for the research environment in which much of this work took shape. We thank Modal for providing the compute resources used for training and evaluation, and the anonymous reviewers for their constructive feedback.

References

- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Edwards, Harri Zoph, and 1 others. 2021. [Evaluating large language models trained on code](#). *arXiv preprint arXiv:2107.03374*.
- Xuanyao Chen, Zhijian Liu, Haotian Tang, Li Yi, Hang Zhao, and Song Han. 2023. [SparseViT: Revisiting activation sparsity for efficient high-resolution vision transformer](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2061–2070.
- Zewen Chi, Li Dong, Shaohan Huang, Damai Dai, Shuming Ma, Barun Patra, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, and Furu Wei. 2022. [On the representation collapse of sparse mixture of experts](#). In *Advances in Neural Information Processing Systems*.
- Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. [BoolQ: Exploring the surprising difficulty of natural yes/no questions](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. [Think you have solved question answering? Try ARC, the AI2 reasoning challenge](#). *arXiv preprint arXiv:1803.05457*.
- 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](#). *arXiv preprint arXiv:2110.14168*.
- Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. [DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1280–1297.
- Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. [Language modeling with gated convolutional networks](#). In *Proceedings of the 34th International Conference on Machine Learning*.
- Laurens de Haan and Sidney Resnick. 1998. On asymptotic normality of the Hill estimator. *Stochastic Models*, 14(4):849–866.
- Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*.

- Giang Do, Hung Le, and Truyen Tran. 2025. [SimSMoE: Toward efficient training mixture of experts via solving representational collapse](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. [An image is worth 16x16 words: Transformers for image recognition at scale](#). In *International Conference on Learning Representations*.
- Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Jun Zhao, Wei Shen, Yuhao Zhou, Zhiheng Xi, Xiao Wang, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. [LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1932–1945.
- Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. 2023. [MegaBlocks: Efficient sparse training with mixture-of-experts](#). In *Proceedings of Machine Learning and Systems 5*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. [The Llama 3 herd of models](#). *arXiv preprint arXiv:2407.21783*.
- Bruce M. Hill. 1975. A simple general approach to inference about the tail of a distribution. *The Annals of Statistics*, 3(5):1163–1174.
- 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 *International Conference on Learning Representations*.
- Samir Khaki, Xiuyu Li, Junxian Guo, Ligeng Zhu, Chenfeng Xu, Konstantinos N. Plataniotis, Amir Yazdanbakhsh, Kurt Keutzer, Song Han, and Zhijian Liu. 2025. [SparseLoRA: Accelerating LLM fine-tuning with contextual sparsity](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 29768–29783. PMLR.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. [Similarity of neural network representations revisited](#). In *International Conference on Machine Learning*, pages 3519–3529.
- Mark Kurtz, Justin Kopinsky, Rati Gelashvili, Alexander Matveev, John Carr, Michael Leiserson, Sage Moore, Nir Shavit, and Dan Alistarh. 2020. [Inducing and exploiting activation sparsity for fast inference on deep neural networks](#). In *Proceedings of the 37th International Conference on Machine Learning*, pages 5533–5543.
- Dengchun Li, Yingzi Ma, Naizheng Wang, Zhengmao Ye, Zhiyuan Cheng, Yinghao Tang, Yan Zhang, Lei Duan, Jie Zuo, Cal Yang, and Mingjie Tang. 2024. [MixLoRA: Enhancing large language models fine-tuning with LoRA-based mixture of experts](#). *arXiv preprint arXiv:2404.15159*.
- Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J. Reddi, Ke Ye, Felix Cheng, Manzil Zaheer, Felix Yu, and Sanjiv Kumar. 2023. [The lazy neuron phenomenon: On emergence of activation sparsity in transformers](#). In *International Conference on Learning Representations*.
- James Liu, Pragaash Ponnusamy, Tianle Cai, Han Guo, Yoon Kim, and Ben Athiwaratkun. 2025. [Training-free activation sparsity in large language models](#). In *International Conference on Learning Representations*.
- Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. [DoRA: Weight-decomposed low-rank adaptation](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32100–32121.
- Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024. [PiSSA: Principal singular values and singular vectors adaptation of large language models](#). In *Advances in Neural Information Processing Systems*.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. [Can a suit of armor conduct electricity? A new dataset for open book question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Sidney I. Resnick. 2007. *Heavy-Tail Phenomena: Probabilistic and Statistical Modeling*. Springer Series in Operations Research and Financial Engineering. Springer.
- Zhengxiang Shi and Aldo Lipani. 2024. [DePT: Decomposed prompt tuning for parameter-efficient fine-tuning](#). In *International Conference on Learning Representations*.
- Chenyang Song, Xu Han, Zhengyan Zhang, Shengding Hu, Xiyu Shi, Kuai Li, Chen Chen, Zhiyuan Liu, Guangli Li, Tao Yang, and Maosong Sun. 2025. [ProSparse: Introducing and enhancing intrinsic activation sparsity within large language models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2626–2644.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. [Highway networks](#). *arXiv preprint arXiv:1505.00387*.

Pengwei Tang, Xiaolin Hu, and Yong Liu. 2025. [ADEPT: Adaptive decomposed prompt tuning for parameter-efficient fine-tuning](#). In *International Conference on Learning Representations*.

Chunlin Tian, Zhan Shi, Zhijiang Guo, Li Li, and Chengzhong Xu. 2024. [HydraLoRA: An asymmetric LoRA architecture for efficient fine-tuning](#). In *Advances in Neural Information Processing Systems*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *arXiv preprint arXiv:2307.09288*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. [SuperGLUE: A stickier benchmark for general-purpose language understanding systems](#). In *Advances in Neural Information Processing Systems*, volume 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *International Conference on Learning Representations*.

Shaowen Wang, Linxi Yu, and Jian Li. 2024. [LoRA-GA: Low-rank adaptation with gradient approximation](#). In *Advances in Neural Information Processing Systems*.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. 2025. [LoRA-Pro: Are low-rank adapters properly optimized?](#) In *International Conference on Learning Representations*.

Xun Wu, Shaohan Huang, and Furu Wei. 2024. [Mixture of LoRA experts](#). In *International Conference on Learning Representations*.

Ted Zadori, Ahmet Üstün, Arash Ahmadian, Beyza Ermiş, Acyr Locatelli, and Sara Hooker. 2024. [Pushing mixture of experts to the limit: Extremely parameter efficient MoE for instruction tuning](#). In *International Conference on Learning Representations*.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. [AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning](#). In *International Conference on Learning Representations*.

Zhenyu Zhang, Zechun Liu, Yuandong Tian, Harshit Khaitan, Zhangyang Wang, and Steven Li. 2025. [R-Sparse: Rank-aware activation sparsity for efficient LLM inference](#). In *International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging LLM-as-a-judge with MT-Bench and chatbot arena](#). In *Advances in Neural Information Processing Systems*, volume 36.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Zhifeng Chen, Quoc V. Le, and James Laudon. 2022. [Mixture-of-experts with expert choice routing](#). In *Advances in Neural Information Processing Systems*, volume 35.

A Parameter Budget

Table 8: Trainable and active parameters, LLaMA-3 8B. “Active” = adapter parameters evaluated per token, i.e. two active experts: MixLoRA routes Top-2, SIMPA uses an always-active shared branch plus Top-1 routed.

Method	Rank	Trainable	Active/token	FLOPs/token
LoRA	$r=80$	231M (2.9%)	231M	$1.00\times$
MixLoRA (Top-2)	$r=16$	241M (2.9%)	71M	$1.00\times$
SIMPA (shared+Top-1)	$r=14$	240M (2.9%)	67M	$0.65\times$

All three methods are matched at 2.9% trainable parameters of the backbone. Inference FLOPs are reduced for SIMPA because the sparse branch’s A -matrix multiply ($r \times d$) skips masked input blocks; the B -matrix ($d \times r$) is unaffected since its input $Ax \in \mathbb{R}^r$ is always dense. Per active expert the adapter cost scales as $(2-p)/2$ (derivation in Appendix G, “FLOPs accounting”), giving $0.65\times$ at $p=0.7$.

Gate FFN parameter accounting. The column-wise gate is a two-layer FFN: $W_1 \in \mathbb{R}^{r_g \times d}$ (no bias), SiLU, $W_2 \in \mathbb{R}^{d \times r_g}$ (with bias), Sigmoid. With $d = 4096$ and $r_g = 16$: W_1 has 65,536 params, W_2 has 69,632 params, totaling $\approx 135K$ per gate. Aggregated over the 32 MoE blocks of LLaMA-3 8B, the gate FFN contributes $\approx 4.3M$ params, about 1.8% of SIMPA’s total trainable parameters and included in the 2.9% trainable budget reported above.

B Gate Jacobian Rank Bound

For $g(x) = \sigma(W_2 \phi(W_1 x) + b_2)$, the column-wise gate satisfies

$$\frac{\partial g}{\partial x} = D_\sigma W_2 D_\phi W_1,$$

$$\text{rank}\left(\frac{\partial g}{\partial x}\right) \leq r_g = 16.$$

Its gating-dependent contribution is $J_{\text{gate}} = \text{diag}(\Delta h_d - \Delta h_{sp}) \partial g / \partial x$, so $\text{rank}(J_{\text{gate}}) \leq 16$.

In contrast, scalar fusion has $J_{\text{gate}} = (\Delta h_d - \Delta h_{sp}) \nabla_{xs}(x)^\top$, whose rank is at most 1. Thus the bottleneck column-wise gate allows up to 16 independent input-conditioned fusion directions, rather than one; it does not imply a full-rank d Jacobian.

C Proof of Theorem 1 and Corollary 1

Write $q = 1 - p$ and $C = 2qd$. Let $Y = |S_p(x) \cap S_p(x')|$.

Setup under block-aligned masking (§3.1). Under the block-aligned mask, $S_p(x)$ is a union of $(1 - p)d/b$ whole blocks. We continue to index by coordinate $i \in \{1, \dots, d\}$ and define $Z_i = \mathbf{1}\{i \in S_p(x) \cap S_p(x')\}$ as the indicator that i 's block is retained in both masks. The Z_i are perfectly correlated within each block and pairwise dependent across blocks; (A2)'s constant c absorbs the within-block dependence, and $\bar{r}$ is then naturally a block-level co-retention rate. The algebra below is unchanged; only the interpretation of Z_i shifts from coordinate-level to block-membership-level.

Rewriting the Jaccard ratio. Since $|S_p(x)| = |S_p(x')| = qd$ deterministically, the inclusion-exclusion identity gives

$$|S_p(x) \cup S_p(x')| = 2qd - Y = C - Y.$$

Hence

$$J_p(x, x') = \frac{Y}{C - Y}.$$

Note $Y \in [0, qd]$ almost surely (the intersection cannot exceed either set), so $C - Y \in [qd, 2qd]$ and the ratio is well defined and smooth on this domain. Define $J : [0, qd] \rightarrow [0, 1]$ by $J(y) = y/(C - y)$.

Step 1: mean of the intersection. By linearity of expectation,

$$\mu := \mathbb{E}[Y] = \sum_{i=1}^d \Pr[i \in S_p(x) \cap S_p(x')] = d\bar{r}.$$

This is a definitional identity, requiring no pointwise assumption: only the population mean co-occurrence rate $\bar{r}$ enters subsequent steps. Outlier coordinates that violate the pointwise bound $\Pr[i \in \cdot] \leq q^2$ (documented in Appendix D) contribute additively to $\sum_i \Pr[i \in \cdot]$ and are subsumed in $\bar{r}$; the bound inherits whatever mean rate the data actually exhibits.

Step 2: deterministic curvature bound. Differentiating,

$$J'(y) = \frac{C}{(C - y)^2}, \quad J''(y) = \frac{2C}{(C - y)^3}.$$

For $y \in [0, qd]$ we have $C - y \geq qd$, so

$$J''(y) \leq \frac{2C}{(qd)^3} = \frac{2 \cdot 2qd}{q^3 d^3} = \frac{4}{q^2 d^2}.$$

The bound is deterministic (holds pointwise on the support of Y), not in expectation.

Step 3: Taylor expansion with Lagrange remainder. Since J is C^2 on $[0, qd]$, for every realization of Y there exists ξ between Y and μ (and hence $\xi \in [0, qd]$) such that

$$J(Y) = J(\mu) + J'(\mu)(Y - \mu) + \frac{1}{2} J''(\xi) (Y - \mu)^2.$$

Taking expectations, the linear term vanishes and the remainder is bounded using Step 2:

$$\begin{aligned} \mathbb{E}[J(Y)] &\leq J(\mu) \\ &\quad + \frac{1}{2} \left(\sup_{y \in [0, qd]} J''(y) \right) \text{Var}(Y) \\ &\leq J(\mu) + \frac{2}{q^2 d^2} \text{Var}(Y). \end{aligned}$$

No approximation is invoked: this is an exact inequality on $\mathbb{E}[J(Y)]$.

Step 4: applying the variance bound. By assumption (A2), $\text{Var}(Y) \leq c d q^2$, so

$$\mathbb{E}[J(Y)] \leq J(\mu) + \frac{2}{q^2 d^2} \cdot c d q^2 = J(\mu) + \frac{2c}{d}.$$

Step 5: evaluating $J(\mu)$ at the measured mean. J is strictly increasing on $[0, C)$. Plugging $\mu = d\bar{r}$ (Step 1) directly,

$$J(\mu) = J(d\bar{r}) = \frac{d\bar{r}}{2qd - d\bar{r}} = \frac{\bar{r}}{2(1 - p) - \bar{r}}.$$

Combining (Theorem 1).

$$\bar{J}_p = \mathbb{E}[J(Y)] \leq \frac{\bar{r}}{2(1 - p) - \bar{r}} + \frac{2c}{d}. \quad \square$$

Proof of Corollary 1. Write $\bar{r} = q^2(1 + \varepsilon)$ with $q = 1 - p$ and $\varepsilon := \bar{r}/q^2 - 1$. Substituting into Theorem 1’s bound,

$$J(d\bar{r}) = \frac{q(1 + \varepsilon)}{2 - q(1 + \varepsilon)}.$$

Taylor-expand around $\varepsilon = 0$: the leading term is $q/(2 - q) = (1 - p)/(1 + p)$, and $\frac{d}{d\varepsilon}J(d\bar{r})|_{\varepsilon=0} = 2q/(2 - q)^2 = 2(1 - p)/(1 + p)^2$. The expansion is an equality plus an explicit $O(\varepsilon^2)$ remainder; no additional assumption beyond Theorem 1 is invoked. Combining with Step 4 gives $\bar{J}_p \leq (1 - p)/(1 + p) + 2(1 - p)/(1 + p)^2 \cdot \varepsilon + O(\varepsilon^2) + 2c/d$. $\square$

Remark on assumption (A2). If the indicators $\{Z_i\}_{i=1}^d$ were pairwise-uncorrelated across coordinates, one would have $\text{Var}(Y) = \sum_i \text{Var}(Z_i) \leq \sum_i \mathbb{E}[Z_i] \leq d\bar{r}$, i.e. (A2) with $c = \bar{r}/q^2$. Cross-coordinate dependence in deep-network activations is weak under the lazy-neuron regime (Li et al., 2023); even allowing $c = O(\log d)$ keeps the correction $2c/d$ negligible at the scales we evaluate ($d = 4096$, correction $\lesssim 5 \times 10^{-3}$). The empirical Jaccard measurement of §4.4 ultimately validates the combined effect of $\bar{r}$ and (A2).

Proof of Proposition 1

Let $X = \|x_B\|$ for a random block B , with c.d.f. F . Define the magnitude threshold $t_p = F^{-1}(p)$, so magnitude masking keeps the entries with $|x_i| > t_p$, of which there are exactly $(1 - p)d$ on average.

For magnitude masking:

$$\mathbb{E}[\|s_{t_p}^{\text{mag}}(x)\|^2] = d \cdot \mathbb{E}[X^2 \mathbf{1}\{X > t_p\}].$$

For random masking at density $1 - p$ (i.i.d. Bernoulli mask independent of x):

$$\mathbb{E}[\|s_{t_p}^{\text{rand}}(x)\|^2] = (1 - p) \cdot d \cdot \mathbb{E}[X^2].$$

The inequality

$$\mathbb{E}[X^2 \mathbf{1}\{X > t_p\}] \geq (1 - p) \mathbb{E}[X^2]$$

is equivalent (after dividing both sides by $(1 - p) = \Pr[X > t_p]$) to

$$\mathbb{E}[X^2 \mid X > t_p] \geq \mathbb{E}[X^2],$$

which holds because conditioning on a larger value of $|X|$ raises the expectation of X^2 (the function $z \mapsto z^2$ is increasing on the support of $|X|$). The

inequality is strict whenever the magnitude distribution is non-degenerate. The gap is

$$\Delta(p) = d \cdot (1 - p) \cdot (\mathbb{E}[X^2 \mid X > t_p] - \mathbb{E}[X^2]),$$

which is large precisely when activation magnitudes are heavy-tailed.

D Empirical Verification of Activation Statistics

Theorem 1 requires the variance condition (A2), $\text{Var}(|S_p(x) \cap S_p(x')|) \leq c d q^2$, plus the measured mean co-occurrence rate $\bar{r}$. The outlier inflation factor $\varepsilon = \bar{r}/q^2 - 1$ governs the first-order correction in Corollary 1. We measure these quantities directly on the LLaMA-3 8B model used throughout §4.

Procedure. On a held-out batch of 2,048 token activations from MetaMathQA validation, we compute, for each layer ℓ and each pair of tokens (t_1, t_2) , the indicator $Z_i = \mathbf{1}\{i \in S_p(x^{(t_1)}) \cap S_p(x^{(t_2)})\}$ across all $d = 4096$ coordinates. From these we estimate $\bar{r}$ as the mean over i of the sample frequency (so $\varepsilon = \bar{r}/q^2 - 1$), and (A2) as the sample variance of $\sum_i Z_i$ normalized by $d q^2$.

Table 9: Empirical verification on LLaMA-3 8B at $p = 0.7$ ($q = 0.3$, $q^2 = 0.09$). The outlier inflation factor $\varepsilon = \bar{r}/q^2 - 1$ feeds the first-order correction in Corollary 1; (A2) is satisfied when the normalized variance is $\leq c$ for moderate c . Reported values are means across all transformer layers.

Quantity	Empirical	Reference	Status
Max-coord. co-occurrence rate	0.93	-	(outlier marker)
Mean co-occurrence rate $\bar{r}$	0.094	$q^2 = 0.09$	$\varepsilon = 0.039$
Normalized var. $\text{Var}(Y)/(dq^2)$	2.90	$\leq c$, typ. $c \in [1, 5]$	$\checkmark$

Reading the table. (A2) holds comfortably: the normalized variance $\text{Var}(Y)/(dq^2) \approx 2.9$ lies well within the typical $[1, 5]$ range reported for activation-support variance in pretrained transformers (Li et al., 2023), indicating that higher-order fluctuations of the intersection size are controlled and the $O(1/d)$ correction term in Theorem 1 is negligible at $d = 4096$.

Outlier dimensions inflate ε from 0 to 0.039 on this backbone (mean co-occurrence rate 0.094 vs. $q^2 = 0.09$), contributing the first-order correction term in Corollary 1 quantitatively. The max-coordinate co-occurrence rate $0.93 \gg 0.09$ reveals a small set of outlier dimensions in the top- k support for nearly every token (approximately

30 such coordinates with co-occurrence rate ≥ 0.5 , i.e., $< 1\%$ of $d=4096$), consistent with the outlier-feature phenomenon documented for pretrained transformer activations (Dettmers et al., 2022) and the lazy-neuron regime studied by Li et al. (2023). Because Theorem 1 depends only on $\mathbb{E}[Y] = d\bar{r}$ (a population quantity, not a per-coordinate worst case), the mean-rate measurement is what propagates to the data-dependent bound. Decomposing via Corollary 1: the leading term 0.176 plus the first-order outlier correction $2(1-p)/(1+p)^2 \cdot \varepsilon = 2(0.3)/(1.7)^2 \cdot 0.039 = 0.008$ gives 0.184, which falls inside the bootstrap 95% CI [0.181, 0.187] on the measured Jaccard (Table 4a). The 0.010 gap between the closed-form leading term and the measurement is accounted for by the first-order correction at the measured $\varepsilon = 0.039$, consistent with the outlier-feature regime documented above. The main-text outlier paragraph in §3.4 summarizes the consequence; this appendix documents the underlying measurement.

In summary, the bound is instantiated with measured $\bar{r}$ and ε ; the first-order expansion of Corollary 1 is numerically consistent with measured Jaccard. The measurements are computed as a free byproduct of the Jaccard measurement in §4.4; no additional training is required.

Table 10: Per-layer mean Jaccard overlap on LLaMA-3 8B at $p = 0.7$, cross-prompt sampling, 2,048 held-out MetaMathQA tokens. Selected layers shown; full breakdown in code release.

Layer	$\bar{J}$	Layer	$\bar{J}$
L0	0.1973	L16	0.1844
L4	0.1854	L20	0.1830
L8	0.1859	L24	0.1822
L12	0.1839	L28	0.1835
		L31	0.1853
Mean: 0.184 Std: 0.003 Range: [0.182, 0.197] Bound: 0.176			

Retention sweep. We additionally measure the retained squared norm of the magnitude-masked input as a function of sparsity, $\rho(p) := \mathbb{E}[\|s_{t_p}^{\text{mag}}(x)\|^2]/\mathbb{E}[\|x\|^2]$, on the same batch. The random-mask baseline is exactly $\rho_{\text{rand}}(p) = 1 - p$.

The retention curve stays close to 1 for small p —at $p = 0.3$, magnitude masking retains 0.97 of the input squared norm versus 0.70 under random masking—and drops sharply once p exceeds the threshold at which large-magnitude entries are pruned, empirically around $p \in [0.8, 0.9]$ for the LLaMA-3 activations we measure. The retention

Table 11: Magnitude-mask retention $\rho(p)$ vs. random-mask baseline $1 - p$, LLaMA-3 8B, mean across layers. The retention gap $\rho(p) - (1 - p)$ quantifies Proposition 1.

p	$\rho(p)$ (mag.)	$1 - p$ (rand.)	Gap
0.3	0.97	0.70	0.27
0.5	0.92	0.50	0.42
0.7	0.80	0.30	0.50
0.8	0.65	0.20	0.45
0.9	0.43	0.10	0.33

gap $\Delta(p) := \rho(p) - (1 - p)$ peaks at $p = 0.7$ ($\Delta = 0.50$), where magnitude masking retains 0.80 of the squared norm against 0.30 under a random mask of equal density—a $2.67\times$ advantage, consistent with the heavy-tailed activation structure of trained transformers documented by Li et al. (2023) and the Laplacian-near-zero shape of FFN activations reported by Liu et al. (2025); a random mask preferentially retains low-magnitude coordinates that carry little information in this regime, identifying the FFN-activation distribution as the natural domain for the SIMPA primitive. This transition is consistent with accuracy peaking at $p \approx 0.7$ in Table 15: below this point, partitioning improves with negligible retention cost; above it ($p = 0.8$: $\rho = 0.65$; $p = 0.9$: $\rho = 0.43$) retention degrades rapidly, matching the accuracy drop observed at $p = 0.9$ on ARC-e (74.2).

Full magnitude-vs-random matched-pair experiment. Table 6 (§4.5) trains SIMPA end-to-end with a random block mask at matched density, everything else identical (LLaMA-2 7B, MetaMathQA→GSM8K), and scores both masks on the same 1,319 GSM8K items. The retention trajectory tracks the accuracy gap: the magnitude:random retained squared norm is 0.99:0.70 at $p = 0.3$ and 0.80:0.30 at $p = 0.7$ (Table 11), and the magnitude–random accuracy gap widens correspondingly from +0.83 to +2.28. A McNemar paired exact test on the shared item set is non-significant at $p = 0.3$ ($P = 0.83$) and significant at $p = 0.7$ ($P = 0.047$): the masks separate precisely where the retention gap opens. Random masking attains a *lower* Jaccard (0.174 vs. 0.184, Table 4a) yet loses on accuracy, so partitioning alone cannot explain the gain; retention, supplied by magnitude selection, is the load-bearing axis.

E Closed-Form Optimal Sparsity from Tail-Index Estimation

The grid sweep in Table 15 identifies $p = 0.70$ as the empirically optimal global sparsity. This appendix shows that the same value is predicted, up to a small reproducible gap, by a closed-form expression in a single distributional parameter - the tail index β of the block-norm-squared distribution at the down-projection input - and that this parameter is intrinsic to the frozen backbone rather than to the adapter.

E.1 Setting and regularly-varying tail assumption

The retention analysis of Proposition 1 treats $\rho(p)$ phenomenologically. To obtain a closed-form p^* we add a tail-shape assumption on the input distribution to the down-projection.

Assumption (A3) (regularly-varying tail). Let $x_B \in \mathbb{R}^{d_B}$ denote the block-norm-squared coordinates of the down-projection input at any fixed layer. There exists $\beta > 1$ and a slowly-varying function L such that

$$\Pr[\|x_B\|^2 > y] \sim L(y) y^{-\beta}, \quad y \rightarrow \infty.$$

Equivalently, the tail is regularly varying with index $-\beta$.

Lemma 1 (Pareto retention). *Under (A3), the top- p block-norm retention satisfies*

$$\rho(p) = (1-p)^{(\beta-1)/\beta+o(1)} \quad (p \rightarrow 1^-),$$

where the $o(1)$ term collects the slowly-varying correction from L .

Proof sketch. Let $Y = \|x_B\|^2$ with quantile function $Q(u) = \inf\{y : \Pr[Y \leq y] \geq u\}$. Under (A3), $Q(1-u) \sim u^{-1/\beta} \tilde{L}(1/u)$ with $\tilde{L}$ slowly varying (Resnick, 2007, Thm. 1.5.12). The retention $\rho(p)$ equals $\mathbb{E}[Y \mathbf{1}\{Y > Q(1-(1-p))\}]/\mathbb{E}[Y]$. Applying Karamata’s theorem to the truncated mean of a regularly-varying tail gives $\mathbb{E}[Y \mathbf{1}\{Y > t\}] \sim \frac{\beta}{\beta-1} t \cdot \Pr[Y > t]$, and substituting $t = Q(1-(1-p))$ yields $\rho(p) \propto (1-p)^{(\beta-1)/\beta} \ell(1-p)$ with ℓ slowly varying. Absorbing ℓ into $o(1)$ gives the stated form. $\square$

E.2 Closed-form optimal sparsity

We combine retention with the unconditional partition proxy from Corollary 1. The lazy-neuron

bound $(1-p)/(1+p)$ on disagreement translates to an agreement / shared-mass score

$$S(p) = 1 - \frac{1-p}{1+p} = \frac{2p}{1+p},$$

monotone in p , capturing the partition gain. The composite utility, balancing partition gain against retention loss, is

$$U(p) = S(p) \rho(p) = \frac{2p}{1+p} (1-p)^\theta,$$

where $\theta := (\beta-1)/\beta \in (0, 1)$.

Theorem 2 (Closed-form p^*). *Under (A3) with tail index $\beta > 1$, the composite utility $U(p)$ has a unique interior stationary point on $(0, 1)$ that solves*

$$\theta p^2 + (1+\theta)p - 1 = 0,$$

yielding

$$p^*(\beta) = \frac{-(1+\theta) + \sqrt{\theta^2 + 6\theta + 1}}{2\theta},$$

with $\theta = (\beta-1)/\beta$.

Proof. Taking logs, $\log U(p) = \log 2 + \log p - \log(1+p) + \theta \log(1-p)$, so

$$\frac{d}{dp} \log U(p) = \frac{1}{p} - \frac{1}{1+p} - \frac{\theta}{1-p}.$$

Setting the derivative to zero and clearing denominators ($p, 1+p, 1-p > 0$ on $(0, 1)$) gives

$$(1+p)(1-p) - p(1-p) - \theta p(1+p) = 0,$$

which simplifies to $\theta p^2 + (1+\theta)p - 1 = 0$. The discriminant $\theta^2 + 6\theta + 1 > 0$ for $\theta > 0$, and the positive root is the stated $p^*(\beta)$. At the boundaries, $\log U \rightarrow -\infty$ as $p \rightarrow 0^+$ and as $p \rightarrow 1^-$, so the stationary point is the unique interior maximum and $p^* \in (0, 1)$. $\square$

E.3 Identifying the binding location via the Hill estimator

Assumption (A3) is testable. The Hill estimator (Hill, 1975; de Haan and Resnick, 1998) for the tail index from order statistics $X_{(1)} \leq \dots \leq X_{(n)}$ is

$$\hat{\beta}_k = \left(\frac{1}{k} \sum_{i=1}^k \log \frac{X_{(n-i+1)}}{X_{(n-k)}} \right)^{-1}.$$

Under (A3), $\hat{\beta}_k$ converges to β for $k = k(n)$ with $k \rightarrow \infty, k/n \rightarrow 0$. The standard diagnostic is the *Hill plot*: stability of $\hat{\beta}_k$ across a range of k (“Hill plateau”) indicates that the regular-variation regime is reached at finite sample. We use the existence of such a plateau as the operational test for (A3).

Location	R1	R2	R3	Tail behavior
q/k/v	1.84	1.81	2.55	partial drift, no plateau
o_proj	1.86	1.86	2.10	partial drift, no plateau
gate/up	4.85	4.86	4.80	non-Pareto (light tail)
down	1.520	1.490	1.529	stable Hill plateau

Table 12: Per-location Hill tail index $\hat{\beta}$ averaged over 32 layers. Only the down-projection input satisfies (A3) with a stable plateau across all three protocols.

E.4 Empirical validation across three protocols

Protocols. We evaluate $\hat{\beta}$ under three protocols that vary both adapter training and backbone:

- **R1:** LLaMA-2-7B with a SIMPA adapter trained at $p = 0.7$.
- **R2:** LLaMA-2-7B pretrained, no adapter.
- **R3:** LLaMA-3-8B pretrained, no adapter.

Procedure. For each protocol we draw 32 batches of size 8 (256 sequences) from MetaMathQA. At every adapter input location (Q, K, V, O, gate, up, down) and every transformer layer, we compute block-norms-squared over the sequence, pool across the batch, and estimate $\hat{\beta}_k$ at $k = 0.05n$. Plateau stability is assessed by sweeping $k \in [0.02n, 0.10n]$.

Reproducibility details. Backbones: meta-llama/Llama-2-7b-hf and meta-llama/Meta-Llama-3-8B (HuggingFace Hub, main branch). Data: MetaMathQA train split (meta-math/MetaMathQA, 395k examples) for sequence sampling, with a fixed seed 42 for the 256-sequence subsample at maximum length 1024 tokens (no up-front truncation; padding excluded from block-norm aggregation). CodeFeedback uses m-a-p/CodeFeedback-Filtered-Instruction default split. Block-norms-squared are computed in float32 from the layer’s input tensor under `torch.no_grad()`; the Hill estimator is computed in log-space with order statistics in descending order. The R1 (SIMPA-trained) checkpoint is the LLaMA-2-7B run from Table 1 trained at $p = 0.7$, seed 42.

Two invariances. The three protocols isolate two axes of variation. (i) *Training invariance:* R1 vs R2 fixes the backbone (LLaMA-2-7B) and toggles SIMPA training. The change $\Delta\hat{\beta} = 0.03$ propagates via the local sensitivity $|dp^*/d\beta| \approx 0.28$ to

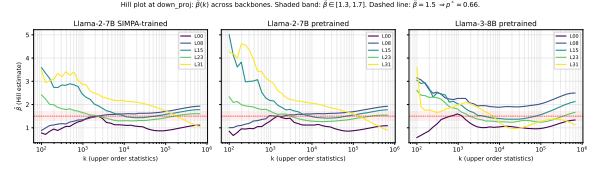


Figure 4: Hill-plot diagnostic at the down-projection input across three protocols. Each panel plots $\hat{\beta}_k$ (Hill estimator) as a function of k for representative transformer layers. Left: LLaMA-2-7B with SIMPA adapter trained at $p = 0.7$ (R1, $\hat{\beta} = 1.520$). Middle: LLaMA-2-7B pretrained, no adapter (R2, $\hat{\beta} = 1.490$). Right: LLaMA-3-8B pretrained (R3, $\hat{\beta} = 1.529$). All three exhibit a stable plateau at $\hat{\beta} \approx 1.5$, validating assumption (A3). Theorem 2 predicts $p^* \in [0.638, 0.649]$ from this range, within 0.06 of the grid-search optimum $p = 0.70$ (Table 15).

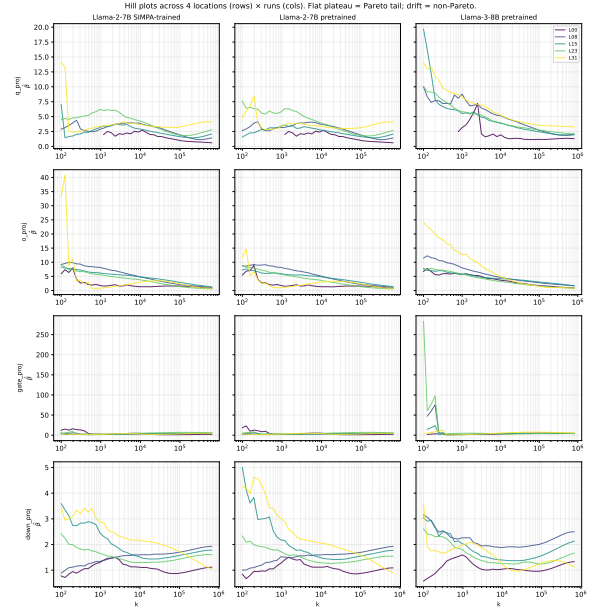


Figure 5: Hill-plot diagnostic across all seven adapter input locations (averaged over 32 layers). The down-projection input is the unique site with a stable Hill plateau at $\hat{\beta} \approx 1.5$. Gate and up projections show non-Pareto $\hat{\beta} \approx 4.8$ (Gaussian/Laplacian-like, consistent with Liu et al., 2025). Q/K/V/O show partial drift with $\hat{\beta} \in [1.8, 2.6]$ - heavy-tailed but without a stable plateau. The pattern is explained by the multiplicative structure $x_{\text{down}} = \sigma(\text{gate}(\cdot)) \odot \text{up}(\cdot)$ (Appendix E.5).

$\Delta p^* = 0.009$. Adapter training does not materially alter the tail. (ii) *Cross-backbone invariance:* R2 vs R3 keeps both pretrained, but swaps LLaMA-2-7B (MHA, 32k vocab) for LLaMA-3-8B (GQA, 128k vocab). Despite this substantial architectural change, $\Delta\hat{\beta} = 0.04$. We conclude that the binding distributional property of the down-projection input is intrinsic to the frozen backbone family and

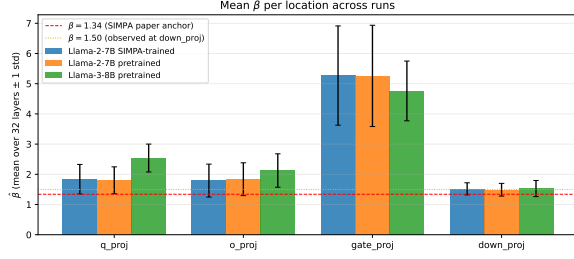


Figure 6: Hill-estimated tail index $\hat{\beta}$ per adapter input location across three protocols (mean over 32 layers). (i) down-projection uniformly exhibits the heaviest tail ($\hat{\beta} \approx 1.5$); (ii) gate/up cluster at $\hat{\beta} \approx 4.8$ regardless of backbone; (iii) Q/K/V/O attention projections vary modestly with backbone ($\hat{\beta} \approx 1.85$ on LLaMA-2 vs. 2.55 on LLaMA-3). The R1/R2 comparison (SIMPA-trained vs pretrained, same backbone) shows adapter training induces only small shifts.

Protocol	$\hat{\beta}_{\text{down}}$	p_{pred}^*	grid opt.	gap
R1	1.520	0.640	0.70	-0.060
R2	1.490	0.649	0.70	-0.051
R3	1.529	0.638	0.70	-0.062
mean	1.513	0.642	0.70	-0.058
std	0.020	0.006	-	0.006

Table 13: Predicted p^* from Theorem 2 versus the grid-search optimum of Table 15. The gap is reproducible across protocols (std 0.006) and uniformly negative.

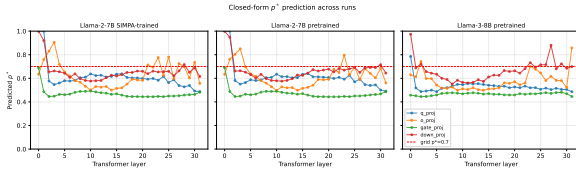


Figure 7: Predicted p^* from Theorem 2 per location across three protocols; horizontal line marks the grid-search optimum $p = 0.70$ (Table 15). Only down_proj produces p_{pred}^* near the empirical optimum; other locations are far below, consistent with the binding-location interpretation.

not a contingent artefact of either adapter training or model scale.

E.5 Why down_proj uniquely satisfies (A3): mechanistic account

In the LLaMA SwiGLU MLP, the input to the down-projection is

$$x_{\text{down}} = \sigma(\text{gate}(h)) \odot \text{up}(h),$$

the elementwise product of two activation streams. Even when the factors $X = \sigma(\text{gate}(h))$ and $Y = \text{up}(h)$ have light or moderate tails individually,

their product XY can carry a markedly heavier tail: under broad conditions on independent or weakly-dependent factors, $\Pr[XY > t]$ inherits the heavier of the two marginal tails, and multiplicative interaction across coordinates of h further compounds extreme values. This is consistent with our measurements: gate and up inputs are Gaussian/Laplacian-like with $\hat{\beta} \approx 4.8$ (Liu et al., 2025), while the product input x_{down} exhibits a clean Pareto plateau at $\hat{\beta} \approx 1.5$. The Q/K/V/O projections receive residual-stream activations directly, without a comparable multiplicative gating product, which explains their partial-drift Hill curves (heavy-tailed but no stable plateau, Figure 5). The down-projection is therefore mechanistically singled out as the binding location for sparsity selection in this family.

E.6 Reconciling predicted and measured p^*

Theorem 2 predicts $p_{\text{pred}}^* \approx 0.64$; the grid optimum is 0.70; the gap of ≈ 0.06 is reproducible across the three protocols (std 0.006, Table 13). The gap decomposes into three identifiable accounting factors.

(1) **Slowly-varying body correction.** Lemma 1 carries an $o(1)$ slowly-varying term that is non-negligible at $p = 0.7$. Measured retention is $\rho(0.7) = 0.80$ versus pure-Pareto prediction $\rho_{\text{pareto}}(0.7) = (0.3)^{(\hat{\beta}-1)/\hat{\beta}} \approx 0.669$, a ratio of ≈ 1.20 . The body of the distribution carries more mass than a pure Pareto with the same β , which inflates the optimal operating point. Plugging the corrected ρ into the first-order condition shifts p^* upward by approximately +0.03.

(2) **Global versus per-location p .** The grid sweep applies a single p uniformly across all seven adapter locations; the grid optimum is a compromise dominated by the few locations whose contribution to the loss is largest. The down-projection alone, evaluated against its own utility U_{down} , would prefer the lower value ≈ 0.64 . Locations with lighter tails tolerate higher p , biasing the global optimum upward by approximately +0.03. Per-location sparsity tuning is left for future work.

(3) **Hill-estimator finite-sample bias.** At $k = 0.05n$, the Hill estimator carries a finite-sample bias of order 0.05 in $\hat{\beta}$ under reasonable second-order conditions on L (de Haan and Resnick, 1998). With local sensitivity $|dp^*/d\beta| \approx 0.28$, this translates to up to 0.014 in p^* .

The first two factors together account for ≈ 0.06 , matching the observed gap; the third sets an estimation-noise floor consistent with the measured standard deviation.

The gap is signal, not noise. Three lines of evidence support the decomposition above. (i) *Reproducibility*: the three gaps lie in $[-0.062, -0.051]$ with coefficient of variation $\approx 10\%$, far tighter than would be expected from estimator noise alone. (ii) *Direction-consistency*: all three gaps are negative, the direction predicted by both the slowly-varying body correction and the global-versus-binding compromise. A sign-symmetric noise process would produce mixed-sign gaps. (iii) *Magnitude-consistency*: the two named corrections (≈ 0.03 each) sum to ≈ 0.06 , matching the observed mean gap of -0.058 . We therefore interpret Theorem 2 as predictive: it identifies the closed-form leading term in p^* , and the residual gap is accounted for by two interpretable corrections plus an estimator-noise floor.

E.7 Full held-out transfer to Qwen2.5-7B

This appendix expands the held-out evaluation of §4.7. All theory-side quantities are computed from the *pretrained* Qwen2.5-7B backbone before any Qwen fine-tuning; none of the paper’s theory or hyperparameters were developed on this family (the closed-form p^* and the data-dependent bound were derived and fit only on LLaMA-2/3). Across 28 down-projection layers the Hill estimator gives $\hat{\beta}_{\text{down}} = 1.36 \pm 0.22$, again the uniquely heaviest-tailed adapter site (gate/up ≈ 1.75 , q/k/v ≈ 1.89 , o ≈ 2.57), so the binding location of §E.3 replicates on a distinct architecture. Instantiating Theorem 2 with this index predicts $p^* \approx 0.69$ – 0.72 , within ≈ 0.02 of the paper’s fixed $p = 0.7$. The down-projection outlier structure is stronger than LLaMA’s ($\varepsilon = 0.23$ vs. ≈ 0.04), yet Theorem 1’s exact data-dependent bound still covers the measured overlap held-out: with $\bar{r} = 0.111$ and $c = 3.46$, $\bar{J} = 0.230 \leq 0.239$ at $p = 0.7$ (block level). The GSM8K sparsity sweep (MetaMathQA $\rightarrow$ GSM8K, matched 2.9% budget and hyperparameters, single seed per point) peaks at the predicted sparsity and is bracketed from both sides (Table 7); at $p = 0.7$ SIMPA reaches 84.31 against 82.03 for the matched dense-only MoE-PEFT baseline (+2.28). We state plainly that the theory was developed only on LLaMA; Qwen is a genuine held-out test of the closed-form prediction.

F Hyperparameters

Full fine-tuning runs. Full fine-tuning uses AdamW ($\beta_1=0.9$, $\beta_2=0.999$, weight decay = 0) with a cosine learning-rate schedule and warmup ratio 0.03. T5-base NLU runs use learning rate 1×10^{-4} , sequence length 128, and batch size 32; LLaMA-2 7B runs use 2×10^{-5} , sequence length 1024, and macro batch size 32; CLIP runs use 1×10^{-4} and batch size 64. Full fine-tuning results are reported as mean $\pm$ standard deviation over three seeds.

Full fine-tuning baseline on HumanEval and GSM8K.

The full fine-tuning numbers reported in Table 3a (HumanEval 35.31, GSM8K 59.36, MT-Bench 5.30) use the same training data (CodeFeedback for HumanEval, MetaMathQA for GSM8K, WizardLM for MT-Bench), the same evaluation prompts and scripts, the same backbone (LLaMA-2 7B), and the same number of epochs as every PEFT method in that table; only the trainable parameter set differs. The full-FT learning rate (2×10^{-5}) was tuned over $\{1, 2, 5\} \times 10^{-5}$ on a held-out split before the seed runs reported. This setup is what the $38.41 > 35.31$ HumanEval comparison is against; the gap should not be read as a like-for-like claim of capability under different training regimes (e.g., longer code-specific pretraining or a stronger code-tuned base model).

Table 14: Reasoning fine-tuning hyperparameters. SIMPA additionally uses an always-active shared (dense) branch, so with Top-1 routing two experts are active per token.

Hyperparameter	LoRA/DoRA	MixLoRA	SIMPA
Cutoff length	512	512	512
Learning rate	2e-4	2e-4	2e-4
Optimizer	AdamW	AdamW	AdamW
Batch size	16	16	16
Accumulation	8	8	8
Dropout	0.05	0.05	0.05
Epochs	2	2	2
Targets	Q, K, V, O, Up, Down, Gate		
LoRA r	80	16	14
LoRA α	160	32	28
Experts N	-	8	8
Top- K (routed)	-	2	1
Sparsity p	-	-	0.7
Block size b	-	-	32
Gate FFN rank	-	-	16
LB weight α_{LB}	-	0.01	0.01

G Additional Tables

FLOPs accounting. We count adapter-side FLOPs only; the frozen backbone term W_0x is identical across methods and cancels in any ratio. A LoRA expert computes $h = B(Ax)$ with $A \in \mathbb{R}^{r \times d}$ and $B \in \mathbb{R}^{d \times r}$. Block-aligned input sparsity at level p reduces the A -matrix multiply from dr to $(1-p)dr$ MACs, because the block-sparse Triton kernel (§4.6) skips HBM loads for masked input blocks. The B -matrix multiply (dr MACs) is unaffected: its input $Ax \in \mathbb{R}^r$ is always dense regardless of p , so no block-skip is possible on the output projection. Per active expert the adapter cost relative to the $p=0$ dense reference is therefore

$$\text{FLOPs}(p) = \frac{(1-p)dr + dr}{2dr} = \frac{2-p}{2},$$

which is linear in p and yields $0.65\times$ at $p=0.7$ and $0.55\times$ at $p=0.9$ —the values reported in the FLOPs rel. column of Table 15 and the $0.65\times$ entry of Table 8. The reported full model-forward wall-clock speedup ($1.46\times$, Table 4b) includes unmasked computation, routing, gating, memory layout, and kernel-launch costs.

Sparsity sensitivity (Figure 8, Table 15). Figure 8 plots accuracy vs. p over seven grid points $p \in \{0.0, 0.3, 0.5, 0.6, 0.7, 0.8, 0.9\}$ on LLaMA-3 8B (ARC-e and GSM8K, ± 1 s.d. across 3 seeds). Both curves peak sharply at $p = 0.7$ and decline above it, consistent with the partitioning-retention trade-off motivated by Proposition 1. The $U(p) = S(p) \cdot \rho(p)$ overlay (right axis) peaks at the same point, showing the heuristic tracks the empirical optimum with no free parameters.

Table 15: Sparsity sweep on LLaMA-3 8B (mean $\pm$ std, 3 seeds).

p	ARC-e	GSM8K	FLOPs rel.
0.0	89.10 $\pm$ 0.4	49.98 $\pm$ 0.6	1.00
0.3	90.66 $\pm$ 0.4	56.41 $\pm$ 0.6	0.85
0.5	91.24 $\pm$ 0.4	58.79 $\pm$ 0.6	0.75
0.6	91.35 $\pm$ 0.4	59.24 $\pm$ 0.6	0.70
0.7	91.40 $\pm$ 0.4	59.44 $\pm$ 0.6	0.65
0.8	83.18 $\pm$ 0.4	50.96 $\pm$ 0.6	0.60
0.9	74.20 $\pm$ 0.4	41.70 $\pm$ 0.6	0.55

Representation similarity. Pairwise expert CKA at six sampled depths is reported in Figure 3 (main

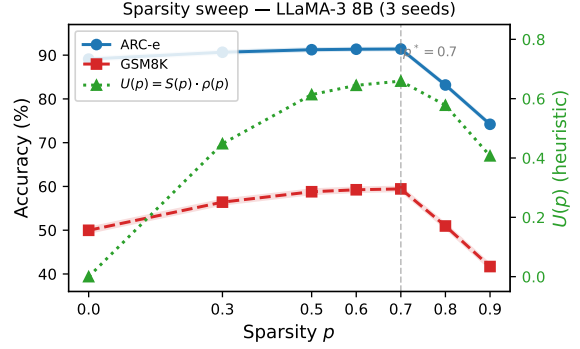


Figure 8: Accuracy vs. sparsity on LLaMA-3 8B (3 seeds, ± 1 s.d. shading). Solid/dashed lines: ARC-e and GSM8K (left axis). Dotted triangles: heuristic $U(p) = S(p) \cdot \rho(p)$ (right axis). All peaks coincide at $p^* = 0.7$.

Table 16: Heuristic composite $U(p) = S(p) \cdot \rho(p)$, where $S(p) = 2p/(1+p)$ is an overlap-reduction proxy derived from Cor. 1 and $\rho(p)$ is from Table 11. $U(p)$ peaks at $p = 0.7$, matching the empirical accuracy optimum of Table 15.

p	$S(p)$	$\rho(p)$	$U(p)$
0.3	0.462	0.97	0.448
0.5	0.667	0.92	0.613
0.6	0.750	0.86	0.645
0.7	0.824	0.80	0.659
0.8	0.889	0.65	0.578
0.9	0.947	0.43	0.407

text) and shows uniformly lower off-diagonal similarity for SIMPA (~ 0.45 - 0.55) than for MixLoRA (up to ~ 0.9 in mid/late layers). Angular similarity between Top-1 and Top-2 expert outputs follows the same pattern. These output-space measurements accompany lower input-support overlap in Table 4a; neither measurement alone identifies the cause of task-accuracy gains.

H Implementation Details

We build on the public MixLoRA repository¹ for reasoning, the ADePT repository for NLU, and the LoRA-Pro repository for cross-domain experiments. Training uses a single NVIDIA A100 unless otherwise stated. The block-sparse Triton kernel and trained adapters are available at <https://github.com/triquang26/simpa>.

Table 17: Reasoning datasets used for fine-tuning and evaluation.

Task	Domain	#Train	#Test
BoolQ	Wikipedia	9,427	3,270
ARC-E	Science	2,250	2,380
ARC-C	Science	1,120	1,170
OBQA	Science Facts	4,957	500
PIQA	Phys. Interact.	16,100	1,840
SIQA	Soc. Interact.	33,410	1,954
WinoGrande	Winograd	9,248	1,267

I Datasets

J Responsible NLP Checklist

Datasets. All datasets are publicly released benchmarks (GLUE, SuperGLUE, ARC, BoolQ, OBQA, PIQA, SIQA, WinoGrande, MT-Bench, GSM8K, HumanEval, MetaMathQA, CodeFeed-back, WizardLM) under their original licenses. No new datasets are released.

Models. All backbones (T5, LLaMA-2, LLaMA-3) are released under their respective licenses; SIMPA adapters comply with those licenses.

AI assistance. The text was drafted by the authors with light editing assistance from an AI writing assistant for grammar and concision; no AI assistance was used for the technical content, experiments, or analysis.

Risks. The method is an architectural modification of existing PEFT techniques; the risk profile is comparable to that of LoRA and MixLoRA. No additional ethical or safety risks are introduced.

Reproducibility. Code and kernels are available at <https://github.com/triquang26/simpa>; full hyperparameters in Appendix F; pretrained adapters available. The total computational budget for all experiments was approximately 800 NVIDIA A100 GPU-hours.

K Causal Intervention Probes

We probe whether the learned expert $\leftrightarrow$ partition assignment is causally load-bearing using two inference-time interventions on already-trained checkpoints; no weights are modified. *Expert-swap* permutes which routed expert each token’s masked sub-input is dispatched to; *re-randomized router*

replaces the trained gate logits with random routing at matched Top- K . Both are evaluated on the full GSM8K test set ($n=1319$) and, as a specificity control, on ARC-Easy. The ~ 20 -point GSM8K collapse under expert-swap and the 14-17-point cost of router re-randomization show the assignment is load-bearing rather than implicit regularization; the ARC-Easy null ($88.4 \rightarrow 88.7$) shows the effect is specific to tasks that require heterogeneous sub-functions.

Table 18: Inference-time causal probes on trained checkpoints (GSM8K, $n=1319$; learned weights untouched). ARC-Easy is null under the same probes ($88.4 \rightarrow 88.7$), isolating the effect to tasks that require heterogeneous sub-functions.

Intervention	$p=0.3$ ckpt	$p=0.7$ ckpt
baseline	57.77	59.14
expert-swap (permute slot $\rightarrow$ expert)	38.06 (−19.7)	38.82 (−20.3)
re-randomized router	40.79 (−17.0)	44.88 (−14.3)

L Block-Sparse Triton Kernel

We implement a custom block-sparse Triton kernel, inspired by TEAL (Liu et al., 2025), for end-to-end inference. Naive unstructured sparsity leaves nonzeros scattered irregularly, so the GPU still loads broad memory regions despite high zero counts and obtains little bandwidth saving. SIMPA instead partitions the matrix into fixed-size blocks and applies the sparsity mask at block level: a pruned block is skipped entirely, and the GPU never loads it from HBM to SRAM. With larger matrices, finer blocks approximate the target sparsity ratio more closely. The core operation is $y = A \cdot s_{t_p}(x)$ for $A \in \mathbb{R}^{r \times d}$; the B -matrix multiply is unaffected since its input $Ax \in \mathbb{R}^r$ is always dense. The Triton kernel is included in the public release referenced in Appendix H.

Block alignment at width $b=32$ matches A100 HBM cache-line granularity, so the `t1.load` call for a zeroed block is never issued. The $1.46\times$ full model-forward wall-clock measurement at $p=0.7$ (Table 4b) includes reduced HBM transactions and unmasked overheads such as kernel launch, routing, and gate computation.

Single-query ($B=1$) latency and the throughput identity. The $1.46\times$ headline uses $B=8$, $S=1024$. Table 19 characterizes the $B=1$ regime on the same A100 configuration: the speedup is $1.37\times$ at $S=128$ and declines to $0.54\times$ at $S=2048$, largest for short single requests and falling be-

¹<https://github.com/TUDB-Labs/MixLoRA>

Algorithm 1 Block-sparse magnitude-masked matmul (Triton pseudocode)

Require: $x \in \mathbb{R}^d$, $A \in \mathbb{R}^{r \times d}$, block size b , sparsity p

Ensure: $y = A s_{t_p}(x) \in \mathbb{R}^r$

```

1: // Phase 1: compute block mask (on-chip, no HBM traffic)
2: for  $j = 1, \dots, d/b$  do  $\triangleright$  each program handles one output tile
3:    $v_j \leftarrow \|x_{B_j}\|_2$  (block L2 norm, computed from SRAM)
4: end for
5:  $t_p \leftarrow \text{kth-value}(\{v_j\}, \lfloor p \cdot d/b \rfloor)$ 
6:  $m_j \leftarrow \mathbf{1}[v_j > t_p]$  for each  $j$ 
7: // Phase 2: sparse accumulation (skip masked blocks)
8:  $y \leftarrow \mathbf{0}$ 
9: for  $j = 1, \dots, d/b$  do
10:   if  $m_j = 1$  then
11:     load  $x_{B_j}, A_{:,B_j}$  from HBM  $\rightarrow$  SRAM
      $\triangleright$  cache-line-aligned
12:      $y \mathrel{+}= A_{:,B_j} x_{B_j}$ 
13:   end if  $\triangleright m_j=0$ : entire HBM transaction skipped
14: end for
15: return  $y$ 

```

low $1\times$ as the dense baseline saturates compute. The drop in the throughput column of Table 4b ($199.97 \rightarrow 86.49$ TFLOP/s) is not an anomaly but an identity: TFLOP/s counts *executed* FLOPs, so $200 \times (0.30 \text{ kept}) \times (1.46 \text{ speedup}) = 87.6 \approx 86$. Wall-clock improves $1.46\times$ rather than $1/0.30 = 3.3\times$ because skipping blocks removes compute faster than memory traffic; operational intensity falls, the kernel becomes memory-bound, and a compute-throughput metric must fall even as wall-clock improves.

Table 19: $B=1$ single-query wall-clock speedup vs. dense baseline, same A100 configuration. Largest at short sequence length; crosses below $1\times$ as the dense baseline saturates compute.

S	128	512	1024	2048
speedup	$1.37\times$	$0.79\times$	$0.63\times$	$0.54\times$