COLLABMASK: IDENTIFYING NEURON COLLABORATION FOR INTERPRETABLE LLM ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The internal interpretability of large language models (LLMs) remains a critical challenge, particularly for understanding how distributed neurons coordinate across tasks. In this work, we identify neuron collaboration, an empirical phenomenon wherein specific neurons form closely connected neighborhoods that are systematically co-activated. We validate this collaborative pattern through two complementary lines of evidence: semantic alignment in independently obtained functional interpretations of collaborating neurons, and selective co-activation on targeted tasks and token types. To apply this structure to adaptation, we propose CollabMask, which generates collaboration-aware gradient masks to induce structured sparsity during continual adaptation. Experimental results show that leveraging this structure improves standard adaptation on mathematical benchmarks by an average of 2.7%. Furthermore, in continual learning scenarios, preserving these collaborative subnetworks helps mitigate catastrophic forgetting, surpassing existing baselines on new tasks while retaining previously acquired knowledge. By connecting neuron collaboration, structured sparsity, and task-specific dynamics, our findings provide an interpretable perspective for understanding LLM adaptation and advance the empirical explainability of multi-task learning.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed across scientific reasoning, programming, education, and other rapidly changing domains (Bubeck et al., 2023; Thirunavukarasu et al., 2023; Wen et al., 2024). This trend raises two coupled challenges. First, the internal interpretability of LLMs remains limited: we still lack simple empirical units for describing how distributed neurons coordinate during task-specific behavior. Second, models must continually adapt to new data and skills, but sequential adaptation often causes catastrophic forgetting, where newly acquired performance comes at the cost of previously learned knowledge and reasoning ability (Shi et al., 2025a; Luo et al., 2025). Understanding internal coordination is therefore not only an analysis problem, but also a prerequisite for more interpretable continual adaptation.

Existing interpretability methods provide valuable views of LLM internals, but their results are often difficult to connect to post-training practice. Neuron-level attribution, probing, graph analysis, and automatic description methods can identify task-associated units or semantic descriptions (Sajjad et al., 2022; Zhao et al., 2023; Yu & Ananiadou, 2024; Oikarinen & Weng, 2024; Foote et al., 2023; Choi et al., 2024). However, these analyses frequently require nontrivial pipelines for collecting activations, estimating attributions, or assigning descriptions, and they are usually presented as post-hoc explanations. Less attention has been paid to whether the resulting interpretable structures can serve as usable units for post-training or continual learning.

Conversely, continual learning methods for LLMs tend to focus on controlling how adaptation changes the model rather than on the model's naturally emerging functional organization. Regularization and model merging constrain parameter drift (Kirkpatrick et al., 2017; Yang et al., 2026); PEFT methods such as OLoRA and OPLoRA encourage modular or orthogonal low-rank updates (Büyükakyüz, 2024; Xiong & Xie, 2026); and gradient-sparse methods reduce update density (Xu et al., 2021; Hui et al., 2024; Li et al., 2025a; Wang et al., 2022b). These strategies are useful for stability, but they typically isolate, orthogonalize, or sparsify updates without explicitly asking whether LLM neurons already form task-conditioned functional structures that can guide continual adap-

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

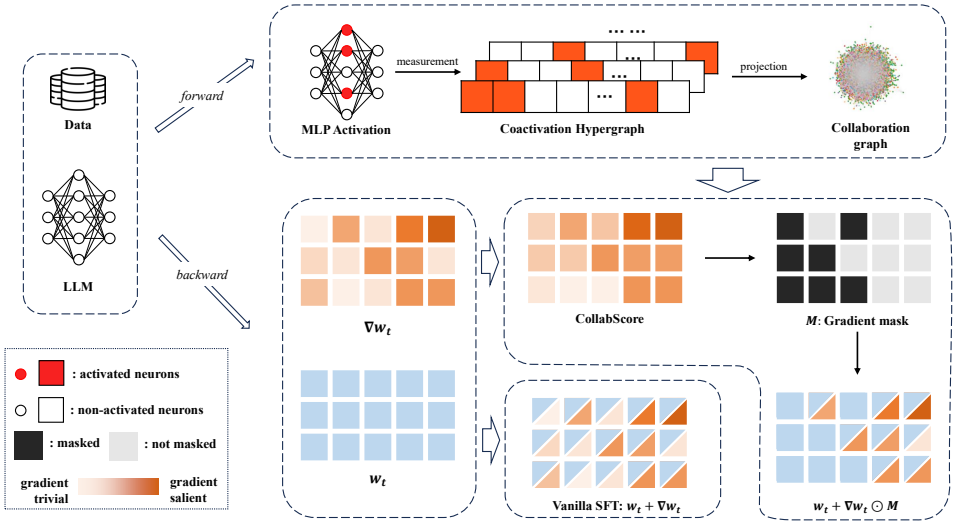


Figure 1: `CollabMask` is an interpretability-guided adaptation framework with two phases, matching the figure labels. **Phase 1: Identifying collaboration structure in the forward pass.** Training samples are fed through the LLM, MLP neuron co-activation is measured for each layer, and token-induced co-activation hypergraphs are projected into weighted collaboration graphs. These graphs serve as empirical summaries of which neurons tend to activate together under the measured task-token regime. **Phase 2: Collaboration-guided adaptation in the backward pass.** During fine-tuning, the stored collaboration graph is combined with the current parameter gradients $\nabla \boldsymbol{w_t}$ to compute CollabScore, which estimates the relevance of each parameter update at step t. The top-$p\%$ CollabScore entries form the gradient mask $\boldsymbol{M}$, enabling selective updates to functionally related parameters and reducing redundant updates relative to vanilla SFT. Gains over matched masking controls provide adaptation-level support for the usefulness of the collaboration structure, not as causal proof of a complete mechanism.

tation. Recent explanation-guided approaches, such as FVG (Jiang et al., 2025a), begin to bridge this gap, but they primarily target attention-derived signals and leave MLP neuron collaboration less explored.

In this work, we introduce *neuron collaboration* as an empirical unit for studying and using MLP structure in LLM adaptation. Our method has two phases. First, we measure task-conditioned neuron co-activation from forward passes, construct token-induced hypergraphs, and project them into weighted collaboration graphs whose edges summarize which neurons tend to activate together. We validate these graphs independently of adaptation performance: high-weight edges connect neurons with more similar external descriptions, and held-out task-token co-activations receive larger graph weights than non-calculation and off-topic controls. These results support collaboration graphs as interpretable summaries of task-conditioned neuron organization.

We then propose `CollabMask` as a direct application of collaboration graphs to continual adaptation. During fine-tuning, `CollabMask` combines the stored collaboration graph with the current batch gradient to compute CollabScore and select a top-$p\%$ gradient mask, concentrating updates on parameters associated with collaborating neurons while inducing structured sparsity. Across mathematical reasoning and math-to-code continual adaptation, this collaboration-guided update rule improves aggregate math accuracy over SFT and gradient-masking controls, and yields competitive retention–adaptation trade-offs in sequential experiments. The performance gains do not constitute a causal circuit explanation, but they show that the identified collaboration structure is useful for adaptation.

Our contributions can be summarized as follows:

1. We identify an empirical phenomenon termed *neuron collaboration* in LLMs and validate it through semantic alignment and task-token specific co-activation.
2. We propose `CollabMask`, which leverages collaboration graphs to induce structured sparsity during continual adaptation.
3. We show that the adaptation gains produced by `CollabMask`, including a 2.7% average accuracy improvement and mitigated forgetting, provide empirical evidence for the functional significance of neuron collaboration.

2 Related Work

Neuron-level interpretability. Neuron-level studies characterize individual units and their associations with model behavior (Sajjad et al., 2022; Zhao et al., 2023). Attribution methods identify neurons associated with particular tasks or knowledge retrieval (Yu & Ananiadou, 2024; Lan et al., 2023; Wang et al., 2022a), while linear probes (Oikarinen & Weng, 2024), graph-based analyses (Foote et al., 2023), and automated descriptions (Choi et al., 2024) provide statistical or semantic views of internal representations. Extending analysis beyond isolated units, La Rosa et al. (2023) study multi-level activation patterns. Interpretability signals have also been used to guide adaptation (Yin et al., 2024; Li et al., 2025b; Xu et al., 2025), but are usually derived from individual neurons or attention modules. Our work instead identifies task-conditioned co-activation among MLP neurons and tests whether independently obtained neuron descriptions align with the resulting collaboration structure. This is an empirical representational analysis; it does not claim to recover a causal computation graph.

Continual adaptation. Continual learning studies how models acquire a sequence of capabilities while retaining prior knowledge (Wang et al., 2024a; Shi et al., 2025b; French, 1999; Luo et al., 2025). Approaches for LLMs include regularizing task-relevant parameters, as in EWC (Kirkpatrick et al., 2017); modular or orthogonal adapters, including OLoRA (Büyükakyüz, 2024) and OPLoRA (Xiong & Xie, 2026); model merging (Yang et al., 2026); and interpretability-guided updates such as FVG (Jiang et al., 2025b). These methods provide complementary stability–plasticity trade-offs, but do not directly test whether task-conditioned collaboration among MLP neurons can serve as an informative structure during adaptation.

Gradient sparsity. Adapting large models with limited data can produce redundant updates and distort pretrained representations (Kumar et al., 2022; Zhang et al., 2024). LoRA (Hu et al., 2021) restricts adaptation to low-rank subspaces. Child-Tuning (Xu et al., 2021) and HFT (Hui et al., 2024) randomly mask gradients, whereas GMT (Li et al., 2025a) selects coordinates using accumulated gradient magnitude; SparCL (Wang et al., 2022b) studies sparse updates in continual settings. These techniques motivate gradient sparsity, but their selection rules do not incorporate task-conditioned MLP co-activation. `CollabMask` uses that statistical structure to guide sparse adaptation: if collaboration-aware masks repeatedly yield gains over appropriate masking controls, the result provides empirical support for the usefulness of the identified structure.

3 Identifying Neuron Collaboration

We use *neuron collaboration* to denote an empirical pattern in which specific MLP neurons have closely connected neighbors under a task-conditioned co-activation statistic. The pattern is identified from forward-pass observations; the term does not imply that an edge is causal or that the graph is a complete account of the model's computation. Figure 1 summarizes the identification protocol, and Algorithm 1 gives its operational form.

3.1 Collaboration Graph Construction

For each MLP layer ℓ, we identify candidate collaboration in two stages: we construct a co-activation hypergraph from forward passes over $\mathcal{D}$ and project it to a weighted undirected graph. This protocol produces a reproducible statistic for testing semantic and adaptation predictions in Sections 4 and 5.

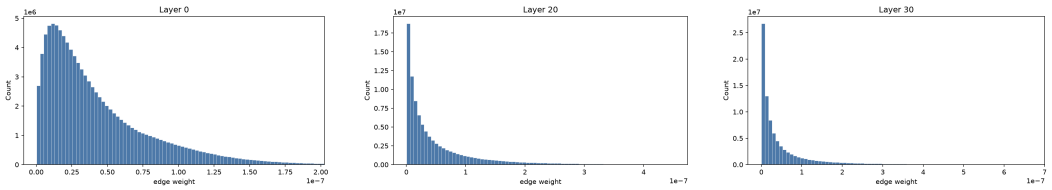


Figure 2: Layer-wise distributions of projected collaboration edge weights. The heavy-tailed structure motivates validating not only graph neighborhoods but also whether the strongest weighted edges correspond to semantically aligned neuron pairs.

3.1.1 NEURON ACTIVATION INDICATOR

Consider MLP layer ℓ with N_ℓ neurons. For token $x \in \mathcal{D}$, let $act_{\ell,n}(x)$ denote the n-th output of the gate projection after the nonlinear activation (e.g., SiLU in LLaMA). This differs from the conventional signed neuron output $act_{\ell,n}(x) \cdot up_{\ell,n}(x)$, where $up_{\ell,n}(x)$ is the corresponding up-projection component. We use $act_{\ell,n}(x)$ only as a binary activation detector: any neuron with a non-zero contribution yields $act_{\ell,n}(x) > 0$ regardless of sign. We define

$$I_{\ell,n}(x) = \mathbf{1}\big[act_{\ell,n}(x) > 0\big].$$

The indicator is a measurement convention rather than a claim that positive gate output uniquely determines a neuron's functional contribution. Appendix A.5 reports empirical activation statistics and distributional evidence supporting this detector.

3.1.2 CO-ACTIVATION HYPERGRAPH

For each layer ℓ, we define a hypergraph $\mathcal{H}_\ell = (V_\ell, \mathcal{E}_\ell)$ with vertex set $V_\ell = \{1, \ldots, N_\ell\}$ and one hyperedge per token:

$$E_\ell^{(h)}(x) = \big\{n \in V_\ell \mid I_{\ell,n}(x) = 1\big\},$$
$$\mathcal{E}_\ell = \big\{E_\ell^{(h)}(x) \mid x \in \mathcal{D}\big\}.$$

Each hyperedge records the set of neurons co-activated by a single token.

3.1.3 PROJECTION TO COLLABORATION GRAPH

Following Li & Milenkovic (2017), we project $\mathcal{H}_\ell$ onto a weighted undirected graph $G_\ell = (V_\ell, E_\ell)$ with edge weights

$$w_\ell(i,j) = \sum_{x \in \mathcal{D}} \frac{\mathbf{1}\big[i,j \in E_\ell^{(h)}(x)\big]}{|E_\ell^{(h)}(x)|^\alpha} TR(x), \tag{1}$$

where $\mathbf{1}[\cdot]$ is the indicator function, $|E_\ell^{(h)}(x)|$ is the hyperedge size, α controls the relative weight of large vs. small hyperedges, and $TR(x)$ is a token reweighting factor. Intuitively, $w_\ell(i,j)$ accumulates evidence that neurons i and j fire together, down-weighting large hyperedges (which co-occur by chance) and frequent but uninformative tokens.

Figure 2 shows representative edge-weight distributions after projection. The weights are strongly skewed rather than uniform, so the highest-weight edges form a natural target for the semantic validation in Section 4.1.

Hyperedge degree reweighting. The exponent α trades off broad vs. localized co-activation: values $0 < \alpha < 2$ emphasize high-degree hyperedges, while $\alpha > 2$ favors small, specific co-activation patterns. We set $\alpha = 2.5$ to prioritize tighter patterns while retaining broader context; Appendix A.3 provides the theoretical motivation.

Token frequency reweighting. Naive summation over tokens overweights high-frequency symbols (e.g., `eos_token`) that dominate co-activation counts but carry little task-specific signal.

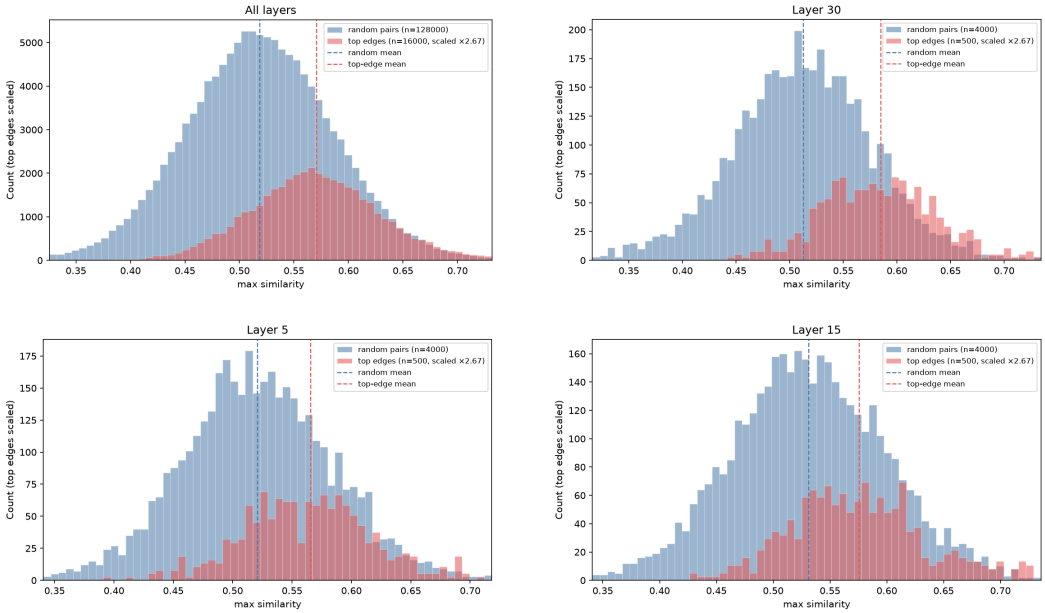


Figure 3: Semantic-similarity distributions for top collaboration edges versus random within-layer neuron pairs. High-weight collaboration edges are shifted toward larger neuron-description similarity, both when pooling all layers and in representative individual layers.

Inspired by Zipf's law (Zipf, 1949), we rank tokens by ascending frequency in $\mathcal{D}$ and define $TR(x) = \mathrm{rank}(x)^{\beta}$, with $\beta = 1.3$ in our experiments. Medium-frequency tokens thus contribute more than very frequent ones; details are in Appendix A.6.

4 Validating Neuron Collaboration

In this section, we validate the collaboration phenomenon through two lines of evidence: semantic alignment and task-token specific co-activation. Unless otherwise noted, graphs are estimated on LLaMA 3.1–8B–Instruct from GSM8K (Cobbe et al., 2021) calculation spans, i.e., answer tokens enclosed by <<...>>. This isolates arithmetic reasoning tokens from surrounding natural-language explanation and formatting; Appendix A.1 gives a concrete example.

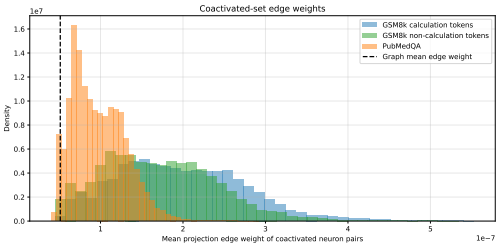
4.1 Semantic Alignment

We first ask whether high-weight collaboration edges connect neurons with similar independently assigned functional descriptions. Using the neuron-description database of Choi et al. (2024), we compare the strongest within-layer collaboration edges against uniformly sampled within-layer neuron pairs via description-embedding similarity. The full scoring protocol, sampling design, and hypothesis tests are given in Appendix A.7.

Figure 3 shows that top collaboration edges have higher semantic similarity than random pairs (KS D=0.3252; Cliff's δ=0.4347). This indicates that neurons with stronger collaboration tend to have more similar functional meanings, as reflected by their external descriptions. The result is therefore evidence that the collaboration graph provides a valid interpretable perspective on neuron organization.

4.2 Task-Token Specific Co-activation

We next evaluate whether the collaboration graph is specific to the token regime from which it was estimated. For each co-activated neuron set, we compute its mean pairwise edge weight in the stored graph and compare held-out GSM8K calculation tokens, held-out GSM8K non-calculation



Condition	Mean	Std
GSM8K calc.	1.98×10^{-7}	7.80×10^{-8}
GSM8K non-calc.	1.71×10^{-7}	6.92×10^{-8}
PubMedQA	9.74×10^{-8}	3.24×10^{-8}
Graph-wide avg.	5.12×10^{-8}	–

Figure 4: *(Left)* Distributions of mean collaboration-edge weights for co-activated neuron sets. *(Right)* Summary statistics for held-out GSM8K calculation tokens, GSM8K non-calculation tokens, PubMedQA, and the graph-wide average edge weight.

tokens (Cobbe et al., 2021), and answer tokens from PubMedQA (Jin et al., 2019), an unrelated biomedical-domain dataset, with the graph-wide mean edge weight as a reference. A calculation-specific graph should assign the largest weights to held-out calculation tokens, lower weights to in-domain but non-calculation tokens, and still lower weights to the off-topic control.

Figure 4 shows the expected ordering: held-out calculation tokens have the largest average edge weight, followed by GSM8K non-calculation tokens, PubMedQA, and the graph-wide reference. Notably, even the unrelated PubMedQA tokens have higher average co-activated edge weight than the graph-wide mean. This suggests that the collaboration graph captures both topic- and token-type-specific relations and more general co-activation patterns shared across text domains.

5 Collaboration-Guided Continual Adaptation

We use the validated collaboration graph to guide gradient updates during continual adaptation. The goal is to concentrate a fixed update budget on parameters associated with task-relevant co-activating neurons. This makes `CollabMask` a direct application of collaboration graphs to adaptation, with performance against matched masking controls serving as supporting evidence for the graph's usefulness.

5.1 Collaboration-Aware Gradient Masking

We build on masked SGD (Xu et al., 2021; Hui et al., 2024; Li et al., 2025a). Given mini-batch B_t, a binary mask $\boldsymbol{M}_t$ restricts the update:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \nabla_{\boldsymbol{w}_t} \mathcal{L}(B_t) \odot \boldsymbol{M}_t. \tag{2}$$

For every MLP parameter block[1], `CollabMask` computes gradient magnitude, modulates it with the identified collaboration graph, and retains the top-$p\%$ coordinates.

Collaboration modulation. For the graph G_ℓ associated with parameter block $\boldsymbol{w}_t$, we retain each neuron's top-k neighbors, row-normalize the result, and align it with parameter coordinates:

$$\boldsymbol{A}(\boldsymbol{w}_t) = \left|\nabla_{\boldsymbol{w}_t} \mathcal{L}\right| \odot \mathrm{Norm}\big(k\mathrm{NN}(G_\ell)\big).$$

The resulting score is

$$\text{Collab-S}(\boldsymbol{w}_t) = (1-\lambda)\left|\nabla_{\boldsymbol{w}_t} \mathcal{L}\right| + \lambda \boldsymbol{A}(\boldsymbol{w}_t).$$

We set $\lambda = 0.5$ and assign mask value 1 to the top-$p\%$ coordinates. The current gradient makes the mask batch-adaptive, while the measured collaboration graph remains fixed. Algorithm 1 summarizes both graph construction and mask application.

6 Empirical Evaluation

We evaluate whether collaboration-aware masks improve adaptation under matched update budgets. Single-task results test mathematical adaptation, and sequential math-to-code results test retention and plasticity. Component-disruption ablations are reported in Appendix A.2.

[1]We mask `up_proj`, `gate_proj`, and `down_proj` independently in LLaMA-style MLPs.

6.1 EXPERIMENTAL SETUP

Datasets. We use GSM8K (Cobbe et al., 2021) as the training corpus and evaluation benchmark for mathematical reasoning. In the continual setting, models adapt sequentially to GSM8K and OpenCodeInstruct (Ahmad et al., 2025). We assess coding with EvalPlus (Liu et al., 2023) on HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021). MMLU-Pro (Wang et al., 2024b) and contextual PubMedQA (Jin et al., 2019), neither used for adaptation, provide additional retention and generalization measurements.

Base Models. We evaluate pretrained language models from three widely adopted families: LLaMA (Grattafiori et al., 2024), Mistral (Jiang et al., 2023), and Qwen (Team, 2025). For the mathematical reasoning benchmark, we employ LLaMA-3.1-8B, Mistral-v0.1-7B, and the base variants of Qwen3-4B and Qwen3-8B. The continual adaptation experiments are conducted on LLaMA-3.1-8B and Mistral-v0.1-7B. All models are initialized from their official pretrained checkpoints without any prior instruction tuning or task-specific adaptation.

Baselines. We compare `CollabMask` against the following baselines grouped into three categories:

- **Standard & Parameter-Efficient Tuning. SFT** denotes standard full-parameter supervised fine-tuning. **LoRA** (Hu et al., 2021) injects trainable low-rank matrices into linear layers.
- **Gradient-Masked Fine-Tuning. Random** applies random gradient masking, as implemented in Child-Tuning (Xu et al., 2021) and Half Fine-Tuning (Hui et al., 2024). **Highest** greedily updates only the top $p\%$ of gradients at each adaptation step. **GMT** (Li et al., 2025a) accumulates gradients over a batch before selecting and updating the top $p\%$.
- **Continual Learning. IncLoRA** trains a fresh adapter sequentially for each new task without replay or regularization. **OLoRA** (Büyükakyüz, 2024) initializes LoRA matrices via QR decomposition to enforce orthogonality and reduce parameter interference across tasks. **FVG** (Jiang et al., 2025b) leverages task-specific function vectors extracted from transformer hidden states to guide parameter updates. **EWC** (Kirkpatrick et al., 2017) is a regularization-based approach that penalizes deviations in task-critical parameters; we combine it with LoRA for parameter-efficient continual learning.

Hyperparameters. For the single-task mathematical reasoning benchmark, we adapt all models under two distinct mask ratio configurations (p). In the continual setting, we fix the mask ratio to $p = 0.5$ across all gradient-masking methods. We also evaluate a non-sequential baseline where data from both tasks are combined, shuffled, and used in one adaptation run. Specifically, **Mix-SFT** denotes full-parameter supervised fine-tuning on the mixed dataset, while **Mix-LoRA** applies LoRA adaptation under identical conditions. Detailed hyperparameter settings, including learning rates, batch sizes, and schedules, are provided in Appendix A.9.

6.2 RESULTS

Single-task adaptation. Table 1 shows that `CollabMask` obtains the highest aggregate accuracy, 69.7%, compared with 67.0% for SFT, an absolute gain of 2.7 percentage points. Individual configurations are not uniformly dominant: LoRA, Random, or GMT wins for some model–pass-rate combinations. The cross-model aggregate and repeated competitive results therefore support the usefulness of collaboration-guided masking, without requiring the collaboration edges to encode a causal mechanism.

Continual adaptation. For LLaMA-3.1-8B (Table 2), `CollabMask` attains the highest post-second-task code average (55.5%) and overall average (60.0%). It does not win every metric, but the joint retention and new-task result is consistent with a favorable stability–plasticity trade-off. Mixed-data baselines do not consistently dominate sequential methods, so the comparison also indicates that access to both task datasets alone does not explain the observed aggregate result.

Some LLaMA runs improve GSM8K after coding adaptation, whereas the same pattern does not hold uniformly for Mistral. LoRA also exceeds full-parameter SFT on some task-specific metrics

Table 1: Accuracy results (%) of adapted models on mathematical tasks. The best results are **bold**, and the second-best results are underlined. $|\theta|$ denotes the total number of model parameters. p indicates the gradient mask pass rate. For gradient masking strategies, the reported value in average is from the higher accuracy achieved across different p settings.

Model	$\lvert\theta\rvert$	Base	SFT	LoRA	p	Random	Highest	GMT	CollabMask
Llama3	8B	25.5	61.0	42.6	0.5	58.8	50.5	38.0	**63.5**
					0.7	63.0	54.8	38.0	60.3
Mistral	7B	24.0	46.0	48.3	0.5	48.0	49.3	**55.6**	52.0
					0.7	50.0	46.0	50.3	47.0
Qwen3	4B	48.3	77.0	**83.7**	0.5	62.5	80.3	60.0	62.8
					0.7	64.5	53.3	64.3	80.0
Qwen3	8B	43.0	83.8	80.0	0.5	**85.5**	70.0	75.8	82.8
					0.7	79.3	52.8	70.8	83.3
Average	–	–	67.0	63.7	–	65.8	63.6	58.4	**69.7**

Table 2: LLaMa-3.1-8B.

	GSM8K		Humaneval		MBPP		PubmedQA		MMLU		Code	AVG
	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2		
Base	25.5		37.2		61.6		87.0		31.6		49.4	48.6
Mix-SFT	45.0		40.9		63.5		74.0		33.1		52.2	51.3
Mix-Lora	55.5		42.1		62.7		75.5		37.1		52.4	54.6
LoRA	61.0	60.5	40.2	36.6	63.5	63.5	69.0	74.0	39.7	43.4	53.5	54.2
LoRAInc	60.5	**62.0**	36.6	42.1	62.4	63.3	70.5	65.5	41.0	41.9	39.2	42.3
OLoRA	26.5	25.0	26.2	26.2	52.1	52.1	74.5	77.0	31.3	31.1	48.6	46.4
FVG	24.0	19.0	36.0	36.0	61.4	61.1	86.0	82.5	35.4	33.6	46.0	50.9
EWC	55.5	49.5	36.0	37.2	61.9	54.8	74.0	72.5	35.3	40.3	50.1	55.6
SFT	39.5	57.0	37.2	45.1	61.6	64.8	89.0	88.0	36.1	40.3	52.7	55.0
Random	45.5	51.0	37.8	42.1	63.0	65.1	89.5	89.5	37.4	44.3	53.6	58.4
Highest	40.0	46.0	35.4	37.8	58.5	56.9	82.5	76.0	45.1	47.9	47.4	52.9
Collabmask	33.0	48.5	38.4	**45.7**	60.8	**65.3**	88.0	**91.5**	35.1	**49.1**	**55.5**	**60.0**

but is less consistent on held-out general benchmarks, echoing prior reports of adaptation trade-offs (Kumar et al., 2022). These variations motivate reporting all domains rather than interpreting a single benchmark as evidence for the collaboration structure.

On Mistral-7B (Table 3), `CollabMask` has the highest code average (43.4%) and a 50.3% overall average, 0.1 points below the best overall method. This near-tie supports transfer of the collaboration-guided update rule across model families while also delimiting the claim: collaboration-aware masking is competitive, not uniformly superior.

7 Limitations and Future Work

Limitations. Our evidence should be interpreted as statistical and empirical rather than as a complete mechanistic account of LLM computation. The semantic-alignment and task-token analyses show that collaboration edges carry meaningful structure, but they do not by themselves establish causal pathways. The positive-gate indicator is also one measurement choice among several; empirical activation-rate and magnitude diagnostics for this choice appear in Appendix A.5. In addition, the current graphs are estimated from a limited task sample, token-level co-activation is used to guide batch-level optimization, and the measurement is restricted to MLP layers. We do not study whether similar collaboration relations arise in attention layers, nor do we model potentially meaningful cross-layer neuron collaboration. Experiments cover short task sequences, four base models, and primarily mathematics-to-code adaptation. `CollabMask` further introduces one-time forward-pass, hypergraph, and CPU projection costs, plus lightweight graph modulation during adaptation.

Table 3: Mistral-7B Results.

Model	GSM8K		Humaneval		MBPP		PubmedQA		MMLU		Code	AVG
	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2	t_1	t_2		
Base	24.0		29.9		50.8		54.0		33.1		40.4	38.4
Mix-SFT	44.5		28.0		50.0		**80.5**		39.1		39.0	48.4
Mix-Lora	51.0		30.5		53.2		74.0		40.4		41.9	49.8
Lora	54.0	**54.0**	29.9	**34.1**	52.9	51.1	83.5	67.5	32.3	41.0	42.6	49.5
LoraInc	50.0	53.0	28.7	29.9	51.6	53.4	84.5	74.5	32.3	**41.2**	41.7	**50.4**
Olora	26.0	26.5	29.9	29.9	50.8	50.8	59.5	60.0	34.7	34.7	40.4	40.4
FVG	21.5	27.0	29.3	29.3	50.5	**53.7**	62.0	65.0	31.9	33.0	41.5	41.6
EWC	37.0	31.5	26.8	25.6	46.3	46.6	87.0	65.5	36.9	34.1	36.1	40.7
Normal	53.5	46.5	32.3	29.3	52.1	53.4	80.5	79.0	32.0	32.9	41.4	48.2
Random	48.0	49.5	29.3	31.1	49.5	49.2	79.5	69.5	32.4	38.7	40.2	47.6
Highest	56.5	0.0	33.5	0.0	54.5	0.0	79.5	0.0	32.3	0.0	0.0	0.00
Collabmask	59.0	49.0	31.1	**34.1**	52.1	52.6	87.5	79.0	31.9	36.7	**43.4**	50.3

Future work. Future experiments should extend controlled token-type and topic contrasts beyond GSM8K/PubMedQA, compare alternative activation definitions, estimate graph stability across samples and task sequences, and add interventions that distinguish correlation from causal influence. We also aim to extend collaboration measurement to broader and more general scenarios, including attention modules, cross-layer relationships, mixture-of-experts models, longer continual sequences, and preference adaptation such as DPO (Rafailov et al., 2023). Another important direction is to study the semantic meaning of co-activation in greater detail: beyond showing that collaborating neurons have related descriptions, future analyses could examine what types of concepts, operations, or task roles are represented by recurrent co-activation patterns.

8 Conclusion

We identify neuron collaboration through a task-conditioned hypergraph projection and find that collaboration neighbors align with independently derived neuron descriptions and task-token co-activation patterns. `CollabMask` then applies this structure to continual adaptation: collaboration-aware masks yield a 2.7-point aggregate gain in single-task mathematics and competitive retention–adaptation trade-offs in sequential experiments. These results connect neuron collaboration, structured gradient sparsity, and task-specific dynamics, supporting collaboration graphs as a useful interpretable structure for LLM adaptation.

Reproducibility statement

All experiments in this paper are implemented in PyTorch 2.8 with CUDA 12.4 on NVIDIA A100 GPUs. The codebase, including training and evaluation scripts, will be made publicly available, and the codebase is submitted with this manuscript to reviewers. Full hyperparameter settings are listed in Appendix A.9. LLaMA-3-8B, Qwen-3-4B, and Mistral-7B are adapted using 4 NVIDIA A100 GPUs, while Qwen-3-8B is adapted using 8 A100 GPUs. Each training step takes approximately 30 seconds. The construction of the collaboration graph for each layer requires approximately 200 seconds on a CPU.

Use of generative AI tools in writing and research disclosure statement

We used large language models (LLMs) to assist in the preparation of this manuscript. Specifically, LLMs were employed for text editing and grammar refinement. GPT-5.1-Codex Max, integrated via the Cursor coding agent, was used to support the development of experimental code. All technical ideas, experimental design, implementation, and analysis were performed solely by the authors. We have reviewed all AI-assisted work and take responsibility for the final content of this work.

ETHICS STATEMENT

CollabMask is a low-risk adaptation method that uses neuron-collaboration-aware gradient masking. Potential considerations include: (i) unintended amplification of pretrained model biases, (ii) possible impact on generalization to out-of-distribution data, (iii) additional computation and memory overhead, and (iv) approximations in collaboration graphs that may limit interpretability. While no direct safety risks are introduced, we recommend careful evaluation and monitoring before deploying adapted models in high-stakes or sensitive applications. All external codebases, tools, models, and datasets used in this work are publicly available and are properly cited. The datasets used are widely adopted public benchmarks (e.g., GSM8K, HumanEval) and do not contain personally identifying information or offensive content.

REFERENCES

Wasi Uddin Ahmad, Aleksander Ficek, Mehrzad Samadi, Jocelyn Huang, Vahid Noroozi, Somshubra Majumdar, and Boris Ginsburg. Opencodeinstruct: A large-scale instruction tuning dataset for code llms. 2025. URL https://arxiv.org/abs/2504.04030.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv:2108.07732*, 2021.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, and et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv:2303.12712*, 2023.

Kerim Büyükakyüz. Olora: Orthonormal low-rank adaptation of large language models, 2024. URL https://arxiv.org/abs/2406.01775.

Mark Chen, Jerry Tworek, Heewoo Jun, and et al. Yuan. Evaluating large language models trained on code. *arXiv:2107.03374*, 2021.

Dami Choi, Vincent Huang, Kevin Meng, Daniel D. Johnson, Jacob Steinhardt, and Sarah Schwettmann. Scaling automatic neuron description, 2024. https://transluce.org/neuron-descriptions.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021.

Alex Foote, Neel Nanda, Esben Kran, Ioannis Konstas, Shay Cohen, and Fazl Barez. Neuron to graph: Interpreting language model neurons at scale. *arXiv:2305.19911*, 2023.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv:2106.09685*, 2021.

Tingfeng Hui, Zhenyu Zhang, Shuohuan Wang, Weiran Xu, Yu Sun, and Hua Wu. Hft: Half fine-tuning for large language models. *arXiv:2404.18466*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lelio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothee Lacroix, and William El Sayed. Mistral 7b. *arXiv:2310.06825*, 2023.

Gangwei Jiang, Caigao Jiang, Zhaoyi Li, Siqiao Xue, Jun Zhou, Linqi Song, Defu Lian, and Ying Wei. Unlocking the power of function vectors for characterizing and mitigating catastrophic forgetting in continual instruction tuning. In *ICLR*, 2025a. URL https://openreview.net/forum?id=gc8QAQfXv6.

Gangwei Jiang, Caigao Jiang, Zhaoyi Li, Siqiao Xue, Jun Zhou, Linqi Song, Defu Lian, and Ying Wei. Unlocking the power of function vectors for characterizing and mitigating catastrophic forgetting in continual instruction tuning, 2025b. URL https://arxiv.org/abs/2502.11019.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. In *EMNLP-IJCNLP*, pp. 2567–2577, 2019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521–3526, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution, 2022. URL https://arxiv.org/abs/2202.10054.

Biagio La Rosa, Leilani H. Gilpin, and Roberto Capobianco. Towards a fuller understanding of neurons with clustered compositional explanations. *arXiv:2310.18443*, 2023.

Michael Lan, Philip Torr, and Fazl Barez. Towards interpretable sequence continuation: Analyzing shared circuits in large language models. *arXiv:2311.04131*, 2023.

Haoling Li, Xin Zhang, Xiao Liu, Yeyun Gong, Yifan Wang, Qi Chen, and Peng Cheng. Enhancing large language model performance with gradient-based parameter selection. *AAAI*, 39 (23):24431–24439, Apr. 2025a. doi: 10.1609/aaai.v39i23.34621. URL https://ojs.aaai.org/index.php/AAAI/article/view/34621.

Pan Li and Olgica Milenkovic. Inhomogeneous hypergraph clustering with applications. In *NeurIPS*, volume 30, 2017.

Yueyan Li, Wenhao Gao, Caixia Yuan, and Xiaojie Wang. Fine-tuning is subgraph search: A new lens on learning dynamics. *arXiv preprint arXiv:2502.06106*, 2025b.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In *NeurIPS*, 2023.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/TASLPRO.2025.3606231.

Tuomas Oikarinen and Tsui-Wei Weng. Linear explanations for individual neurons. *arXiv:2405.06855*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

Hassan Sajjad, Nadir Durrani, and Fahim Dalvi. Neuron-level interpretation of deep nlp models: A survey. *arXiv:2108.13138*, 2022.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Comput. Surv.*, 58(5), November 2025a. ISSN 0360-0300. doi: 10.1145/3735633. URL https://doi.org/10.1145/3735633.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(5):1–42, 2025b.

Qwen Team. Qwen3 technical report. *arXiv:2505.09388*, 2025.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature Medicine*, 29(8):1930–1940, 2023.

Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4:1185–1197, 2022.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence*, 46(8):5362–5383, 2024a.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. Finding skill neurons in pre-trained transformer-based language models. *arXiv:2211.07349*, 2022a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *NeurIPS*, 37:95266–95290, 2024b.

Zifeng Wang, Zheng Zhan, Yifan Gong, Geng Yuan, Wei Niu, Tong Jian, Bin Ren, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. Sparcl: Sparse continual learning on the edge. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 20366–20380. Curran Associates, Inc., 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/80133d0f6eccaace15508f91e3c5a93c-Paper-Conference.pdf.

Qingsong Wen, Jing Liang, Carles Sierra, Rose Luckin, Richard Tong, Zitao Liu, Peng Cui, and Jiliang Tang. Ai for education (ai4edu): Advancing personalized education with llm and adaptive learning. In *KDD*, pp. 6743–6744, 2024.

Yifeng Xiong and Xiaohui Xie. Oplora: Orthogonal projection lora prevents catastrophic forgetting during parameter-efficient fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34088–34096, 2026.

Haoyun Xu, Runzhe Zhan, Yingpeng Ma, Derek F. Wong, and Lidia S. Chao. Let's focus on neuron: Neuron-level supervised fine-tuning for large language model. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 9393–9406, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.630/.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. Raise a child in large language model: Towards effective and generalizable fine-tuning. *arXiv:2109.05687*, 2021.

Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Computing Surveys*, 58(8):1–41, 2026.

Fangcong Yin, Xi Ye, and Greg Durrett. Lofit: Localized fine-tuning on LLM representations. In *NeurIPS*, 2024. URL https://openreview.net/forum?id=dfiXFbECSZ.

Zeping Yu and Sophia Ananiadou. Neuron-level knowledge attribution in large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *EMNLP*, pp. 3267–3280, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.191. URL https://aclanthology.org/2024.emnlp-main.191/.

Variant	GSM8K (%)	Δ (%)
Full	60.3	–
w/o KNN	57.0	-3.3
w/o α	58.0	-2.3
w/o token reweight	58.0	-2.3

Table 4: Disrupting components of CollabMask on GSM8K.

Biao Zhang, Zhongtao Liu, Colin Cherry, and Orhan Firat. When scaling meets LLM finetuning: The effect of data, model and finetuning method. In *ICLR*, 2024.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *arXiv:2309.01029*, 2023.

George Kingsley Zipf. *Human Behavior and the Principle of Least Effort: An Introduction to Human Ecology*. Addison-Wesley, Cambridge, MA, 1949.

A Appendix

A.1 GSM8K Calculation-Span Example

GSM8K answers often interleave natural-language reasoning with explicit calculation spans delimited by <<...>>. In our collaboration-graph construction, we measure co-activation on tokens in these calculation spans and exclude the surrounding explanatory text.

Question. Julie is reading a 120-page book. Yesterday, she was able to read 12 pages and today, she read twice as many pages as yesterday. If she wants to read half of the remaining pages tomorrow, how many pages should she read?

Answer. Maila read 12 x 2 = `<<12*2=24>>`24 pages today. So she was able to read a total of 12 + 24 = `<<12+24=36>>`36 pages since yesterday. There are 120 - 36 = `<<120-36=84>>`84 pages left to be read. Since she wants to read half of the remaining pages tomorrow, then she should read 84/2 = `<<84/2=42>>`42 pages. `#### 42`

For this example, the graph-construction token filter targets the calculation expressions `12*2=24`, `12+24=36`, `120-36=84`, and `84/2=42`, rather than the prose tokens around them.

A.2 Component Disruption

We disrupt three parts of the method on LLaMA 3.1–8B: hyperedge degree reweighting (α), token-frequency reweighting, and k-nearest-neighbor sparsification. All variants use GSM8K and $p = 0.7$. Removing any component reduces accuracy relative to the full method (60.3%); removing KNN has the largest drop (3.3 points), while either reweighting ablation drops 2.3 points. These interventions show that the observed adaptation gain depends on the specified measurement and sparsification choices. They do not independently establish that those choices recover a unique underlying mechanism.

A.3 Theoretical Foundation of Projection to Collaboration Graph

This section explains why the hypergraph projection used by `CollabMask` is a valid way to convert token-level neuron co-activation into pairwise collaboration weights. The key requirement is that cuts in the projected graph should reflect cuts in the original co-activation hypergraph: if a group of neurons is frequently activated together, separating those neurons should incur a large cut cost. We first recall the projection principle from Inhomogeneous Hypergraph Clustering (InH) (Li & Milenkovic, 2017), then show that our projection satisfies the same cut-preservation condition exactly for the cut cost used in `CollabMask`.

A.3.1 HYPERGRAPH PROJECTION PRINCIPLE

InH shows that a hypergraph can be represented by a weighted pairwise graph when the graph approximately preserves the cost of cutting each hyperedge. This is useful in our setting because each token activates a set of neurons, naturally forming a hyperedge, while the gradient-masking step requires pairwise neuron-neuron collaboration scores.

Formally, let $\mathcal{H} = (V, E)$ denote a hypergraph. For a hyperedge $e \in E$, let $w_e(S)$ denote the cost of separating a subset $S \subset e$ from its complement $e \setminus S$. A cut-cost function is *consistent* if

$$w_e(S) = w_e(e \setminus S),$$

meaning that the cost does not depend on which side of the cut is named first.

For each hyperedge e, the projection constructs a complete weighted subgraph $G_e = (V^{(e)}, E^{(e)}, w^{(e)})$ over the same vertices,

$$V^{(e)} = e, \quad E^{(e)} = \{\{v, \tilde{v}\} \mid v, \tilde{v} \in e, v \neq \tilde{v}\}.$$

The projected edge weights should approximate the original hyperedge cut cost:

$$w_e(S) \leq \sum_{v \in S, \tilde{v} \in e \setminus S} w_{v\tilde{v}}^{(e)} \leq \beta^{(e)} w_e(S), \tag{3}$$

where $\beta^{(e)}$ is the approximation factor. When $\beta^{(e)} = 1$, the projected graph preserves the hyperedge cut cost exactly.

After constructing G_e for each hyperedge, the final projected graph sums the contributions from all hyperedges:

$$w_{v\tilde{v}} = \sum_{e \in E} w_{v\tilde{v}}^{(e)}. \tag{4}$$

A.3.2 PROJECTION USED IN COLLABMASK

In `CollabMask`, each hyperedge $e = E_\ell^{(h)}(x)$ is the set of neurons co-activated by token x at layer ℓ. We want the projection to assign a large penalty when many co-activated neurons are split apart, while avoiding excessive influence from very large hyperedges that may reflect generic activation patterns rather than task-specific collaboration. We therefore define the cut cost

$$w_e(S) = \frac{|S| \cdot (|e| - |S|)}{|e|^\alpha}, \tag{5}$$

where α controls how strongly large hyperedges are down-weighted. This cost is consistent because replacing S with $e \setminus S$ leaves the numerator unchanged.

For this cut cost, an exact graph projection is obtained by assigning every pair of distinct vertices inside the hyperedge the same edge weight:

$$w_{v\tilde{v}}^{(e)} = \frac{1}{|e|^\alpha}, \qquad v, \tilde{v} \in e,\ v \neq \tilde{v}. \tag{6}$$

Indeed, for any $S \subset e$,

$$\sum_{v \in S, \tilde{v} \in e \setminus S} w_{v\tilde{v}}^{(e)} = |S| \cdot (|e| - |S|) \cdot \frac{1}{|e|^\alpha} = w_e(S).$$

Thus, equation 3 holds with $\beta^{(e)} = 1$: the projected pairwise graph exactly preserves the chosen hyperedge cut cost.

Effect of the hyperedge-size exponent. The maximum cut cost for a hyperedge is reached when the split is balanced, $|S| = |e|/2$, giving

$$\max\ w_e(S) = \frac{1}{4|e|^{\alpha-2}}.$$

Algorithm 1 `CollabMask`

Hyperparameters: Dataset $\mathcal{D}$; mask rate p; hyperedge factor α; token factor β; Collab weight λ; nearest neighbor range k; lr η
Initialize model parameters $\boldsymbol{w}_0$ from pretrained model
// Token Statistics
$\mathrm{rank}(x) \leftarrow$ Compute token frequency on $\mathcal{D}$
for $x \in \mathcal{D}$ **do**
  $TR(x) \leftarrow \mathrm{rank}(x)^{\beta}$
end for
// Empirical Identification of Collaboration
for each layer ℓ **do**
  for $x \in \mathcal{D}$ **do**
    $E_{\ell}^{(h)}(x) \leftarrow \{n \mid act_{\ell,n}(x) > 0\}$
  end for
  for $i, j \in E_{\ell}^{(h)}(x)$ **do**
    $G_{\ell}(i,j) \mathrel{+}= |E_{\ell}^{(h)}(x)|^{-\alpha} TR(x)$
  end for
end for
// Collaboration-Guided Masked Adaptation
for step $t = 0, ..., T-1$ **do**
  $g_t \leftarrow \nabla_{\boldsymbol{w}_t} \mathcal{L}$
  $\boldsymbol{A} \leftarrow |g_t| \odot \mathrm{Norm}(k\mathrm{NN}(G_{\ell}))$
  $S \leftarrow (1-\lambda)|g_t| + \lambda \boldsymbol{A}$
  $\boldsymbol{M}_t \leftarrow$ Top-$p\%(S)$
  $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta\, g_t \odot \boldsymbol{M}_t$
end for

With our setting $\alpha = 2.5$, this becomes

$$\max\ w_e(S) = \frac{1}{4\sqrt{|e|}},$$

so larger hyperedges contribute less to the projected graph than smaller, more selective co-activation patterns. This matches the design goal of `CollabMask`: broad hyperedges often arise from common tokens or generic activation behavior, whereas smaller hyperedges are more likely to capture specific neuron collaboration.

Resulting collaboration graph. Summing the projected contributions of all token-induced hyperedges yields the layer-wise collaboration graph:

$$w_{\ell}(i,j) = \sum_{x \in \mathcal{D}} \frac{\mathbf{1}\big[i, j \in E_{\ell}^{(h)}(x)\big]}{|E_{\ell}^{(h)}(x)|^{\alpha}}.$$

The main text further multiplies each token contribution by the token reweighting factor $TR(x)$, producing equation 1. This reweighting does not change the projection argument; it only rescales each token-induced hyperedge before its pairwise contribution is added to the final collaboration graph.

A.4 Pseudocode

Algorithm 1 summarizes the empirical identification and collaboration-guided adaptation protocol described in the main text.

A.5 Neuron activation results

In this section, we visualize neuron activation distributions and illustrate how activations are classified as **activated** by `CollabMask`. Activations are collected from LLaMA-3.1-8B on 400 GSM8K samples. We record layers $\{0, 5, 10, 15, 20, 25, 31\}$ to span early, middle, and late blocks.

Table 5: Layer-wise neuron activation statistics on GSM8K (400 samples, 409,600 token positions per layer). $v_{\ell,n}$: `down_proj` input; $I_{\ell,n} = \mathbf{1}[act_{\ell,n} > 0]$.

Layer	**Activation rate** (> 0)	$\mathbb{E}[v_{\ell,n}]$	$\mathbb{E}[\lvert v_{\ell,n}\rvert \mid I_{\ell,n} = 1]$
0	0.581	0.0077	0.0101
5	0.194	0.0166	0.0178
10	0.216	0.0207	0.0216
15	0.189	0.0282	0.0355
20	0.239	0.0305	0.0356
25	0.244	0.0370	0.0408
31	0.536	0.2072	0.2701

Table 5 summarizes layer-wise statistics, where $v_{\ell,n}$ denotes the per-neuron `down_proj` input. Across all recorded layers, $\mathbb{E}[|v_{\ell,n}| \mid I_{\ell,n} = 1]$ exceeds the layer-wise mean $\mathbb{E}[v_{\ell,n}]$, confirming that our detector preferentially selects token–neuron pairs with stronger downstream responses rather than background noise.

Figure 5 shows that detection effectiveness also varies across neurons: layer 0 neuron 1000 and layer 5 neuron 0 are flagged on ≈80% of tokens, whereas layer 0 neuron 0 and layer 20 neuron 1500 remain sparse (≈18–20%). In each panel, orange (detected) mass aligns with the dominant histogram modes, but some gray regions with non-trivial `down_proj_input` are left out. A likely reason is that LLaMA-3 uses SiLU, which is non-zero on negative gate inputs, while our criterion $I_{\ell,n}(x) = \mathbf{1}[act_{\ell,n}(x) > 0]$ ignores the sign of $up_{\ell,n}(x)$ in the full product $act_{\ell,n}(x) \cdot up_{\ell,n}(x)$; salient negative contributions may therefore be missed.

A.6 Token-frequency reweighting

Token-frequency imbalance can skew co-activation hypergraph construction. Following Zipf's law (Zipf, 1949), a small set of tokens accounts for a disproportionate share of occurrences. In language, these are often function words with limited semantic content (e.g., "the"); in LLMs, special tokens such as `<pad>` are especially dominant under fixed-length padding, sometimes comprising nearly half of all token positions. Because each token induces a hyperedge $E_\ell^{(h)}(x)$, such high-frequency tokens can disproportionately shape the projected collaboration graph despite contributing little task-specific signal.

To counter this bias, we reweight each token by its frequency rank. Tokens are ranked in ascending order of frequency on the training corpus, and token x receives

$$TR(x) = \mathrm{rank}(x)^\beta,$$

with $\beta = 1.3$ in all experiments. This upweights medium- and low-frequency tokens relative to the most frequent ones. We additionally discard hyperedges induced by the three highest-frequency tokens, which are typically padding- or format-related.

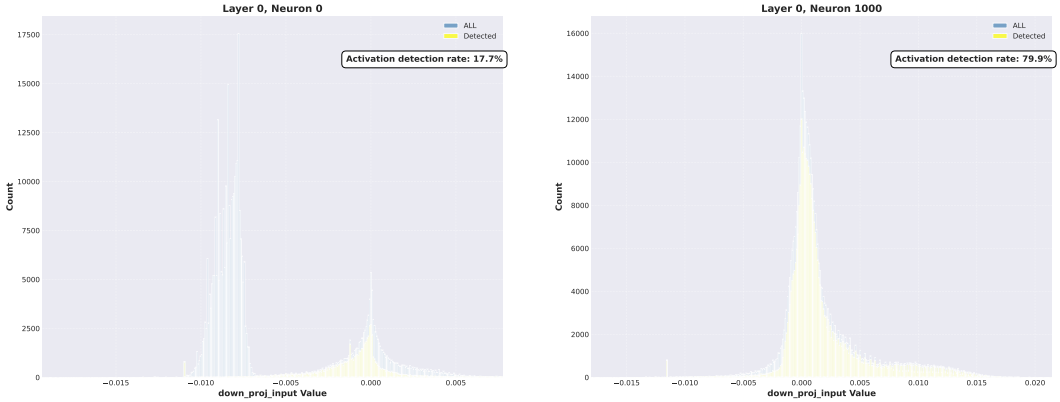
Figure 6 shows how β shifts mass away from the highest-frequency ranks. At $\beta = 0$ (no reweighting), influence is concentrated among the most frequent tokens; as β increases, medium-frequency tokens contribute a larger share of the total influence, yielding a collaboration graph less dominated by syntactic and padding artifacts.

A.7 Semantic Alignment Details

This appendix describes the semantic-alignment validation used in Section 4.1. The goal is to test whether high-weight collaboration edges connect neurons whose independently generated descriptions are semantically closer than expected under random within-layer pairing.

Score-gated neuron-description similarity. For each neuron n, Choi et al. (2024) provide a positive and a negative natural-language explanation, denoted $D^+(n)$ and $D^-(n)$, together with quality scores $q^+(n)$ and $q^-(n)$. Let $e^d(n) = \mathrm{BGE}(D^d(n))$ be the BGE-M3 embedding of direction $d \in \{+, -\}$. For a threshold τ, we define a validity indicator

$$V^d(n;\tau) = \mathbf{1}\big[q^d(n) > \tau \ \wedge \ D^d(n) \text{ is nonempty}\big].$$



Layer 0, neuron 0 (17.7%). Layer 0, neuron 1000 (79.9%).

Layer 5, neuron 0 (81.3%). Layer 20, neuron 1500 (19.9%).

Figure 5: Neuron activation distributions on GSM8K. Percentages denote the activation detection rate. Gray: all observations; orange: detected activations.

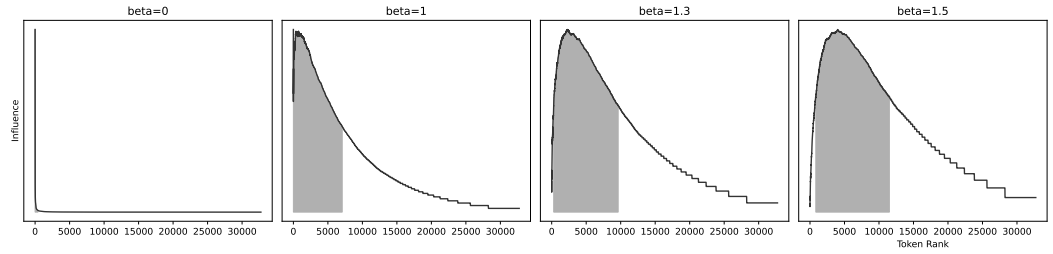


Figure 6: Token influence under different reweighting exponents β. Influence is the normalized product of token frequency and $TR(x)$. The shaded band marks the narrowest rank interval that accounts for $> 60\%$ of total influence at each β.

For a neuron pair (i, j) in layer ℓ, the score-gated directional similarity set is

$$\mathcal{S}_\tau(i,j) = \left\{ \cos\left(e^{d_i}(i), e^{d_j}(j)\right) \;\middle|\; d_i, d_j \in \{+,-\},\ V^{d_i}(i;\tau)V^{d_j}(j;\tau) = 1 \right\}.$$

The pairwise explanation similarity used for validation is the maximum valid directional match,

$$\mathrm{Sim}_\tau(i,j) = \max \mathcal{S}_\tau(i,j), \tag{7}$$

with the pair rejected if $\mathcal{S}_\tau(i,j)$ is empty. This score gate prevents low-quality explanations from determining the similarity score while retaining the best supported semantic correspondence be-

Table 6: Pooled semantic-alignment tests: top-500 collaboration edges per layer versus random within-layer neuron pairs (32 layers; n_{top}=16,000, n_{rand}=128,000).

Statistic	Value
Random mean / median	0.5185 / 0.5180
Top-edge mean / median	0.5704 / 0.5686
Mean difference (95% CI)	0.0519 [0.0509, 0.0529]
Median difference (95% CI)	0.0506 [0.0495, 0.0517]
Hodges–Lehmann shift	0.0511
KS D (two-sided p)	0.3252 (≈ 0)
Mann–Whitney one-sided p	≈ 0
Cliff's δ (95% CI)	0.4347 [0.4271, 0.4425]
Label-permutation p (MW / mean diff.)	0.0001 / 0.0001
Pair-key overlap (random ∩ top)	0

tween the two neurons. In the top-edge tables we use $\tau = 0.6$; for the distributional validation and random-pair baseline we use $\tau = 0.5$ to obtain stable accepted sample sizes.

Top-edge and random-pair sampling. Let G_ℓ be the projected collaboration graph for layer ℓ and $w_\ell(i, j)$ its undirected edge weight. For each of the 32 layers, we extract

$$\mathcal{T}_\ell = \text{TopK}_{(i,j):i<j} \big(w_\ell(i, j), K = 500\big),$$

the 500 strongest within-layer collaboration edges. As a null comparison, we uniformly sample within-layer neuron pairs and accept a pair only if Equation 7 is defined, continuing until 4000 accepted random pairs are obtained per layer. This yields $n_{\text{top}} = 16{,}000$ top-edge pairs and $n_{\text{rand}} = 128{,}000$ random pairs after pooling layers. The two pools have zero pair-key overlap.

Distributional validation. For each accepted pair, we compute $\text{Sim}_{0.5}(i, j)$ and compare the pooled top-edge distribution against the pooled random-pair distribution under the one-sided alternative that top collaboration edges have larger semantic similarity. Table 6 reports the summary statistics. The top-edge mean (0.5704) and median (0.5686) exceed the random mean (0.5185) and median (0.5180), with mean difference 0.0519 (95% bootstrap CI $[0.0509, 0.0529]$) and Hodges–Lehmann shift 0.0511. A Kolmogorov–Smirnov test gives $D = 0.3252$ ($p \approx 0$), and a one-sided Mann–Whitney test gives the same direction with Cliff's $\delta = 0.4347$ (95% bootstrap CI $[0.4271, 0.4425]$). A label-permutation null preserving group sizes gives $p = 0.0001$ for both the Mann–Whitney statistic and the mean difference. These results support the claim that strong collaboration edges preferentially connect neurons with semantically aligned external explanations.

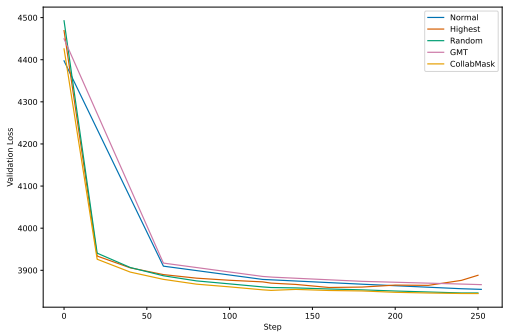
Auxiliary weight–similarity correlation. As an additional check, we sample 2000 nonzero edges per layer with probability inversely proportional to edge-weight-bin density, which flattens the long-tailed edge-weight distribution and gives rare high-weight edges more coverage. For these sampled edges, we again compute the score-gated maximum valid similarity in Equation 7. This correlation-style analysis complements the top-versus-random test by asking whether semantic similarity varies continuously with collaboration weight.

A.8 Additional Adaptation Results

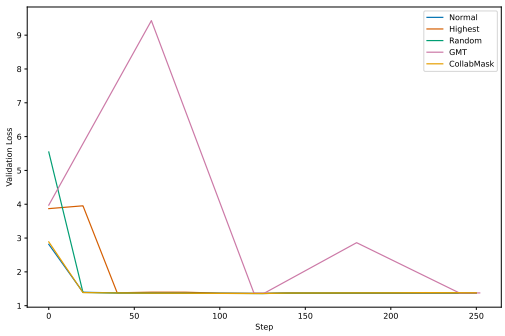
Figure 7 reports validation curves for single-task GSM8K adaptation on LLaMA-3-8B and Mistral-7B and a continual run on LLaMA-3.1-8B with GSM8K followed by OpenCodeInstruct (Ahmad et al., 2025). Across these runs, `CollabMask` remains competitive with the strongest masking controls, providing complementary trajectory-level adaptation evidence.

A.9 Hyperparameters

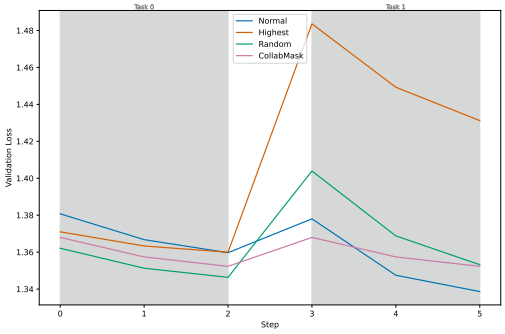
Table 7–11 summarize the hyperparameters used in our experiments. Unless otherwise noted, the same settings are shared across `CollabMask` and all baselines within each experimental setting.



LLaMA-3-8B, GSM8K (single-task).



Mistral-7B, GSM8K (single-task).



LLaMA-3.1-8B, continual learning (OpenCodeInstruct).

Figure 7: Validation perplexity or loss during adaptation. Top row: single-task math reasoning. Bottom: two-task continual adaptation on LLaMA-3.1-8B.

Table 7: Shared adaptation hyperparameters.

Hyperparameter	**Symbol**	**Value**	**Notes**
Training samples per task	–	4,000	GSM8K / OpenCodeInstruct
Validation samples per task	–	1,000	Held-out from training split
Max sequence length	–	512	Tokenized input length
Batch size	–	32	Per device
Training epochs	–	3	Best checkpoint selected
Optimizer	–	Adam	AdamW in LoRA runs
Learning rate	–	1×10^{-5}	Mistral-7B on GSM8K: 5×10^{-6}
Weight decay	–	0.01	
Learning-rate schedule	–	constant	
Random seed	–	42	
Precision	–	bfloat16	

General adaptation settings. All full-parameter methods use Adam with validation after each epoch; the checkpoint with the lowest validation loss is used for downstream evaluation. Dataset splits are fixed with shuffling disabled for reproducibility. For Mistral-7B on the GSM8K math task, we halved the learning rate to 5×10^{-6}.

Gradient masking settings. For single-task mathematical reasoning, we evaluate two mask pass rates (p). For continual learning and all gradient-masking baselines in that setting, we fix $p = 0.5$.

Collaboration-graph construction. Collaboration graphs are built once before adaptation from a subset of the downstream training data. The same graph is reused throughout adaptation unless otherwise noted.

Table 8: Gradient-masking hyperparameters (`CollabMask`, Random, Highest, GMT).

Hyperparameter	Symbol	Value	Notes
Mask pass rate (single-task)	p	0.5, 0.7	Swept in Table 1
Mask pass rate (continual learning)	p	0.5	Fixed for all masking methods
Collaboration weight	λ	0.5	`CollabMask` only; CollabScore in Sec. 5
GMT batch accumulation	–	3	GMT-specific setting
Masked parameter blocks	–	MLP only	`up_proj`, `gate_proj`, `down_proj`

Table 9: Collaboration-graph and co-activation hyperparameters (`CollabMask`).

Hyperparameter	Symbol	Value	Notes
Co-activation measurement batches	`end_batch`	3	Number of training batches used
Hyperedge degree exponent	α	2.5	Projection reweighting
Token rank exponent	β	1.3	Token-frequency reweighting
Filtered high-frequency tokens	–	Top 3	Removed before projection
KNN neighbors per neuron	k	10	Collaboration-graph sparsification

Baseline-specific settings. Non-`CollabMask` baselines share the general settings in Table 7 unless noted below. Where available, we reuse official implementations or widely referenced open-source code released by the original authors or the community Büyükakyüz (2024); Jiang et al. (2025b); van de Ven et al. (2022).

Downstream evaluation. Generation and judging settings used for benchmark evaluation after adaptation.

Table 10: Baseline-specific hyperparameters.

Method	Key hyperparameter	Value	Notes
LoRA / Mix-LoRA	r / α / dropout	16 / 32 / 0.05	Attention and MLP projections
IncLoRA	–	same as LoRA	Fresh adapter per task
OLoRA	r / λ_1	8 / 0.5	Per-task orthogonal adapter
EWC + LoRA	λ_{reg}	1.0	Fisher-weighted L2 penalty
FVG	edit layer / top heads	9 / 10	Function-vector injection
	α_{KL1}	20.0	FV-KL divergence weight (task 1)
	α_{KL2}	0.1	FV-KL weight (secondary)
	α_{KL3}	1.0	FV-KL weight (adapter regularization, task $\geq$2)
	LoRA r / α / dropout	8 / 16 / 0.1	target: `q_proj, v_proj`
Mix-SFT / Mix-LoRA	–	same as SFT / LoRA	Mixed GSM8K + coding data

Table 11: Evaluation hyperparameters.

Hyperparameter	Symbol	Value	Notes
Evaluation split	–	test / official	Per benchmark
Max generated tokens	–	512	
Sampling temperature	–	0.0	Greedy decoding unless noted
Judge model	–	Qwen/Qwen3-30B-A3B-Instruct-2507	For answer verification if used
Coding benchmark framework	–	EvalPlus	HumanEval / MBPP
Evaluated samples (single-task)	–	300	GSM8K
Evaluated samples (continual learning)	–	200	GSM8K, PubMedQA
	–	50 / category	MMLU-Pro
	–	full split	HumanEval, MBPP