CollabMask: Neuron Collaboration Gradient Masks for LLM Fine-Tuning

Anonymous ACL submission

Abstract

The rapid advancement of large language models (LLMs) has increased the need for effective task-specific adaptation. Fine-tuning remains the primary approach but often suffers from redundant parameter updates and catastrophic forgetting in continual learning settings. Existing methods mitigate these issues using gradient masks, yet they largely ignore interactions among neurons. We observe neuron collaboration, where neurons tend to have closely connected neighbors that are co-activated for specific tasks. Leveraging this concept, we propose `CollabMask` (Collaborative Neuron Mask Fine-tuning), which generates dynamic, collaboration-aware gradient masks to induce gradient sparsity. Experiments on mathematical tasks show a 2.7% improvement over full-parameter fine-tuning and a 3.9% improvement over other gradient masking methods. In continual learning settings, `CollabMask` achieves performance within 1% of the best baseline on new tasks, previously learned tasks, and general reasoning ability, demonstrating its effectiveness in enhancing fine-tuning and mitigating catastrophic forgetting. We also discuss the connection between neuron collaboration and neuron-level interpretability, highlighting its potential to improve the explainability of LLMs.

1 Introduction

Large language models (LLMs) have advanced rapidly in recent years and are increasingly adapted to diverse application domains (Bubeck et al., 2023; Achiam et al., 2023; Thirunavukarasu et al., 2023; Wen et al., 2024). Fine-tuning pretrained models is commonly the dominant strategy for task specialization and thus remains a central focus of current research (Wang et al., 2024c; Hu et al., 2021).

However, the increasing scale of pretrained models, coupled with the limited size of downstream datasets, poses significant challenges for fine-tuning (Zhang et al., 2024). Updating billions of parameters using gradients from a relatively small training dataset can introduce noisy and redundant parameter updates, hindering fine-tuning performance and leading to representational changes that degrade previously learned capabilities (Kumar et al., 2022). In continual learning settings, where LLMs are sequentially adapted to multiple tasks, such redundant updates can exacerbate overfitting on current tasks and contribute to catastrophic forgetting (Shi et al., 2025; Büyükakyüz, 2024) and loss of generalization on out-of-distribution data (Kumar et al., 2022).

To mitigate these challenges, recent work has explored constraining parameter updates (introducing sparsity into gradients) during fine-tuning. For instance, Child-Tuning (Xu et al., 2021) and HMT (Hui et al., 2024) apply random gradient masks to selectively update parameters in MLP layers, while GMT (Li et al., 2025) selects gradient masks based on the average gradient across several batches. LoRA (Hu et al., 2021) takes a different approach by restricting updates to low-dimensional parameter subspaces. The core principle of these approaches is to regularize the hypothesis space, thereby reducing overfitting and preserving pretrained knowledge.

Nevertheless, existing methods have two primary drawbacks. First, they use random masks (Hui et al., 2024; Xu et al., 2021) or rely on gradient saliency for selecting task-specific gradient masks (Li et al., 2025). Second, they ignore the explainability of neurons' behaviors from the base model (Choi et al., 2024). Consequently, their redundancy reduction strategies remain less task-sensitive and risk discarding important gradient information. For example, relying solely on high-gradient signals at the batch level can fail to capture neurons that are not gradient-salient for a certain training step but functionally relevant to the downstream task in the pretrained model.

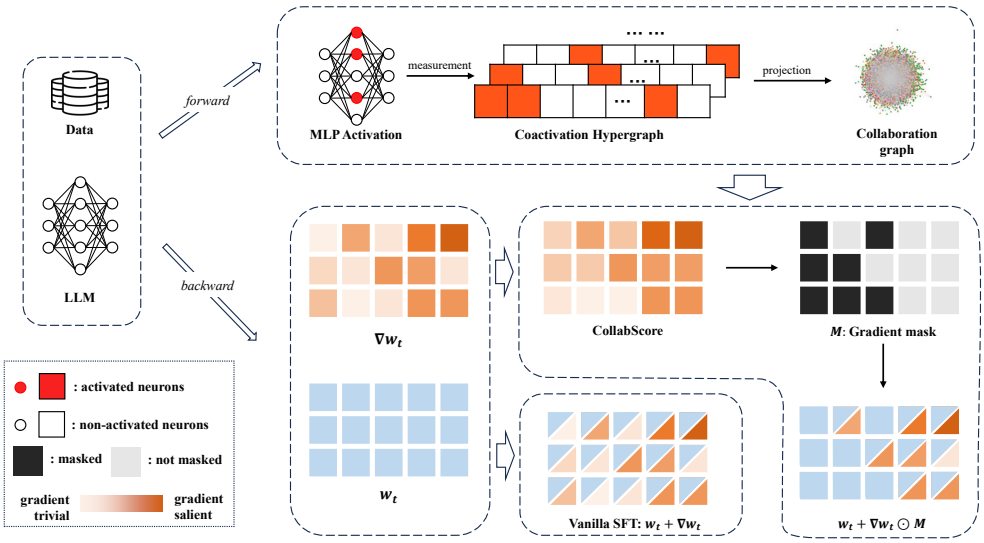

Figure 1: Illustration of our proposed method, CollabMask. First, in the forward pass (top-right block), we perform explainable neuron collaboration measuring. Training samples are fed into the LLM, and the co-activation of neurons in MLP modules is measured for each layer. These co-activations are used to construct a hypergraph, which is then projected to collaboration graphs. Then, in the backward pass during fine-tuning, the parameter gradients $\nabla \boldsymbol{w_t}$ are combined with collaboration graphs to compute the CollabScore, which quantifies the relevance of parameters at the current step t. The top-$p\%$ entries of CollabScore are selected via a gradient mask $\boldsymbol{M}$. Applying the mask enables CollabMask to selectively update functionally relevant parameters, reducing redundancy compared to the vanilla SFT.

In this paper, we propose CollabMask (Neuron Collaboration Gradient Mask Fine-tuning), a novel fine-tuning framework that tackles continual learning challenges from an explainability perspective. CollabMask measures neuron co-activation triggered by samples from the training dataset to capture their collaborative roles in the downstream task, which reflects their functional and semantic similarity. Based on this task-dependent similarity structure, CollabMask applies dynamic gradient masks that adapt to each training batch, constraining parameter updates accordingly. By integrating explainability with dynamic masking, CollabMask enables targeted fine-tuning ,which more precisely reduces redundant parameter updates.

In detail, CollabMask operates in two phases. In the explainable neuron collaboration measuring phase, data from the downstream task are used to construct co-activation hypergraphs for MLP layers, which are then projected into collaboration graphs—weighted graphs where vertices represent neurons and edge weights reflect their tendency to be co-activated. In the masked tuning phase, CollabMask dynamically combines connections in collaboration graphs with gradient information to generate batch-specific gradient masks. These masks suppress updates to less relevant neurons while preserving pretrained collaboration patterns, ensuring that functionally important but non-gradient-salient neurons are still updated. Together, these two phases improve fine-tuning by concentrating updates on functionally meaningful neurons.

We conduct extensive experiments on a mathematical reasoning task and a continual learning setting involving sequential fine-tuning on math followed by a coding task. CollabMask is evaluated against multiple baselines across four LLMs. On the math task, CollabMask improves accuracy by 2.7% over full-parameter SFT and 3.9% over the previous best method. In the continual learning setting, CollabMask achieves performance within 1% of the best baseline on the incoming coding task, the previously learned math task, and a general reasoning task, demonstrating strong generalization and reasoning abilities. These results indicate that CollabMask effectively filters gradient noise during fine-tuning by leveraging neuron collaboration masks, enabling better task adaptation in continual learning scenarios.

Our contributions can be summarized as follows:

1. Through experiments, we show that existing standard fine-tuning methods and gradient masking approaches fail to fully leverage the collaborative behavior of neurons, resulting in less effective parameter updates.

2. We propose `CollabMask`, a method that measures neuron collaboration and constructs collaboration graphs to guide gradient masking. This approach reduces unnecessary parameter updates while preserving neuron collaboration during fine-tuning.

3. We conduct extensive experiments across multiple benchmarks, demonstrating that `CollabMask` outperforms representative baselines in standard fine-tuning and is competitive in continual learning. Furthermore, we show that neuron clusters serve as meaningful explainable units, opening avenues for studying neuron similarity and collaboration.

2 Related work

Gradient-sparse Finetuning strategy A number of fine-tuning strategies have been proposed to reduce parameter redundancy in LLMs. LoRA (Hu et al., 2021) applies low-rank adaptation to parameters, effectively constraining the hypothesis space to a lower-dimensional subspace. In addition, several gradient masking methods have been developed to restrict gradient updates during training. For example, Child-Tuning (Xu et al., 2021) and HFT (Hui et al., 2024) assign each parameter a fixed probability of being updated in each step, while GMT (Li et al., 2025) accumulates gradients and updates only the top-$p\%$ parameters with the largest absolute gradient values.

Neuron-level explainability A major line of research in LLM explainability focuses on understanding the functions and semantics of individual neurons, as well as their influence on model outputs (Sajjad et al., 2022; Zhao et al., 2023). Neuron attribution methods aim to identify critical neurons responsible for specific tasks or knowledge retrieval (Yu and Ananiadou, 2024; Lan et al., 2023; Wang et al., 2022a). Another direction is mechanistic interpretability, which studies how groups of neurons across layers form circuits that explain model behavior or specific decisions (Conmy et al., 2023; He et al., 2024). Some works attempt to assign global functional explanations. Examples include linear explanation approaches (Oikarinen and Weng, 2024), neuron graph explanation (Foote et al., 2023), and using LLMs to describe the semantic functions of neurons (Choi et al., 2024). In addition, La Rosa et al. (2023) extends single-neuron analysis beyond high-activation cases, proposing multi-level activation explanations to capture a richer picture of individual neuron behavior.

Catastrophic Forgetting. Continual learning (Wang et al., 2024a; Shi et al., 2025) studies how models can be trained on a sequence of tasks over time. A fundamental challenge in this setting is catastrophic forgetting (French, 1999), where adaptation to new tasks leads to a degradation of performance on previously learned tasks. In the context of large language models, recent work has explored various strategies to mitigate catastrophic forgetting, including parameter-efficient adaptation methods (Jiang et al., 2025; Büyükakyüz, 2024). Gradient sparsity has also been studied as a means to reduce catastrophic forgetting in continual learning (Wang et al., 2022b).

3 Methodology

In this section, we provide a comprehensive explanation and the underlying principle of `CollabMask`, as shown in Figure 1. First, we measure the co-activation relation among neurons in MLP layers and construct a co-activation hypergraph. Then, the co-activation hypergraph is projected to the collaboration graph. Finally, we utilize the collaboration graph of MLP neurons and the gradient from the backward pass to predict the gradient mask for parameter updates.

3.1 Overview of CollabMask

We begin the introduction of `CollabMask` from the standard stochastic gradient descent (SGD) algorithm and the masked tuning technique, used in (Hui et al., 2024; Xu et al., 2021; Li et al., 2025). In SGD, parameters are updated in the negative direction of the gradient, under the assumption that the gradient of the batch-wise local gradient approximates the global loss function. Let w_t denote the parameters at step t, $\mathcal{L}(B_t, w_t)$ as the loss function on batch B_t, and η the learning rate.The update rule can be described as follows:

$$w_{t+1} = w_t - \eta \frac{\partial \mathcal{L}(batch_t, w_t)}{\partial w_t}. \quad (1)$$

Masked optimization methods such as child-tuning (Xu et al., 2021) and GMT (Li et al., 2025) use

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
// Collaboration Graph Construction
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
// Masked Optimization
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

a parameter-wise mask $\boldsymbol{M}$ to restrict gradient updates:

$$w_{t+1} = w_t - \eta \frac{\partial \mathcal{L}(batch_t, w_t)}{\partial w_t} \odot \boldsymbol{M}, \quad (2)$$

where the mask M is binary,$M \in \{0,1\}^{|w_t|}$,with $|w_t|$ denoting the shape of parameters.

The key advantage of `CollabMask` lies in its treatment of neuron collaboration as an estimate of functional similarity among neurons, which enables an explainable mechanism for gradient mask selection. In contrast to prior approaches, `CollabMask` introduces batch-level dynamic gradient sparsity (Wang et al., 2022b) into the fine-tuning process and preserves non-salient gradients that would otherwise be discarded by methods relying solely on top-gradient magnitude. Moreover, by allowing closely collaborating neurons to be updated jointly, the proposed gradient masking strategy strengthens task-specific neuron collaboration, reduces redundancy in parameter updates, and ultimately leads to improved performance.

3.2 Explainable Neuron Collaboration Measuring

To guide fine-tuning using an interpretable neuron collaboration structure, `CollabMask` partitions neurons in each MLP layer into functional clusters based on their co-activation patterns in the base model. This procedure consists of two stages: (i) constructing co-activation hypergraphs, and (ii) projecting these hypergraphs into weighted undirected graphs, referred to as collaboration graphs. We further empirically verify that the resulting weighted collaboration graphs effectively capture functional similarity among neurons.

3.2.1 Co-activation Hypergraph Construction

Neuron Activation Definition. Consider an MLP layer ℓ containing N_{ℓ} neurons. For an input token x drawn from the downstream dataset $\mathcal{D}$, let $act_{\ell,n}(x)$ denote the output of the nonlinear activation function (e.g., SiLU in LLaMA) applied to the n-th component of the gate-projection vector.

We emphasize that $act_{\ell,n}(x)$ does not correspond to the conventional neuron activation. Under the standard definition, the activation of neuron n is given by

$$act_{\ell,n}(x) \cdot up_{\ell,n}(x), \quad (3)$$

where $u_{\ell,n}$ denotes the n-th component of the up-projection vector. A neuron may be activated positively or negatively, encoding distinct semantic information depending on the sign of $up_{\ell,n}(x)$.

In this work, we use $act_{\ell,n}(x)$ solely to detect whether a neuron is activated by a given input token. Importantly, even though this approach ignores the sign provided by $up_{\ell,n}(x)$, it is still capable of detecting both positive and negative activations, since any neuron with a non-zero contribution will produce a positive $act_{\ell,n}(x)$ after the activation function. Accordingly, we define a binary activation indicator as

$$I_{\ell,n}(x) = \mathbf{1}\big[act_{\ell,n}(x) > 0\big], \quad I_{\ell,n}(x) \in \{0,1\}. \quad (4)$$

The detailed justification for using $z_{\ell,n}(x)$ as an activation detection signal is provided in Appendix A.2.

Co-Activation Hypergraph Construction. For each MLP layer ℓ of the LLM, we construct a co-activation hypergraph. Specifically, for layer ℓ, the

hypergraph is defined as $\mathcal{H}_\ell = (V_\ell, \mathcal{E}_\ell)$, where the vertex set

$$V_\ell = \{1, 2, \ldots, N_\ell\} \tag{5}$$

corresponds to all neurons in the layer, and N_ℓ denotes the intermediate size of the MLP.

For a given input token x, the set of activated neurons forms a hyperedge

$$E_\ell^{(h)}(x) = \{n \in V_\ell \mid I_{\ell,n}(x) = 1\}. \tag{6}$$

By collecting all such hyperedges across the dataset $\mathcal{D}$, we obtain the full co-activation hypergraph

$$\mathcal{H}_\ell = \big(V_\ell, \mathcal{E}_\ell\big), \quad \mathcal{E}_\ell = \big\{E_\ell^{(h)}(x) \mid x \in \mathcal{D}\big\}. \tag{7}$$

3.2.2 Projection to collaboration graph

Following (Li and Milenkovic, 2017), we project the hypergraph $\mathcal{H}_\ell$ onto a weighted undirected graph $G_\ell = (V_\ell, E_\ell)$, referred to as the collaboration graph. For a pair of neurons i and j, the edge weight is defined as:

$$E_\ell(i,j) = \sum_{i,j \in E_\ell^{(h)}(x) \in \mathcal{E}_\ell} \frac{1}{|E_\ell^{(h)}(x)|^\alpha} TR(x), \tag{8}$$

where $E_\ell^{(h)}(x)$ is the hyperedge induced by token x, $|E_\ell^{(h)}(x)|$ denotes its cardinality, α is a scaling parameter, and $TR(x)$ is a token reweighting factor. Intuitively, $E_\ell(i,j)$ measures the strength of collaboration between neurons i and j, adjusted for the size of co-activation and token frequency.

Hyperedge Degree Reweighting. The parameter α controls the relative contribution of hyperedges with different degrees. When $0 < \alpha < 2$, the projection emphasizes high-degree hyperedges, which correspond to widespread co-activation patterns across neurons. In contrast, $\alpha > 2$ assigns greater weight to low-degree hyperedges, thereby prioritizing more localized and specific co-activation structures.

In our experiments, we set $\alpha = 2.5$ to strike a balance between these two regimes, favoring tighter co-activation patterns while still retaining information from broader contextual interactions. A more detailed discussion and justification of this design choice are provided in Appendix A.1.

Token Frequency Reweighting. A naive projection of the co-activation hypergraph tends to overemphasize high-frequency tokens (e.g., special tokens such as eos_token), which often dominate the hypergraph while contributing little to task-specific semantics. To mitigate this bias, inspired by Zipf's law (Zipf, 1949), we introduce a token frequency rank–based reweighting factor $TR(x)$ to estimate the informativeness of token x.

Specifically, we first compute token frequencies over the training dataset and rank all tokens in ascending order of frequency. Each token x is then assigned a rank $\mathrm{rank}(x)$, and the reweighting factor is defined as

$$TR(x) = \mathrm{rank}(x)^\beta. \tag{9}$$

In our experiments, we set $\beta = 1.3$. This formulation assigns relatively greater weights to medium-frequency tokens, which are typically more informative for downstream tasks, while suppressing the influence of overly frequent tokens. Additional details are provided in Appendix A.3.

3.3 Collaboration-Aware Gradient Masking

At each optimization step t, we construct a binary update mask for each parameter block[1] in the MLP layers. Each mask enforces a fixed pass rate of $p\%$, determining the proportion of parameters updated at each step.

To guide mask selection, we define a collaboration-aware score based on the gradient at step t and the collaboration graph G_ℓ, denoted as $\text{Collab-S}(\boldsymbol{w}_t)$, which measures the importance of parameters $\boldsymbol{w}_t$ at step t:

$$\text{Collab-S}(\boldsymbol{w}_t) = (1-\lambda)\left|\nabla_{\boldsymbol{w}_t}\mathcal{L}\right| \;+\; \lambda\, \boldsymbol{A}(\boldsymbol{w}_t), \tag{10}$$

where $\lambda \in [0,1]$ controls the contribution of neuron collaboration. In all experiments, we set $\lambda = 0.5$.

The collaboration activation term $\boldsymbol{A}(\boldsymbol{w}_t)$ captures the average activation strength of neighboring neurons—i.e., neurons that frequently co-activate according to the collaboration graph—and is defined as

$$\boldsymbol{A}(\boldsymbol{w}_t) = \left|\nabla_{\boldsymbol{w}_t}\mathcal{L}\right| \;\odot\; \mathrm{Norm}(k\mathrm{NN}(G_\ell))\,, \tag{11}$$

where G_ℓ denotes the weighted collaboration graph of layer ℓ, $\mathrm{KNN}(\cdot)$ is a row-wise k-nearest-neighbor operator that sparsifies the graph by retaining the top-k edges per neuron, $\mathrm{Norm}(\cdot)$ is a row-wise normalization operator, and $\odot$ denotes element-wise multiplication.

Finally, parameters corresponding to the top $p\%$ of $\text{Collab-S}(\boldsymbol{w}_t)$ values are assigned a mask value

[1] By parameter block, we refer to weight matrices such as `up_proj`, `down_proj`, and `gate_proj` in LLaMA.

of 1, while the remaining parameters are masked out. This design encourages parameters associated with strongly collaborating neurons to be updated jointly, even when their individual gradients are not among the largest, thereby reducing redundant updates and improving fine-tuning efficiency.

4 Experiments

We conduct extensive experiments to demonstrate that `CollabMask` effectively filters gradient noise and enhances fine-tuning performance by preserving neuron collaboration patterns. In a continual learning setting, where the base model is sequentially trained on two tasks, `CollabMask` mitigates catastrophic forgetting more effectively than full-parameter fine-tuning and other gradient-masking methods. We also include ablation studies to evaluate the contribution of each design component in `CollabMask`.

4.1 Experiment setting

Datasets. To evaluate the effectiveness of `CollabMask` in improving fine-tuning performance, we focus on mathematics tasks using GSM8K (Cobbe et al., 2021) for both fine-tuning and evaluation. In the continual learning setting, the model is sequentially fine-tuned on mathematics and coding tasks, using GSM8K (Cobbe et al., 2021) and CodeAlpaca (Chaudhary, 2023), respectively. For coding evaluation, we use the EvalPlus toolkit (Liu et al., 2023) on the HumanEval dataset (Chen et al., 2021). Additionally, we include the general reasoning dataset MMLU-Pro (Wang et al., 2024b) to assess model generalizability and reasoning ability.

Base Models. We select large language models from three families: LLaMA 3 (Grattafiori et al., 2024), Mistral (Jiang et al., 2023), and Qwen 3 (Team, 2025). Specifically, we use LLaMA-3.1-8B, Mistral-v0.1-7B, and Qwen3-4B/Base and Qwen3-8B/Base. All models are pre-trained base models without prior task-specific fine-tuning.

Baselines. To assess the effectiveness of `CollabMask`, we compare it against the following fine-tuning methods:

1. **SFT**: Standard full parameter supervised fine-tuning.
2. **LoRA**: fine-tuning with LoRA (Hu et al., 2021).
3. **Random**: child-tuning (Xu et al., 2021) and half fine-tuning (Hui et al., 2024). Random gradient masking.
4. **Highest**: Updating only the top $p\%$ of gradients in each step.
5. **GMT**: Gradient-Masked Fine-Tuning (Li et al., 2025), which accumulates gradients over batches and updates only the top $p\%$.

For fairness and breadth, we fine-tune all models on all tasks under two mask-ratio p settings.

Continual Learning Setting. To evaluate how `CollabMask` mitigates catastrophic forgetting, we sequentially fine-tune LLaMA-3.1-8B on mathematics and coding tasks using `CollabMask` and the baseline methods. Hyperparameters are selected to optimize fine-tuning performance for each method and are kept fixed across the two consecutive tasks. After completing each task, we measure performance on coding, mathematics, medical, and general reasoning tasks to assess both retention and transfer.

4.2 Experiment results

As shown in Table 1, `CollabMask` consistently achieves better or comparable performance compared with full supervised fine-tuning (SFT) across multiple models and masking rates. In most cases, it outperforms baseline strategies such as Random, Highest, and GMT, demonstrating the effectiveness of its collaboration-aware gradient masking in preserving important neuron interactions. Averaged across all models and settings, CollabMask achieves the highest overall accuracy (67.3), slightly surpassing SFT (67.0), highlighting its robustness and generalizability as a selective fine-tuning method that maintains or improves performance while potentially reducing redundant parameter updates.

Table 2 shows that `CollabMask` consistently delivers strong performance across multiple benchmarks, particularly compared to other parameter-efficient or masking-based methods. On tasks such as HumanEval and HumanEval+, `CollabMask` remains competitive, achieving scores close to or exceeding standard fine-tuning (SFT), while improving over masking baselines and LoRA. When continual learning data (CodeAlpaca) is included, `CollabMask` maintains robustness, achieving 61.3 on GSM8K and 51.4 on MMLU average—comparable to or slightly below SFT but

Table 1: Accuracy results (%) of fine-tuned 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	$\|\theta\|$	Base	SFT	LoRA	p	Random	Highest	GMT	CollabMask
Llama3	8B	23.3	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: Results of continual learning on LLaMa-3-8B.

Method	Training Data	GSM8K	HumanEval	HumanEval+	MMLU Avg
SFT	GSM8K	61.0	40.9	32.9	66.86
Random	GSM8K	63.0	39.6	32.3	52.43
Highest	GSM8K	54.8	38.4	32.9	33.43
GMT	GSM8K	38.0	37.2	31.7	36.14
LoRA	GSM8K	42.6	40.9	32.3	42.50
CollabMask	GSM8K	63.5	40.2	34.8	63.71
SFT	GSM8K + CodeAlpaca	56.7	38.4	31.1	51.57
Random	GSM8K + CodeAlpaca	62.7	40.2	31.1	48.43
Highest	GSM8K + CodeAlpaca	48.3	39.6	32.9	33.43
GMT	GSM8K + CodeAlpaca	46.7	40.2	32.9	33.00
LoRA	GSM8K + CodeAlpaca	42.3	40.9	32.3	36.29
CollabMask	GSM8K + CodeAlpaca	61.3	39.0	32.3	51.40

clearly outperforming naive masking strategies. Moreover, continual training induces smaller performance drops in MMLU average for `CollabMask` than for SFT. Overall, these results demonstrate the consistent effectiveness of `CollabMask` in preserving or enhancing performance through selective, collaboration-aware parameter updates, highlighting the value of its gradient masking approach.

4.3 Ablation study.

We assess the contribution of key design components in `CollabMask` through an ablation study on LLaMA 3.1–8B, including hyperedge degree reweighting, token frequency reweighting, and the k-nearest-neighbor (KNN) operation used in gradient mask selection. For all ablation experiments, we fix the gradient pass rate to $p = 0.7$. As shown in Table 3, removing any individual component consistently degrades performance, indicating that each component contributes positively to the overall effectiveness of `CollabMask`.

Table 3: Ablation study of CollabMask on GSM8K. Performance drops (Δ) are computed relative to the full CollabMask model with $p = 0.7$.

Variant	GSM8K (%)	Δ (%)
Full	60.3	–
w/o KNN	57	-3.3
w/o α	58	-2.3
w/o token reweight	58	-2.3

Table 4: Fraction of top-5 collaboration-graph connections whose neuron-description similarity ranks in the top-k% (90% confidence).

	Top-5 Conn. (%)
Top 10%	17.6
Top 20%	38.5
Top 30%	56.9

5 Discussion

5.1 Validating Explainability of Collaboration Graph

We evaluate how edge weights in the collaboration graph capture semantic similarity among neurons by constructing collaboration graphs for LLaMA 3.1–8B–Instruct (Grattafiori et al., 2024). Using the neuron-description database for the same model from Choi et al. (2024), we compute cosine similarity between pairs of description embeddings. Note that the neuron-description database is derived from massive general data, whereas the collaboration graphs are constructed from a dataset specialized for mathematics in this experiment; thus, it is expected that the collaboration graphs do not fully reflect the description-based similarities.

For each neuron, we select its top-5 neighbors in the collaboration graph and compare their description similarities to those of randomly sampled neurons from the same layer. We report the cumulative proportion of top-5 connections whose similarity ranks fall within the top k% of all neuron pairs with 90% confidence. As shown in Table 4, the collaboration graph reliably identifies semantically similar neurons. More details are provided in Appendix A.4.

5.2 Limitations

The inconsistent performance of `CollabMask` compared to prior methods may stem from several factors. First, similar to previous methods, `CollabMask` still heavily uses gradient saliency in mask selection, even though with neuron collaboration relation to enhance it. Second, the co-activation behavior of neurons is measured at the token level, while the loss function is optimized at the batch level. This mismatch may reduce the effectiveness of the collaboration graph in guiding the gradients of parameters.

`CollabMask` introduces computational overhead due to co-activation hypergraph construction, and graph projection steps. In addition, our current method focuses on MLP layers; neuron collaboration relations within attention modules remain underexplored. Another limitation is that we treat layers as independent, overlooking cross-layer neuron collaborations and causal relations likely exist, as suggested by mechanistic interpretability research (Conmy et al., 2023).

5.3 Future directions

Future work may explore several directions. First, `CollabMask` could be extended to preference optimization frameworks, such as Direct Preference Optimization (DPO) (Rafailov et al., 2023). Second, the framework may be adapted for mixture-of-experts (MoE) models and extended to attention modules. Finally, we plan to further investigate `CollabMask` in continual learning settings, study the effects of gradient sparsity(Wang et al., 2022b) in LLMs, and develop methods for achieving more precise gradient sparsity.

6 Conclusion

In this paper, we introduced `CollabMask`, a fine-tuning method that leverages neuron collaboration to construct collaboration-aware gradient masks. By identifying relevant parameter groups, `CollabMask` preserves collaboration patterns, reduces redundant updates, and mitigates catastrophic forgetting in continual learning settings, resulting in improved performance and generalization. Experiments demonstrate that `CollabMask` consistently outperforms standard supervised fine-tuning and gradient masking baselines. Future work will focus on producing more stable and interpretable collaboration graphs, as well as extending neuron collaboration analysis to attention mechanisms, cross-layer interactions, and mechanistic interpretability.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and et al. 2023. Gpt-4 technical report. arXiv:2303.08774.

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

Kerim Büyükakyüz. 2024. Olora: Orthonormal low-rank adaptation of large language models. Preprint, arXiv:2406.01775.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca.

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

Dami Choi, Vincent Huang, Kevin Meng, Daniel D. Johnson, Jacob Steinhardt, and Sarah Schwettmann. 2024. Scaling automatic neuron description. 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. 2021. Training verifiers to solve math word problems. arXiv:2110.14168.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In NeurIPS, volume 36, pages 16318–16352. Curran Associates, Inc.

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

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

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.

Zhengfu He, Xuyang Ge, Qiong Tang, Tianxiang Sun, Qinyuan Cheng, and Xipeng Qiu. 2024. Dictionary learning improves patch-free circuit discovery in mechanistic interpretability: A case study on othello-gpt. arXiv:2402.12201.

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

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

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. 2023. Mistral 7b. arXiv:2310.06825.

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

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. Preprint, arXiv:2202.10054.

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

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

Haoling Li, Xin Zhang, Xiao Liu, Yeyun Gong, Yifan Wang, Qi Chen, and Peng Cheng. 2025. Enhancing large language model performance with gradient-based parameter selection. AAAI, 39(23):24431–24439.

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

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

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

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

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

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

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

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

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

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

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. NeurIPS, 37:95266–95290.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, and 1 others. 2024c. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. arXiv:2407.16216.

Zifeng Wang, Zheng Zhan, Yifan Gong, Geng Yuan, Wei Niu, Tong Jian, Bin Ren, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. 2022b. Sparcl: Sparse continual learning on the edge. In Advances in Neural Information Processing Systems, volume 35, pages 20366–20380. Curran Associates, Inc.

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

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

Zeping Yu and Sophia Ananiadou. 2024. Neuron-level knowledge attribution in large language models. In EMNLP, pages 3267–3280, Miami, Florida, USA. Association for Computational Linguistics.

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

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

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

A Appendix

A.1 Theoretical Foundation of Projection to Collaboration Graph

Following the Inhomogeneous Hypergraph Clustering (InH) algorithm by Li and Milenkovic (2017) and its associated notation, the projection process in `CollabMask` is designed to preserve the edge cut when partitioning the graph. We demonstrate the consistency of our projection method within the `CollabMask` pipeline and highlight its advantage in handling hyperedges with small degrees.

A.1.1 Projection of Hypergraphs in InH

In InH, it is proved that using the prescribed method, a hypergraph can be projected onto a latent graph, called collaboration graph, and clustering on this latent graph can approximate the clustering on the original hypergraph.

Formally, let $\mathcal{H} = (V, E)$ denote a hypergraph. For each hyperedge (e, w_e), the cost of cutting e into two disjoint subsets $S \subset e$ and $e \setminus S$ is given by $w_e(S)$. A weight function $w_e(\cdot)$ is called consistent if it satisfies $w_e(S) = w_e(e \setminus S)$.

For each InH-hyperedge (e, w_e), the InH algorithm requires a projected complete subgraph $G_e = (V^{(e)}, E^{(e)}, w^{(e)})$ to represent (e, w_e), where

$$V^{(e)} = e, \quad E^{(e)} = \{\{v, \tilde{v}\} \mid v, \tilde{v} \in e, v \neq \tilde{v}\}. \tag{12}$$

The goal is to find edge weights $w^{(e)}_{v\tilde{v}}$ such that

$$w_e(S) \leq \sum_{v \in S, \tilde{v} \in e \setminus S} w^{(e)}_{v\tilde{v}} \leq \beta^{(e)} w_e(S), \tag{13}$$

where $\beta^{(e)}$ is a constant.

After solving for $w^{(e)}$, InH constructs a complete weighted graph $\mathcal{G} = (V, E_o, w)$, where V is the set of vertices of the hypergraph, E_o is the complete set of edges, and the edge weights are computed as

$$w_{v\tilde{v}} = \sum_{e \in E} w^{(e)}_{v\tilde{v}}. \tag{14}$$

A.1.2 Implementation of Projection in `CollabMask`

In `CollabMask`, when defining the cost of cutting a co-activation hypergraph, it is natural to use a non-constant cost function so that cutting out a small fraction of a hyperedge incurs a lower penalty, while keeping the majority of vertices in the same cluster. We define the cost function of a

co-activation hyperedge, which is obviously consistent by definition, as follows,

$$w_e(S) = \frac{|S| \cdot (|e| - |S|)}{|e|^\alpha}, \tag{15}$$

where α is a constant parameter that controls how the hyperedge degree affects clustering, thereby prioritizing low-degree hyperedges in subsequent steps of the `CollabMask` pipeline. When $|S| = \frac{|e|}{2}$, the maximum cost of a hyperedge is reached:

$$\max\ w_e(S) = \frac{1}{4|e|^{\alpha-2}}. \tag{16}$$

Under the current setting $\alpha = 2.5$, this becomes

$$\max\ w_e(S) = \frac{1}{4\sqrt{|e|}}, \tag{17}$$

which emphasizes hyperedges of smaller degree.

The edge weights in the projected graph are set as $w_{v\tilde{v}}^{(e)} = \frac{1}{|e|^\alpha}$, which satisfy the approximation requirement in equation 13:

$$\sum_{v \in S, \tilde{v} \in e \setminus S} w_{v\tilde{v}}^{(e)} = \frac{|S| \cdot (|e| - |S|)}{|e|^\alpha} = w_e(S). \tag{18}$$

Finally, the projection graph is constructed following equation 14, yielding the collaboration graph (similar to equation 8, but without token reweighting):

$$E_\ell(i,j) = \sum_{i,j \in E_\ell^{(h)}(x) \in \mathcal{E}_\ell} \frac{1}{|E_\ell^{(h)}(x)|^\alpha}. \tag{19}$$

A.2 Neuron activation results.

In this section, we visualize neuron activation distributions and illustrate how activations are classified as **activated** by the Collab method using representative samples. The upper panels show the activation distributions of selected neurons at specific positions in LLaMA-3-8B-Instruct, computed over 400 samples from the GSM8K dataset. The lower panels report the proportion of activations that are identified as activated by our algorithm.

A.3 Token-frequency reweighting

In linguistics, the frequency of words is often analyzed as a function of their frequency rank, following Zipf's law (Zipf, 1949). Empirically, words serving primarily syntactic purposes (e.g., "the") typically appear among the highest-frequency ranks. In LLMs, a similar imbalance occurs: the `padding_token` often dominates the input, sometimes accounting for nearly half of the tokens, due to the usual 's padding-to-max-length strategy.

As a result, most hyperedges $E_\ell^{(h)}(x)$ in the co-activation hypergraph are induced by high-frequency tokens that contribute relatively little semantic information. To mitigate this bias, we introduce a token-frequency-based reweighting factor

$$R(x) = \mathrm{rank}(x)^\beta, \tag{20}$$

where $\beta = 1.3$ in the current implementation, and additionally remove hyperedges induced by extremely high-frequency tokens (in our current setting, the top 3 tokens).

Here, the influence of a token is measured as the normalized product of its frequency and its reweighting factor. This adjustment reduces the dominance of high-frequency, low-semantic tokens while amplifying the relative contribution of medium- and low-frequency tokens. The resulting influence distribution across tokens of different frequencies is illustrated in Figure 3.

A.4 Explainability of Collaboration Graph

Choi et al. (2024) provide two descriptions per neuron (positive and negative) along with quality scores. Let $D^+(n)$ and $D^-(n)$ denote the positive and negative descriptions of neuron n, with scores $s(d)$, and let $\mathrm{emb}(d)$ denote the embedding. For rare missing or invalid descriptions, we set $s(d) = 0$.

Per-Layer Results. The per-layer fraction of top-5 collaboration-graph connections whose neuron-description similarity ranks within the top-k% (with 90% confidence) is reported in Table 5.

A.5 Additional Fine-Tuning Experiments Results

The validation perplexity during training is shown in Figure 4, Figure 5, Figure 6.

A.6 Reprodicibility 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.

Models are trained using the Adam optimizer with a learning rate of 1e-5, batch size 32, weight

decay 0.01, and a maximum of 3 epochs, with validation performed at each epoch and the best model retained as the final model. For all datasets, reordering is disabled to ensure reproducibility. For all tasks, we use 4,000 samples for training and 1,000 for validation.

LLaMA-3-8B, Qwen-3-4B, and Mistral-7B are fine-tuned using 4 NVIDIA A100 GPUs, while Qwen-3-8B is fine-tuned 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.

A.7 Use of LLMs

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.

A.8 Potential risk.

`CollabMask` is a low-risk method that improves fine-tuning via 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 fine-tuned models in high-stakes or sensitive applications.

A.9 Artifact usage

All external codebases, tools, models, and datasets used in this work are publicly available and are properly cited in the main paper. We did not provide a separate discussion of licenses, as all artifacts were used strictly for academic research purposes under their original publicly stated terms, without redistribution or commercial use. Our use of existing artifacts is consistent with their intended research-only usage, and the artifacts we introduce are likewise intended solely for research. The datasets used are widely adopted public benchmarks (e.g., GSM8K, HumanEval) and do not contain personally identifying information or offensive content; therefore, no additional anonymization procedures were required. Finally, we do not provide separate artifact documentation, as dataset characteristics such as task domain and evaluation protocol are well established and documented in their original releases

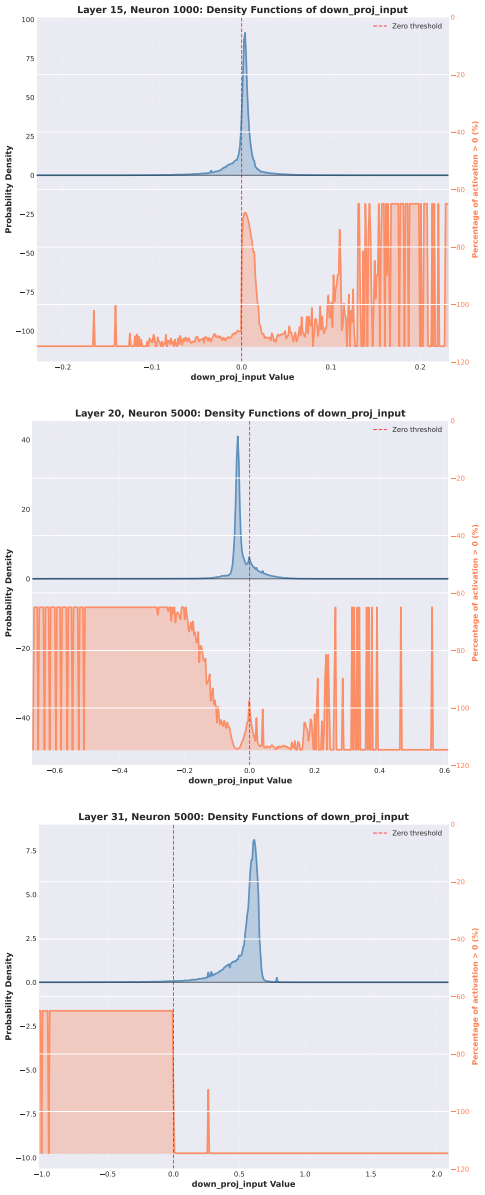

Figure 2: Neuron activation results for neuron at given position, in model LLaMa-3-8B-Instruct trigger by GSM8K.

Table 5: Distribution of neurons with 90% confidence falling into top-x% similarity buckets. The first row reports the average across all layers.

Layer	≥90%	≥80%	≥70%
Avg.	17.6	20.9	18.4
L0	18.6	20.3	17.8
L1	14.9	18.2	16.4
L2	17.7	17.6	15.1
L3	14.6	17.2	16.7
L4	15.3	20.1	18.3
L5	16.5	17.0	13.5
L6	18.2	19.9	17.0
L7	13.3	16.8	15.3
L8	13.8	16.7	16.3
L9	17.1	20.5	18.6
L10	13.6	18.9	18.9
L11	9.7	13.1	14.3
L12	16.3	19.4	18.7
L13	12.2	15.6	15.0
L14	14.2	18.0	17.7
L15	16.4	21.7	19.5
L16	16.7	21.4	19.1
L17	17.5	22.9	20.8
L18	15.8	19.5	18.9
L19	22.1	21.6	17.3
L20	22.3	25.2	20.5
L21	23.3	25.5	20.5
L22	31.0	33.9	20.5
L23	18.8	23.6	21.1
L24	22.3	25.5	19.7
L25	14.2	20.6	19.8
L26	14.6	20.2	19.5
L27	17.2	22.3	20.3
L28	20.9	24.5	20.4
L29	22.7	25.2	20.3
L30	20.8	23.2	19.3
L31	19.7	23.5	20.3

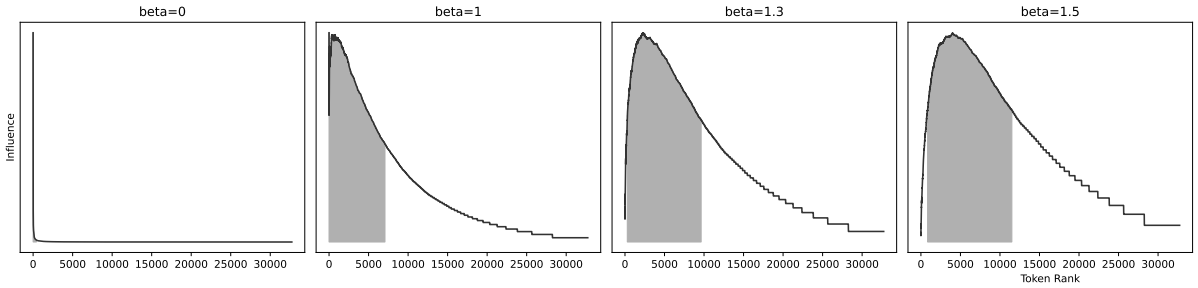

Figure 3: The influence of tokens with different frequency under different reweighting factor β.The dark area is the minimum token rank range that covers more than 60% of the total influence under the β setting.

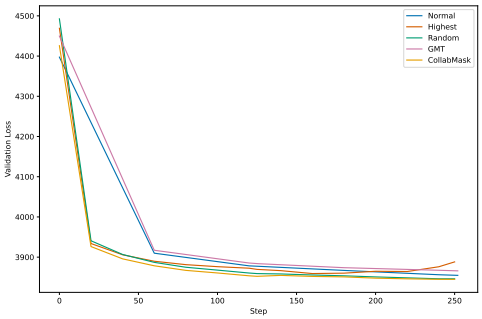

Figure 4: Validation perplexity with training steps on Llama-3-8B for GSM8K.

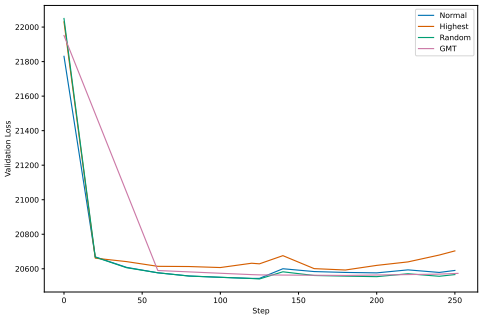

Figure 5: Validation perplexity with training steps on Llama-3-8B for continual learning on CodeAlpaca.

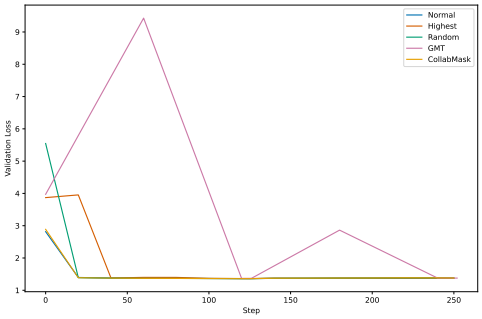

Figure 6: Validation perplexity with training steps on Mistral-7B for GSM8K.