

Table 3: Model Parameters and Sparsity Settings Across Different Models.

Model	Size	# Blocks	≈10% Sparsity	# Removed	# Parameters	≈20% Sparsity	# Removed	# Parameters	≈30% Sparsity	# Removed	# Parameters
LLaMA-2-7B	6.7b	32	9.01%	3	6.1b	21.02%	7	5.3b	30.03%	10	4.7b
LLaMA-2-13B	13b	40	9.75%	4	11.7b	19.49%	8	10.5b	29.24%	12	9.2b
LLaMA-3-8b	8b	32	10.86%	4	7.2b	19.01%	7	6.5b	29.88%	11	5.6b
Vicuna-7b	6.7b	32	9.18%	3	6.1b	21.02%	7	5.3b	30.03%	10	4.7b
Mistral-7B	7.2b	32	9.04%	3	6.6b	21.08%	7	5.7b	30.12%	10	5.1b
Qwen-2.5-7b	7.6b	28	9.18%	3	6.9b	21.42%	7	6.0b	30.60%	10	5.3b
Qwen-3-4b	4.0b	36	10.04%	4	3.6b	20.07%	8	3.22b	30.11%	12	2.8b

## A The Details of Comparison Methods

This paper compares three types of structured pruning paradigms: (1) DLP, which includes Mag+ [16], Taylor+ [16], ShortGPT [24], and SLEB [34]; (2) WSLP, which includes LaCo [42] and MKA [22]; and (3) methods combining layer pruning with post-training, such as LLM-streamline [5] and FuseGPT [27]. We implemented each method using publicly available code.

**Magnitude+ (Mag+).** [8] Kim et al. [16] use this method as a baseline in the pruning method comparison conducted. Initially proposed by Li et al. [19], it assumes that weights with smaller norms contain less information. For block-level analysis, the importance of the  $l$ -th layer is calculated as the sum of the first-order norms of all weight parameters:  $I_{Mag+,l} = \sum_{W \in \mathcal{W}^{(l)}} \sum_{w \in W} |w|$ , where  $\mathcal{W}^{(l)}$  represents the set of all weight matrices in the  $l$ -th layer. We follow popular heuristic algorithms [7, 13]. Kim et al. [16] further mitigated performance degradation by retaining the first four and last two layers [23].

**Taylor+.** [4] This method is also a baseline in the pruning method comparison by Kim et al. [16]. It assumes that the error introduced by removing weight parameters indicates their importance. Given a calibration dataset  $\mathcal{D}$ , this error can be expressed as a change in the training loss  $\mathcal{L}$ :  $|\mathcal{L}(W; \mathcal{D}) - \mathcal{L}(W = 0; \mathcal{D})| \approx \frac{\partial \mathcal{L}(\mathcal{D})}{\partial W} W$ . Following Kim et al. [16] and Ma et al. [23], we define the layer importance parameter as  $I_{Taylor+,l} = \sum_{W \in \mathcal{W}^{(l)}} \sum_{w \in W} |\frac{\partial \mathcal{L}(\mathcal{D})}{\partial w} w|$ . We use similar heuristic optimization methods to retain the first four and last two layers.

**ShortGPT.** [4] Proposed by Men et al. [24], this method assumes redundancy in the model layers and defines redundancy as layers that minimally alter the hidden embeddings. To measure change, they use the cosine distance as a metric, as shown in Eq. (2). Men et al. [24] use the PG19 long-document dataset for calibration, and we control the size of the calibration dataset to 256 samples. The results are consistent with those reported by Men et al. [24]. Men et al. [24] compute the importance scores for models in the LLaMA family, and we directly use the layer importance order provided in the paper. When the number of pruned layers exceeds the required number, we append the least important layers from our calculated importance scores. For models where Men et al. [24] do not provide layer importance sorting, we estimate it using the settings in this paper.

**SLEB.** [5] Song et al. [34] propose using a posterior method to verify the redundancy of specific layers. SLEB uses the exponential part of the PPL score of the pruned model on a specified dataset as the redundancy score:  $I_{SLEB,l} = \sum_{X \in \mathcal{D}} -\frac{1}{K} \sum_{i=0}^n \log p_{M'_l}(x_i | x_{<i}, x_i \in X)$ , where  $M'$  denotes the smaller model obtained from previous pruning steps,  $M'_l$  denotes the model obtained after pruning the  $l$ -th layer, and  $X = x_1, \dots, x_i, \dots, x_n$  represents a sample.  $I_{SLEB,l}$  is the exponential part of the PPL score, which positively correlates with the PPL score; hence,  $I_{SLEB}$  positively correlates with the PPL score. The SLEB method is a progressive structural search optimized for the PPL on the specified dataset.

**LaCo.** [6] Proposed by Yang et al. [42], this method uses the weight differences between layers as important information for layer retention. LaCo groups several adjacent layers and performs a Reserving-Differences-while-Seeking-Common layer merge. For weight fusion from layer  $l$  to  $l+m$ , the fused weight is represented as  $W^* = W^{(l)} + (W^{(l+1)} - W^{(l)}) + \dots + (W^{(l+m)} - W^{(l)}) = W^{(l)} + \sum_i^m (W^{(l+i)} - W^{(l)})$ . It fuses the differences between deeper and shallow layers into the shallow layers. LaCo assesses the redundancy of pruned groups using the cosine similarity of output

<sup>3</sup><https://github.com/Nota-NetsPresso/shortened-llm>

<sup>4</sup><https://github.com/sramshetty/ShortGPT>

<sup>5</sup><https://github.com/jiwonsong-dev/SLEB>

<sup>6</sup><https://github.com/yangyifei729/LaCo>

Table 4: The layer importance ranking of different DLP methods.

Model Sparsity	LLaMA-2-7B 9.0% / 21.0% / 30.0%	LLaMA-2-13B 9.8% / 19.5% / 29.2%	LLaMA-3-8b 10.9% / 19.0% / 29.9%
Mag+	7, 6, 11 / 8, 4, 10, 9 / 12, 14, 13	4, 5, 6, 7 / 10, 8, 9, 13 / 12, 11, 14, 16	5, 8, 7, 11 / 4, 6, 10 / 9, 13, 12, 14
Taylor+	29, 28, 27 / 26, 21, 25, 23 / 24, 19, 20	37, 35, 34, 36 / 33, 28, 26, 29 / 32, 27, 31, 25	29, 28, 26, 25 / 19, 27, 23 / 24, 20, 18, 22
ShortGPT	27, 26, 25 / 28, 24, 29, 23 / 21, 22, 30	33, 31, 32, 30 / 29, 34, 28, 35 / 27, 26, 36, 37	25, 27, 26, 24 / 28, 23, 22 / 29, 21, 20, 19
SLEB	14, 23, 11 / 24, 10, 27, 15 / 21, 25, 8	33, 29, 12, 13 / 26, 31, 14, 32 / 11, 10, 25, 35	10, 26, 11, 12 / 9, 23, 19 / 22, 25, 8, 7
FuseGPT-MI	11, 8, 27 / 24, 22, 14, 21 / 10, 13, 23	33, 29, 12, 10 / 27, 35, 31, 30 / 15, 28, 16, 25	10, 26, 25, 11 / 9, 8, 19 / 22, 7, 23, 20

Model Sparsity	Vicuna-7b 9.0% / 21.0% / 30.0%	Mistral-7B 9.0% / 21.1% / 30.1%	Qwen-2.5-7b 9.2% / 21.4% / 30.6%	Qwen-3-4b 10.0% / 20.1% / 30.1%
Mag+	7, 6, 11 / 8, 9, 10, 4 / 12, 14, 13	4, 6, 5 / 12, 7, 9, 10 / 11, 8, 13	9, 14, 17 / 16, 15, 13, 7 / 12, 6, 10	21, 19, 20, 18 / 22, 17, 15, 16 / 14, 23, 9, 13
Taylor+	29, 26, 21 / 27, 24, 25, 23 / 22, 19, 20	16, 28, 15 / 17, 29, 14, 13 / 22, 18, 12	4, 5, 21 / 22, 20, 23, 18 / 19, 17, 16	26, 25, 27, 29 / 28, 24, 23, 22 / 21, 30, 20, 31
ShortGPT	27, 25, 28 / 29, 24, 26, 23 / 22, 21, 30	25, 26, 24 / 27, 22, 23, 28 / 21, 29, 30	16, 17, 15 / 14, 12, 13, 18 / 11, 25, 24	29, 26, 27, 31 / 32, 33, 28, 25 / 20, 16, 18, 30
SLEB	10, 27, 14 / 23, 11, 12, 24 / 13, 9, 26	14, 13, 15 / 27, 22, 8, 24 / 23, 11, 21	16, 15, 17 / 14, 13, 18, 12 / 11, 10, 9	16, 15, 14, 17 / 18, 2, 19, 32 / 21, 26, 11, 30
FuseGPT-MI	12, 27, 11 / 23, 10, 25, 24 / 21, 9, 8	13, 10, 14 / 11, 8, 27, 23 / 22, 26, 25	16, 19, 17 / 18, 21, 14, 15 / 22, 10, 13	16, 17, 15, 2 / 14, 20, 21, 18 / 10, 26, 32, 11

features between the pruned and unpruned models:  $I_{LaCo} = \frac{1}{N} \sum_{X \in \mathcal{D}} \frac{H_M^{(L)\top} H_{M'}^{(L')}}{\|H_M^{(L)}\|_2 \|H_{M'}^{(L')}\|_2}$ , where  $H_M^{(L)}$  and  $H_{M'}^{(L')}$  represent the output features of the last layer of the model. Due to the threshold adjustment for cosine similarity in LaCo and the need to adjust the starting and ending layers for pruning, as well as the number of layers in each group, the excessive parameter settings made it challenging to optimize performance for each model. Therefore, we implement this method only on models in the LLaMA-2 family.

**MKA.** [7] Proposed by [22], this method uses manifold learning and the Normalized Pairwise Information Bottleneck (NPIB) measures to assess layer similarity and fusion. MKA progressively fuses deeper into shallower layers, merging the last two adjacent layers each time. In the code implementation, we find that MKA calculates the NPIB scores for two layers as approximately equal:  $I_{NPIB,l} : I_{NPIB,l+1} \approx 0.5 : 0.5$ . An exponential mapping increases the fusion proportion of the shallower  $l$ -th layer:  $I_{MKA,l} = \frac{e^{I_{norm}}}{1 - e^{I_{norm}}}$ , where  $I_{norm} = \frac{I_{NPIB,l}}{I_{NPIB,l} + I_{NPIB,l+1}}$  is the normalized NPIB score:  $I_{MKA,l+1} = 1 - I_{MKA,l}$ . After mapping, the similarity ratio between the two layers approaches  $I_{MKA,l} : I_{MKA,l+1} \approx 0.6 : 0.4$ .

**FuseGPT.** [8] Proposed by Pei et al. [27], FuseGPT hypothesizes that layer pruning causes performance loss and uses FFN parameter fusion to integrate layer capabilities into adjacent blocks, as Pei et al. [27] hypothesizes that FFN layers concentrate the main capabilities. Low-rank learnable weight matrices disperse the capabilities of pruned layers, optimizing multiple layers at once to reduce the gap caused by pruning. To better study the effectiveness of fusion, we remove the parameter adjustment part of FuseGPT in pure pruning experiments, using randomly initialized low-rank matrix products to fuse weights. In post-training comparison experiments, we use the complete FuseGPT method. Pei et al. [27] propose a Macro Influence (MI) score to measure the global-level impact of

removing a model layer:  $I_{MI} = 1 - \frac{1}{N} \sum_{X \in \mathcal{D}} \frac{H_M^{(L)\top} H_{M'}^{(L')}}{\|H_M^{(L)}\|_2 \|H_{M'}^{(L')}\|_2} = 1 - I_{LaCo}$ .

**LLM-streamline.** [9] Proposed by Chen et al. [5], LLM-streamline uses *SBI* (Eq. (3)) to measure redundancy of multiple consecutive layers, replacing these layers with the shallowest layer among them, and fine-tuning this shallowest layer post-training to restore model performance.

## B The Details of Experiment Setting

The settings for the experiment methods follow mainly those in the original papers. All experiments are conducted using an A100-40G GPU. We conducted pruning experiments on LLaMA-2-7b [10]

<sup>7</sup><https://github.com/SempraETY/Pruning-via-Merging>

<sup>8</sup><https://github.com/jarvispei/fusegpt>

<sup>9</sup><https://github.com/RUCKBReasoning/LLM-Streamline>

<sup>10</sup><https://huggingface.co/meta-llama/Llama-2-7b-hf>

Table 5: Experimental setting for pruning methods. † idenote methods whose hyperparameters were adjusted to satisfy the sparsity ratio constraints in our implementation. Complete implementation details are documented in Subsection B.1.

Methods	Calibration	# data	seed
Mag+	Wiki2	128	10
Taylor+	Wiki2	128	10
ShortGPT†	PG19	256	10
SLEB	Wiki2	128	10
FuseGPT	Wiki2	32	10
MKA†	MMLU	50 subtask * 5	10
LaCo†	Mouron () is a commune in the Arde		
	Torreorgaz is a municipality in the		
	The 81st Mechanised Brigade () is a mechanised brigade of the Romanian Land Force		
	There are 18 National Natural Landmarks in the U.S. state of Washington, out of nearly		
CoMe	Copa Libertadores 1973 was won by defending champions Independiente of A		
	Wiki2	256	10

Table 6: The Hyper-parameter used in LaCo [42].  $C$  is the number of layers to be merged during each merging optimization.  $\mathcal{I}$  is the minimum interval of layers between two merging operations.  $\mathcal{L}$  and  $\mathcal{H}$  are the minimum and maximum indices of the range of layers for merging.  $\mathcal{T}$  is a similarity threshold.

	Sparsity	$C$	$\mathcal{L}$	$\mathcal{H}$	$\mathcal{I}$	$\mathcal{T}$
LLaMA-2-7b	9.01%	4	1	32	2	0.85
	21.02%	8	1	32	2	0.65
	30.03%	6	1	32	2	0.55
LLaMA-2-13b	9.75%	5	1	40	2	0.85
	19.49%	5	1	40	2	0.70
	29.24%	5	1	40	2	0.55

LLaMA-2-13b<sup>[11]</sup>, LLaMA-3-8b<sup>[12]</sup>, Vicuna-7b<sup>[13]</sup>, Mistral-7b<sup>[14]</sup>, Qwen-2.5-7b<sup>[15]</sup>, and Qwen-3-4b<sup>[16]</sup> and performed post-training experiments on LLaMA-2-7b and Qwen-3-4b.

We modify some settings based on the original implementations and develop an open-source project with multiple pruning methods. Our project code can be found at <https://github.com/MPI-Lab/CoMe>.

## B.1 Implementation of Pruning Methods

Tab. 5 shows the calibration datasets, the dataset number, and the random seeds used in the pruning methods. Tab. 4 presents the pruned layers’ index order for the pruning method. We implement the Mag+, Taylor+, and SLEB using our reproduced code.

For the ShortGPT method, we follow the layer BI score for LLaMA-2-7B provided in the original article. For the LLaMA-2-13b model, the original paper provides only the pruning order for the first 10 layers. We use the open-source project reproduction code to calculate the remaining layers’ BI scores and place the two with the smallest BI scores at the end of the given pruning order. For other models, we obtain the BI scores for each layer entirely through the reproduction method. PG19 is a long-document dataset, and the training set contains 28,602 training samples. Using all samples to get the model’s BI scores would consume significant training resources, so we randomly selected 256 training samples from PG19 for calibration. Even with a small amount of data, the ShortGPT method takes much longer to calculate BI scores than other methods.

<sup>11</sup><https://huggingface.co/meta-llama/Llama-2-13b-hf>  
<sup>12</sup><https://huggingface.co/meta-llama/Meta-Llama-3-8B>  
<sup>13</sup><https://huggingface.co/lmsys/vicuna-7b-v1.5>  
<sup>14</sup><https://huggingface.co/mistralai/Mistral-7B-v0.1>  
<sup>15</sup><https://huggingface.co/Qwen/Qwen2.5-7B>  
<sup>16</sup><https://huggingface.co/Qwen/Qwen3-4B-Base>

When selecting MMLU data, MKA randomly samples five samples from 50 sub-tasks. In our implementation, we uniformly sample 250 samples from each sub-task. We reproduce the experimental results for the LLaMA family using the original MKA code, while we obtain the sparse results for other models using our reproduced code. The Qwen-2.5-7b model contains bias weights, and fusing these weights would degrade the performance of the pruned model, so we do not fuse the bias weights.

The calibration samples for the LaCo method are sourced from the open-source project code, and we fully reproduce the process using the original code. To achieve the number of pruned layers consistent with the settings in this paper, we make simple parameter adjustments to the LaCo method, with the detailed parameter settings shown in Tab. 6. For other models, adjusting the LaCo code is too complex, so we do not reproduce it.

The FuseGPT method is implemented using the original code. To compare different categories of methods, we comment on the post-training code of FuseGPT for the pruning method comparison experiment. In the pure pruning method comparison experiment, FuseGPT-MI+F means that we mask the post-training code, while FuseGPT-MI implies that we additionally mask the fusion code. Implementing this method on the LLaMA-2-13b model with an NVIDIA A100-40G GPU resulted in a memory overflow, so we do not implement it.

In the layer pruning process of CoMe, we fuse two layers of the model per iteration, meaning that we reduce one layer per iteration. When pruning models from the LLaMA family, Vicuna-7b and Mistral-7b, the hyperparameter  $p$  is set to 1. For the Qwen2.5-7b model,  $p$  is set to 32. The Mistral-7b, Qwen2.5-7b, and LLaMA-3-8b models have high knowledge density and less redundancy in parameters, making them very sensitive to hyperparameter settings. To further mitigate performance degradation caused by merging channels with different distributions, we set a minimum parameter retention ratio  $\rho$ , meaning the proportion of parameters from the more critical layer cannot be less than  $\rho$  during the fusion of two layers. The values of  $\rho$  for the Mistral-7b, Qwen2.5-7b, and LLaMA-3-8b models are set to 0.97, 0.85, and 0.97, respectively.

Table 7: Experimental setting for post-training methods.

Method	# Iterations	# Epochs	# Steps	Batch size	Token length
FuseGPT	10	20	128	8	2048
LLM-Streamline	1	5	938	32	2048
CoMe-mp	7	1	2000	32	512
CoMe-sp	1	1	10000	32	512

## B.2 Implementation of Post-Training

Tab. 7 summarizes the post-training settings for all methods. Based on these settings, we quantify the resource consumption of each method by calculating the total number of tokens required to train a single layer, which we denote as  $T_{layer}$ . This metric is computed as follows:

$$T_{layer} = \# \text{ Iterations} \times \# \text{ Layers} \times \# \text{ Epochs} \times \# \text{ Steps} \times \text{Batch size} \times \text{Token length}, \quad (9)$$

where “# Layers” indicates the number of layers updated in each iteration.

The post-training process for the FuseGPT method is synchronized with the pruning process, utilizing 1,024 samples from the Wiki-2 dataset, in accordance with the settings of Pei et al. [27]. In each iteration, the parameters of one layer are merged into seven adjacent layers. Pruning ten layers requires ten iterations, with seven layers updated in each iteration. For FuseGPT,  $T_{layer} \approx 2.93B$ .

The LLM-Streamline trains a merged layer using 30,000 samples and employs five epochs, following the settings of Chen et al. [5]. For LLM-Streamline,  $T_{layer} \approx 0.31B$ .

We carry out the post-training process of CoMe after completing the pruning process. After pruning 10 layers, the pruned model has seven layers corresponding to multiple layers of the original model. Therefore, in CoMe-mp, there are seven training iterations requiring minimal training resources. CoMe-sp trains seven layers in one training round, requiring more training resources. For optimization, we utilize the AdamW optimizer with a weight decay coefficient of  $1e - 2$  and implement cosine decay for learning rate scheduling. The CoMe-sp employs a fixed learning rate of  $1e - 5$ . The

CoMe-mp adopts layer-specific decaying rates during multi-layer distillation, with learning rates progressively decreasing from the shallow to the deep layers as follows:  $5e-4$ ,  $2.5e-4$ ,  $1e-4$ ,  $7.5e-5$ ,  $5e-5$ ,  $2.5e-5$ , and  $1e-5$  for LLaMA-2-7b;  $5e-4$ ,  $2.5e-4$ ,  $5e-5$ ,  $2.5e-5$ ,  $1e-5$ , and  $7.5e-6$  for Qwen-3-4b. For CoMe-mp,  $T_{layer} \approx 0.23B$ . For CoMe-sp,  $T_{layer} \approx 1.15B$ .

## C Channel importance and Concatenation-based Merge

We use the channel importance for parameter division in the concatenation-based merge strategy; thus, we need to analyze the channel importance calculation for different transformer parts. Ma et al. [23] highlight that in transformer-based models, a certain correspondence exists in feature dimensions during forward propagation due to residual connections. For instance, the positional correspondence of output features from Norm, MHA, and FFN is fixed.

In the Norm part, we use the weights to scale the feature inputs. Xiong et al. [40] note that the parameters of deep layer Norms need significant enlargement to stabilize training, which is closely related to the distribution of input features. Our objective is to minimize changes in the output features of each module; therefore, we average the Norm parts of adjacent layers to maintain stability, as  $\bar{\gamma} = \frac{1}{m+1} \sum_l^{l+m} \gamma^{(i)}$ , where  $m+1$  denotes the number of merge layers.

The MHA module generates three feature vectors: Query, Key, and Value. These vectors are concatenated and undergo matrix multiplication for cross-information fusion. Consequently, the weights within the heads used to generate Query, Key, and Value are tightly coupled, making it difficult to make finer divisions. Thus, we consider each head in MHA the basic unit for concatenation. We ignore the coupling between heads to further simplify the calculation of channel importance. Pruning a single head structure reduces the input dimension of the *o\_project* weight (using the transformer structure in LLaMA as an example), leading to changes in the MHA output. We take the average channel importance of the reduced dimensions as the importance corresponding to each head structure.

The FFN module usually contains three weight matrices: *up\_project*, *gat\_project*, and *down\_project*. By neglecting the coupling caused by activation functions, the information loss from channel weight pruning in *up\_project* and *gat\_project* maps to a reduction in intermediate feature dimensions. Therefore, we use the intermediate features and *down\_project* to calculate channel importance.

## D Posterior-based CoMe

When applying CoMe to different model architectures, it is often necessary to adjust the hyperparameter  $p$  to control the parameter preservation ratio. However, the optimal ratio can vary significantly across models, which reduces the convenience and usability of CoMe. Inspired by SLEB [34] and LaCo [42], we propose an adaptive, posterior-based strategy for determining the parameter preservation ratio within CoMe, referred to as Posterior-based CoMe (CoMe-P).

CoMe-P replaces the parameter preservation ratio calculation in the layer merging process of CoMe (Eq. (5)) with a posterior-driven approach. Specifically, consider the case of merging two adjacent layers in an iteration. Let the parameter preservation ratio of the lower-indexed layer be  $r$ , and that of the other layer be  $1-r$ . We define a candidate set for  $r$  as  $\Gamma = \{\frac{i}{n} \mid i = 0, 1, 2, \dots, n\}$ , where  $n$  determines the granularity of the search. CoMe-P iteratively applies different preservation ratios from  $\Gamma$  to generate compressed models, evaluating each candidate model on a calibration dataset using the PPL metric. The compressed model yielding the lowest PPL is selected for the final merging. The detailed algorithm of CoMe-P is presented in Alg. 2.

We set  $n = 20$ , with all other parameters kept consistent with the default settings of CoMe. Tabs. 8 to 10 present comparisons between CoMe-P and other methods across different models. CoMe-P achieves performance comparable to CoMe, and yields higher average accuracy and lower PPL on Qwen3-4b, Vicuna-7b, and Mistral-7B, demonstrating the effectiveness of the posterior-based approach. However, since CoMe-P is a posterior search method, the search space grows exponentially when merging more than two layers in each iteration, resulting in exponentially increased resource consumption.

## E Analysis of CoMe-mp and CoMe-sp

To evaluate the effectiveness of CoMe-mp and CoMe-sp during the post-training process, we examine the cross-entropy loss between the student and teacher models, as shown in Fig. 12. When using CoMe-sp, the cross-entropy loss converges rapidly and stabilizes within the first 4000 steps. This phenomenon indicates an effective alignment of feature representations, as the hierarchical distillation strategy facilitates rapid convergence. In contrast, CoMe-mp shows a more linear convergence pattern, suggesting that aligning features layer by layer significantly enhances the student model’s performance. However, because shallow features require processing by deeper layers, training one layer at a time results in slower convergence.

Subsection B.2 details the number of tokens used during the post-training phase. Although CoMe-sp uses fewer tokens, it requires updating seven times more parameters per step than CoMe-mp, necessitating greater memory resources. The overhead of memory resources is due to CoMe-sp’s simultaneous optimization of multiple layers, which, while resource-intensive, allows for more efficient global information updates compared to the sequential approach of CoMe-mp.

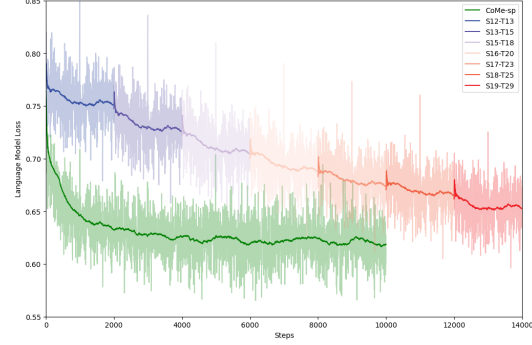


Figure 12: Cross Entropy loss curves for using CoMe-mp and CoMe-sp during post-training. The loss curves for multiple subprocesses of CoMe-mp are concatenated in the order of training.

## F Sum-based Merge vs. Concatenation-based Merge?

**The role of Weight Sum-based Merge in both pruning and post-training processes is LESS.** Tab. 8 and Tab. 11 provide the effects of using parameter fusion (Fusion-MI+F) and only layer pruning without parameter fusion (Fusion-MI) in the FuseGPT method when pruning the LLaMA-2-7b model. In both pruning and post-training, the average accuracy score differences do not exceed 1.2 points, and PPL score differences do not exceed 0.5 points. Moreover, in pruning experiments, the performance of parameter fusion methods is worse when 20% of the parameters are pruned. It indicates that additive inter-layer parameter fusion is ineffective. In all experiments, Fusion-MI+F does not significantly improve pruning performance compared to Fusion-MI.

**The Weight Sum-based Merge does not exhibit significant differences from the DLP methods, but the Concatenation-based Merge can improve the performance of DLP methods.** In Tab. 2, after removing parameter fusion, the MKA method exhibits only a slight decrease in average accuracy scores and a slight increase in PPL, which is almost negligible. With the removal of parameter fusion, LaCo shows a slight rise in average scores and a significant decrease in PPL, indicating that the parameter fusion has a negative impact. For FuseGPT, removing parameter fusion results in a notable reduction on some datasets, such as HellaS and MMLU, a slight decrease in PPL on the C4 dataset, and a slight increase on the Wiki-2 dataset. It is difficult to conclude that parameter fusion further enhances model performance beyond layer pruning, but previous analyses suggest that FuseGPT has a minimal effect. The Weight Sum-based Merge method does not significantly differ from Direct layer pruning methods. However, when CoMe removes parameter fusion, it shows a noticeable performance decline across all test benchmarks, except for the MMLU dataset, with significant increases in PPL on the Wiki-2 and C4 datasets. It strongly indicates that Concatenation-based Merge can further enhance model performance based on DLP.

**Concatenated-based Merge is Effective, but Weight Sum-based Merge is NOT.** In Figs. 3, 11, 13 and 14, we apply both  $\alpha A + (1 - \alpha)B$  (WSLP) and Concatenation-based Merge to blend parameters of two layers in varying proportions. The  $\alpha$ -Add method, whether merging the Self-Attention structure, the FFN structure, or the entire model layer, consistently results in a significantly increased PPL on the Wiki-2 and C4 datasets. It shows that the Weight Sum-based Merge method harms model performance, degrading performance as the fusion ratio approaches equality. Conversely, the



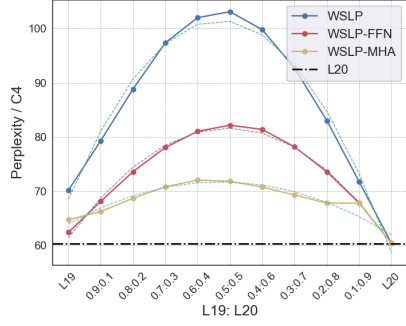


Figure 13: Merge adjacent layers with linear weight aggregation at different ratios, using the C4 calibration dataset.

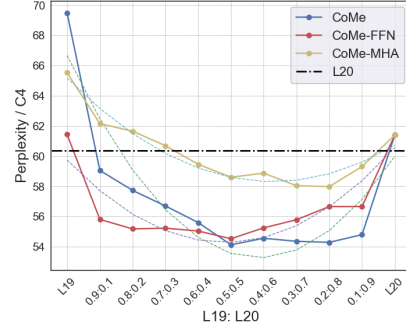


Figure 14: Merge adjacent layers with CoMe at different ratios, using the C4 calibration dataset.

Concatenation-based Merge method can reduce PPL at specific fusion ratios, preserving the model’s language modeling ability.

## G Detailed Experimental Results

In this section, we present comprehensive experimental data. The specific outcomes corresponding to Fig. 6 are detailed in Tabs. 8 to 10. Additionally, the results associated with Figs. 7 to 10 are thoroughly documented in Tabs. 11 to 14, respectively.

Table 8: The Layer Pruning Experiment on the LLaMA Family.

Model	Sparsity	Methods	Benchmark↑							Avg↑		RP↑	C4	PPL↓	
			ARC-c	ARC-e	HellaS	OBQA	PIQA	WinoG	MMLU (5)	Wiki-2					
LLaMA-2-7b	9.0%	Dense	46.33	74.54	75.99	44.20	79.05	69.06	45.60	62.11	100.00		7.27	5.47	
		Mag+	37.97	66.20	67.65	39.80	76.44	59.83	26.79	53.53	84.56		9.21	7.01	
		Taylor+	42.06	69.07	73.00	42.00	75.68	68.19	42.22	58.89	94.51		10.21	7.74	
		ShortGPT	43.00	68.77	71.61	40.40	76.44	68.67	45.56	59.21	95.25		9.33	7.43	
		SLEB	38.57	65.82	70.69	39.80	77.26	63.46	33.83	55.63	88.35		8.71	6.47	
		FuseGPT-MI	39.68	68.86	70.11	40.40	77.48	61.88	31.98	55.77	88.49		8.73	6.55	
		FuseGPT-MI+F	39.51	65.57	71.03	40.80	77.37	62.90	32.13	55.62	88.35		8.73	6.47	
		MKA	44.54	64.98	67.09	37.80	72.80	62.43	45.64	56.47	91.39		44.50	25.41	
		LaCo	43.43	68.60	71.78	40.60	76.39	68.51	45.39	59.24	95.35		9.38	7.46	
		CoMe	44.11	70.96	73.85	42.00	77.04	68.19	46.04	60.31	97.11		8.58	6.23	
	21.0%	Mag+	24.32	44.11	40.23	31.00	65.72	53.12	24.58	40.44	64.10		37.36	49.17	
		Taylor+	36.09	56.52	61.15	37.80	69.04	65.11	41.58	52.47	84.65		23.91	18.77	
		ShortGPT	36.26	55.85	62.62	37.20	70.40	66.30	39.85	52.64	84.60		23.31	18.45	
		SLEB	33.02	56.52	62.51	36.80	73.07	58.96	26.26	49.59	78.29		12.33	9.15	
		FuseGPT-MI	34.64	58.25	64.10	37.00	73.67	57.14	26.08	50.13	79.16		12.22	9.46	
		FuseGPT-MI+F	33.96	56.69	61.28	35.80	73.78	56.27	24.76	48.93	77.16		12.12	9.14	
		MKA	37.46	54.92	53.61	37.40	66.27	58.88	42.96	50.21	81.86		388.57	247.36	
		LaCo	26.79	49.87	52.69	33.80	71.55	55.88	24.77	45.05	70.90		18.62	15.85	
		CoMe	39.59	64.10	68.68	39.80	72.42	67.25	32.82	54.95	87.55		13.02	9.55	
		CoMe-P	34.64	54.55	58.35	35.40	67.95	61.09	26.89	48.41	76.89		18.87	14.74	
	30.0%	Mag+	23.98	39.31	35.77	27.20	61.75	51.70	22.96	37.52	59.49		52.39	59.73	
		Taylor+	32.68	46.04	51.58	31.60	63.87	62.67	42.89	47.33	76.75		63.08	50.96	
		ShortGPT	31.91	47.39	45.96	34.80	63.28	61.48	38.66	46.21	75.07		54.92	49.56	
		SLEB	30.80	51.81	54.07	32.80	68.44	54.14	25.10	45.31	71.62		17.43	13.84	
		FuseGPT-MI	30.20	50.59	52.98	33.60	69.37	54.54	25.17	45.21	71.53		17.60	14.94	
		FuseGPT-MI+F	30.20	50.13	55.08	34.80	68.50	55.41	27.01	45.88	72.82		17.80	14.34	
		MKA	34.04	49.58	48.12	35.00	63.00	59.12	35.64	46.36	75.14		810.04	455.34	
		LaCo	30.97	49.79	50.14	35.00	68.34	53.91	24.84	44.71	71.11		39.18	42.67	
		CoMe	35.24	54.46	56.56	35.40	68.88	61.17	25.50	48.17	76.47		19.93	16.53	
		CoMe-P	34.64	54.55	58.35	35.40	67.95	61.09	26.89	48.41	76.89		18.87	14.74	
LLaMA-2-13b	9.8%	Dense	49.15	77.53	79.39	45.20	80.52	72.14	55.16	65.58	100.00		6.73	4.89	
		Mag+	35.58	62.84	59.88	36.20	73.34	59.35	25.76	50.42	75.57		17.47	15.38	
		Taylor+	45.90	70.71	76.52	42.20	78.62	72.30	43.16	61.34	92.92		9.57	7.47	
		ShortGPT	47.61	72.85	76.62	45.00	79.54	71.74	54.54	63.99	97.71		8.05	5.78	
		SLEB	42.49	72.31	74.11	44.00	79.27	65.51	42.64	60.05	91.00		7.81	5.64	
		FuseGPT-MI	40.96	69.02	74.48	44.00	79.11	68.82	42.52	59.84	90.61		7.83	5.65	
		MKA	47.01	69.07	69.42	45.00	74.81	65.35	54.02	60.67	93.31		34.44	29.88	
		LaCo	46.50	74.07	76.86	44.20	78.94	72.45	54.81	63.98	97.51		8.37	6.05	
		CoMe	47.53	75.25	78.23	43.20	79.49	71.98	55.34	64.43	98.10		7.61	5.36	
		CoMe-P	47.53	75.25	78.23	43.20	79.49	71.98	55.34	64.43	98.10		7.61	5.36	
	19.5%	Mag+	23.12	46.42	37.81	29.80	65.83	50.91	23.65	39.65	59.38		125.08	228.40	
		Taylor+	43.26	65.66	72.09	40.80	75.30	70.48	47.24	59.26	90.09		13.37	12.12	
		ShortGPT	43.94	67.34	72.39	41.00	75.24	69.69	53.83	60.49	92.25		11.36	8.30	
		SLEB	37.88	64.65	70.59	42.40	76.82	64.64	32.32	55.61	83.83		9.47	6.85	
		FuseGPT-MI	38.91	63.72	70.97	40.60	76.93	67.32	42.57	57.29	86.66		9.69	7.04	
		MKA	40.61	59.60	57.18	41.40	68.93	62.90	53.04	54.81	84.58		219.41	206.12	
		LaCo	34.81	54.97	64.67	39.20	74.32	63.61	23.51	50.73	76.14		13.04	10.86	
		CoMe	45.14	72.52	75.87	42.80	76.50	70.80	50.35	62.00	94.30		9.17	6.29	
		CoMe-P	45.14	72.52	75.87	42.80	76.50	70.80	50.35	62.00	94.30		9.17	6.29	
		CoMe-P	45.14	72.52	75.87	42.80	76.50	70.80	50.35	62.00	94.30		9.17	6.29	
	29.2%	Mag+	23.12	33.16	30.27	25.60	56.15	52.17	25.37	35.12	53.23		317.35	593.77	
		Taylor+	38.91	54.97	62.24	37.20	70.73	69.61	48.20	54.55	83.21		23.96	28.38	
		ShortGPT	35.75	52.82	57.94	38.20	69.91	69.06	47.78	53.07	81.08		29.37	39.61	
		SLEB	34.04	58.59	63.38	38.60	75.35	62.35	26.75	51.29	76.94		11.64	8.69	
		FuseGPT-MI	37.29	56.90	64.80	36.60	74.43	65.04	30.78	52.26	78.61		12.65	9.46	
		MKA	36.95	53.07	48.67	36.00	65.56	60.46	50.72	50.20	77.39		759.76	632.28	
		LaCo	33.19	51.43	54.88	39.00	68.39	60.77	24.55	47.46	71.85		27.43	23.81	
		CoMe	42.49	67.05	69.87	42.60	73.34	68.98	51.17	59.36	90.67		12.64	8.85	
		CoMe-P	43.43	66.96	69.38	40.20	73.50	69.77	51.81	59.29	90.43		12.32	8.56	
		CoMe-P	43.43	66.96	69.38	40.20	73.50	69.77	51.81	59.29	90.43		12.32	8.56	
LLaMA-3-8b	9.2%	Dense	53.33	77.74	79.17	45.00	80.79	72.93	65.29	67.75	100.00		9.45	6.14	
		Mag+	34.73	64.02	49.37	36.00	74.16	54.78	25.35	48.34	70.80		25.93	20.44	
		Taylor+	47.70	70.92	66.81	40.20	76.01	72.69	30.57	57.84	85.00		20.58	14.88	
		ShortGPT	47.44	69.99	73.63	39.80	76.28	71.43	63.67	63.18	92.90		20.08	15.07	
		SLEB	41.30	67.47	69.05	39.00	77.53	64.33	30.03	55.53	81.18		13.68	8.85	
		FuseGPT-MI	42.83	70.29	70.79	38.80	77.80	69.14	65.20	62.12	91.05		13.65	8.93	
		FuseGPT-MI+F	40.10	67.26	68.47	38.20	76.66	62.43	29.97	54.73	79.93		13.72	8.95	
		MKA	44.71	63.43	62.75	41.20	72.96	64.09	63.68	58.97	87.42		307.03	191.80	
		CoMe	47.70	72.81	72.28	40.60	76.50	74.19	63.65	63.96	94.08		14.84	9.52	
		CoMe-P	47.70	72.81	72.28	40.60	76.50	74.19	63.65	63.96	94.08		14.84	9.52	
	21.4%	Mag+	25.77	46.04	43.40	30.40	65.29	53.20	25.16	41.32	60.32		43.24	40.83	
		Taylor+	31.91	43.60	35.50	32.20	60.17	58.80	33.08	42.18	62.58		1549.77	1294.94	
		ShortGPT	42.41	56.52	64.65	33.40	70.89	71.11	61.73	57.24	83.99		63.81	57.84	
		SLEB	35.75	58.42	62.29	34.80	73.83	57.85	27.43	50.05	72.99		18.67	13.38	
		FuseGPT-MI	34.56	59.81	59.03	34.00	74.37	56.67	50.43	52.70	76.98		19.38	13.44	
		FuseGPT-MI+F	33.70	53.91	61.17	35.80	72.74	57.93	26.42	48.81	71.33		19.03	13.42	
		MKA	42.58	60.10	55.90	40.40	68.88	62.04	59.27	55.60	82.66		1168.37	1004.27	
		CoMe	40.44	64.23	65.52	35.60	73.50	70.96	56.96	58.17	85.12		23.10	17.15	
		CoMe-P	40.44	64.23	65.52	35.60	73.50	70.96	56.96	58.17	85.12		23.10	17.15	
		CoMe-P	40.44	64.23	65.52	35.60	73.50	70.96	56.96	58.17	85.12		23.10	17.15	
	30.6%	Mag+	22.18	34.22	33.28	27.00	57.73	52.49	24.19	35.87	52.59		242.47	254.76	
		Taylor+	27.99	33.71	30.44	27.60	57.73	52.17	47.06	39.53	58.67		44214.20	50035.02	
		ShortGPT	30.20	38.13	32.89	30.20	59.09	56.75	41.88	41.31	61.35		7021.78	15660.69	
		SLEB	27.73	49.12	48.38	27.80	66.97	51.46	25.82	42.47	61.58		30.80	28.27	
		FuseGPT-MI	29.01	48.23	47.46	29.20	66.65	54.14	40.95	45.09	65.82		37.93	30.70	
		FuseGPT-MI+F	29.01	42.51	49.36	30.00	66.65	56.12	26.06	42.82	62.49		41.82	33.34	
		MKA	38.14	49.75	47.19	34.40	62.84	62.27	59.03	50.52	75.02		7447.81	5460.38	
		CoMe	33.70	50.67	50.42	31.20	67.08	60.62	30.99	46.38	67.86		48.85	43.55	
		CoMe-P	33.70	50.67	50.42	31.20	67.08	60.62	30.99	46.38	67.86		48.85	43.55	
		CoMe-P	33.70	50.67	50.42	31.20	67.08	60.62	30.99	46.38	67.86		48.85	43.55	



Table 9: The Layer Pruning Experiment on the Vicuna-7b and Mistral-7b.

Model	Sparsity	Methods	ARC-c	ARC-e	HellaS	Benchmark $\uparrow$ OBQA	PIQA	WinoG	MMLU (5)	Avg $\uparrow$	RP $\uparrow$	PPL $\downarrow$ C4	Wiki-2
Vicuna-7b	9.0%	Dense	45.90	71.30	73.78	45.00	78.02	69.46	49.89	61.91	100.00	9.19	6.78
		Mag+	38.40	64.35	66.04	40.00	74.21	59.27	33.11	53.63	85.59	11.59	8.54
		Taylor+	42.83	67.17	70.53	41.20	74.97	68.11	46.85	58.81	94.67	11.96	9.58
		ShortGPT	43.34	67.68	70.86	42.00	75.14	<b>69.85</b>	49.99	59.84	96.54	NaN	9.13
		SLEB	40.61	65.91	68.06	39.40	75.95	62.04	41.04	56.14	89.95	10.48	7.65
		FuseGPT-MI	41.30	<b>70.20</b>	69.76	39.60	<b>77.15</b>	64.40	41.41	57.69	92.23	10.48	7.69
		FuseGPT-MI+F	40.96	66.33	68.84	39.40	75.79	63.46	44.98	57.11	91.68	10.74	7.83
		MKA	<b>41.98</b>	64.94	66.55	38.40	71.44	65.51	<b>50.44</b>	57.04	92.15	72.33	43.02
		CoMe	43.86	69.53	<b>72.77</b>	<b>42.80</b>	75.46	69.06	49.57	<b>60.44</b>	<b>97.47</b>	<b>10.48</b>	<b>7.52</b>
		Mag+	25.34	44.91	40.64	31.20	64.58	52.33	27.17	40.88	65.03	52.21	68.98
		Taylor+	38.65	58.38	60.71	35.80	69.10	65.90	45.58	53.45	86.10	21.09	20.45
		ShortGPT	38.74	59.09	62.38	37.40	68.34	66.06	45.33	53.91	86.93	27.14	21.87
	21.0%	SLEB	36.26	61.83	61.66	35.80	<b>73.72</b>	59.59	28.31	51.02	80.84	13.88	<b>10.52</b>
		FuseGPT-MI	38.57	62.58	63.03	37.40	73.23	59.75	32.37	52.42	83.59	<b>13.71</b>	10.55
		FuseGPT-MI+F	36.69	59.72	61.28	35.00	73.07	59.35	29.19	50.61	80.31	14.78	10.73
		MKA	39.85	54.17	53.34	38.00	66.97	61.01	<b>50.61</b>	51.99	84.95	540.72	335.40
		CoMe	<b>40.96</b>	<b>64.52</b>	<b>66.49</b>	<b>42.40</b>	72.47	<b>68.11</b>	35.43	<b>55.77</b>	<b>89.43</b>	16.66	11.73
		Mag+	24.15	40.78	36.13	27.60	62.30	50.83	25.06	38.12	60.48	72.72	92.79
		Taylor+	33.96	46.55	49.17	31.60	60.88	62.12	32.13	45.20	72.57	62.01	183.82
		ShortGPT	33.11	48.40	48.95	34.60	64.25	62.90	41.04	47.61	76.92	61.91	59.84
		SLEB	32.00	56.10	53.01	33.00	<b>70.24</b>	56.20	24.38	46.42	73.34	NaN	NaN
		FuseGPT-MI	33.62	57.45	53.56	36.60	68.88	53.75	25.25	47.02	74.86	18.61	15.68
		FuseGPT-MI+F	33.36	53.11	53.02	34.00	68.28	55.09	25.07	45.99	73.09	NaN	19.85
		MKA	34.04	48.15	47.02	35.80	61.97	61.17	<b>47.99</b>	48.02	78.38	986.16	660.93
		CoMe	<b>36.35</b>	<b>58.84</b>	<b>56.48</b>	<b>42.40</b>	68.66	<b>62.98</b>	25.33	<b>50.15</b>	<b>80.28</b>	29.55	18.69
		CoMe-P	34.56	57.37	56.42	36.80	69.53	62.51	28.46	49.38	78.59	<b>22.32</b>	<b>15.97</b>
Mistral-7b	9.0%	Dense	54.01	79.50	81.06	44.00	82.05	74.03	62.52	68.17	100.00	8.38	5.25
		Mag+	32.68	60.82	55.67	36.20	72.52	58.88	27.15	49.13	71.33	20.33	13.59
		Taylor+	44.88	70.83	75.94	40.60	79.71	69.69	52.97	62.09	90.59	10.13	6.52
		ShortGPT	48.38	73.40	76.75	41.00	79.98	<b>72.77</b>	<b>62.26</b>	64.93	<b>95.02</b>	10.25	7.14
		SLEB	43.09	71.09	74.53	41.40	79.16	64.64	41.81	59.39	86.56	<b>9.76</b>	<b>6.21</b>
		FuseGPT-MI	45.39	72.10	74.88	<b>41.60</b>	<b>80.41</b>	64.80	41.41	60.08	87.63	9.80	6.25
		FuseGPT-MI+F	42.49	70.75	73.49	41.00	79.76	66.46	39.62	59.08	85.98	9.86	6.31
		MKA	43.43	61.03	53.31	41.20	67.41	62.75	58.10	55.32	82.35	274.34	203.48
		CoMe	<b>48.55</b>	<b>74.37</b>	<b>76.93</b>	41.00	79.60	72.61	61.52	<b>64.94</b>	95.00	10.04	6.52
		Mag+	23.12	38.47	33.44	25.60	59.85	52.09	23.53	36.59	53.08	876.24	1409.19
		Taylor+	35.24	54.46	64.30	33.80	73.72	61.64	25.05	49.74	71.87	19.68	15.34
		ShortGPT	40.44	57.66	64.53	32.80	72.14	<b>67.88</b>	<b>59.98</b>	56.49	82.44	33.21	24.01
	21.1%	SLEB	36.95	61.36	64.81	<b>39.00</b>	75.24	61.48	28.83	52.52	76.44	<b>13.55</b>	<b>9.25</b>
		FuseGPT-MI	35.07	58.88	65.71	37.00	<b>75.68</b>	55.96	24.98	50.47	73.13	14.16	10.00
		FuseGPT-MI+F	34.56	57.20	65.94	36.60	74.76	57.30	25.68	50.29	72.87	21.73	15.29
		MKA	35.84	42.47	40.42	33.80	58.49	57.14	53.98	46.02	68.75	36779.48	30245.43
		CoMe	<b>40.61</b>	<b>63.80</b>	<b>67.54</b>	36.20	74.21	67.64	53.50	<b>57.64</b>	<b>84.06</b>	14.33	10.01
		Mag+	25.26	32.07	30.74	28.20	54.08	50.28	25.03	35.09	51.86	253.79	288.66
		Taylor+	29.01	35.73	44.64	32.20	60.94	54.62	24.70	40.26	59.21	112.23	121.24
		ShortGPT	32.00	29.71	33.56	31.60	57.51	56.91	22.72	37.72	56.16	760.27	881.51
		SLEB	30.80	47.69	57.00	<b>34.20</b>	68.44	57.22	25.10	45.78	66.56	21.19	16.46
		FuseGPT-MI	32.25	51.68	56.23	33.80	70.78	52.96	25.92	46.23	67.17	82.31	47.95
		FuseGPT-MI+F	28.58	46.68	56.00	32.00	69.04	53.20	23.75	44.18	63.92	20.62	15.71
		MKA	32.34	35.90	32.46	30.80	54.95	54.70	25.70	38.12	56.72	33065.64	37735.00
		CoMe	31.48	51.85	55.81	31.00	68.77	58.56	27.47	46.42	67.10	22.93	18.32
		CoMe-P	<b>33.45</b>	<b>59.05</b>	<b>58.32</b>	31.00	<b>70.35</b>	<b>59.69</b>	<b>28.14</b>	<b>48.57</b>	<b>70.00</b>	<b>19.19</b>	<b>14.53</b>

Table 10: The Layer Pruning Experiment on the Qwen-2.5-7b and Qwen-3-4b.

Model	Sparsity	Methods	ARC-c	ARC-e	HellaS	Benchmark $\uparrow$			Avg $\uparrow$	RP $\uparrow$	PPL $\downarrow$	
						OBQA	PIQA	WinoG	MMLU (5)		C4	Wiki-2
Qwen-2.5-7b	9.2%	Dense	51.11	77.36	78.95	47.20	79.65	73.01	74.16	68.78	100.00	6.85
		Mag+	43.00	68.48	62.44	38.40	75.14	60.54	49.64	56.81	82.47	9.23
		Taylor+	47.70	71.84	67.62	37.00	73.01	<b>67.01</b>	<b>65.90</b>	61.44	88.93	11.16
		ShortGPT	46.59	<b>72.43</b>	72.27	44.00	79.00	64.88	55.88	<b>62.15</b>	<b>90.42</b>	<b>8.13</b>
		SLEB	46.59	<b>72.43</b>	72.27	44.00	79.00	64.88	55.88	<b>62.15</b>	<b>90.42</b>	<b>8.13</b>
		FuseGPT-MI	43.69	67.26	<b>72.90</b>	44.00	78.89	62.90	63.65	61.90	89.86	8.78
	21.4%	MKA	34.64	47.39	51.55	35.40	64.15	58.33	48.95	<b>48.63</b>	<b>70.82</b>	56126.31
		CoMe	<b>47.87</b>	71.80	72.12	<b>44.20</b>	<b>79.22</b>	62.12	54.85	61.74	90.00	8.17
		Mag+	29.35	51.52	50.35	33.40	67.90	51.85	28.19	44.65	64.69	16.90
		Taylor+	33.02	45.12	45.30	32.00	62.73	55.88	<b>48.73</b>	46.11	67.02	102.48
		ShortGPT	34.98	62.50	<b>60.41</b>	37.40	<b>73.94</b>	54.38	27.87	50.21	72.84	<b>17.98</b>
		SLEB	34.98	62.50	<b>60.41</b>	37.40	<b>73.94</b>	54.38	27.87	50.21	72.84	<b>17.98</b>
		FuseGPT-MI	33.87	58.00	60.79	39.40	73.45	<b>55.96</b>	26.48	49.71	72.33	17.16
	30.6%	MKA	27.82	25.29	27.34	28.20	50.82	52.01	27.85	34.19	50.58	2095234.00
		CoMe	<b>38.82</b>	<b>65.57</b>	60.38	<b>40.00</b>	73.61	55.17	33.10	<b>52.38</b>	<b>76.36</b>	2573306.50
		Mag+	25.51	47.52	39.40	31.20	62.57	48.54	26.01	40.11	58.21	36.02
		Taylor+	25.85	35.90	31.57	28.60	58.76	51.30	25.99	36.85	53.81	422.97
		ShortGPT	30.55	52.53	47.96	32.00	66.76	<b>53.67</b>	25.32	44.11	63.96	32.87
		SLEB	27.30	52.57	48.26	32.00	68.44	51.78	26.66	43.86	63.30	<b>26.33</b>
Qwen-3-4b	10.0%	FuseGPT-MI	26.02	43.81	43.21	28.80	66.10	53.43	25.94	41.04	59.21	51.92
		MKA	24.91	25.25	26.29	29.20	50.71	49.64	24.21	32.89	48.69	14455309.00
		CoMe	<b>33.87</b>	<b>53.54</b>	<b>49.53</b>	<b>34.00</b>	<b>68.72</b>	50.91	<b>26.78</b>	<b>45.34</b>	<b>66.05</b>	17641898.00
		Dense	51.54	76.43	73.70	41.20	77.80	71.03	73.01	66.39	100.00	31.09
		Mag+	43.09	68.01	64.31	38.80	<b>75.79</b>	59.59	53.40	57.57	86.92	13.81
		Taylor+	43.17	63.17	65.15	34.40	71.65	<b>68.11</b>	70.55	59.46	88.99	16.03
	20.1%	ShortGPT	42.58	59.60	64.92	33.80	71.11	66.93	70.20	58.45	87.50	9.43
		SLEB	43.43	67.09	63.37	39.00	75.30	60.38	48.83	56.77	85.91	15.83
		FuseGPT-MI	<b>48.63</b>	<b>73.57</b>	65.54	<b>39.60</b>	<b>75.79</b>	62.35	55.63	<b>60.16</b>	<b>91.01</b>	9.69
		FuseGPT-MI+F	48.38	73.53	<b>65.59</b>	38.80	75.52	61.96	55.55	59.90	90.52	16.03
		MKA	44.11	63.13	53.44	36.20	67.19	59.67	<b>71.19</b>	56.42	85.20	9.44
		CoMe	43.17	69.87	64.86	35.00	74.16	64.25	61.34	<b>58.95</b>	88.28	525.95
	30.1%	Mag+	35.49	61.70	56.05	37.00	<b>72.42</b>	54.54	24.98	48.88	74.22	10.56
		Taylor+	30.55	42.00	47.63	29.40	63.82	59.27	23.69	42.34	64.02	24.16
		ShortGPT	36.35	45.12	52.10	31.00	65.13	59.75	38.11	46.79	70.79	16.60
		SLEB	<b>38.65</b>	<b>64.48</b>	56.04	<b>37.80</b>	70.84	57.85	30.82	50.93	77.41	81.53
		FuseGPT-MI	36.69	59.85	<b>56.41</b>	<b>37.80</b>	72.31	54.38	31.70	49.88	75.81	130.50
		FuseGPT-MI+F	36.77	59.89	56.40	37.60	72.31	54.85	31.65	49.92	75.86	21.69
Qwen-3-4b	10.0%	MKA	37.80	52.95	46.90	33.40	65.23	<b>61.25</b>	<b>71.43</b>	<b>52.71</b>	<b>79.32</b>	13.27
		CoMe	33.19	56.02	55.29	30.40	68.55	59.67	55.10	51.17	76.30	813.38
		Mag+	27.56	45.45	42.60	<b>33.60</b>	65.18	50.99	25.23	41.52	63.20	30.03
		Taylor+	28.58	31.23	32.73	30.60	54.90	50.20	23.24	35.93	55.44	20.75
		ShortGPT	<b>32.17</b>	39.81	45.73	31.40	62.46	53.28	27.75	41.80	63.72	56.14
		SLEB	30.97	53.32	<b>46.42</b>	31.00	65.67	53.83	27.16	<b>44.05</b>	<b>66.50</b>	3861.42
	20.1%	FuseGPT-MI	30.63	55.47	45.90	30.00	65.61	52.80	27.33	43.96	66.17	417.05
		FuseGPT-MI+F	30.29	<b>55.64</b>	45.99	30.40	<b>65.83</b>	52.25	27.38	43.97	66.20	39.60
		MKA	32.08	38.47	40.95	30.60	61.53	<b>60.14</b>	23.43	41.03	62.61	28.11
		CoMe	28.67	47.10	43.95	29.60	63.00	51.46	<b>32.67</b>	42.35	63.84	33.53
		CoMe-P	27.56	47.73	43.41	32.00	63.98	54.06	29.35	42.58	64.43	33.29
												3208.73

Table 11: The detail experiment result of Fig. 7. Effect of  $p$  in heuristic merge ratio.

$p$	ARC-c	ARC-e	HellaS	Benchmark $\uparrow$			WinoG	MMLU (5)	Avg $\uparrow$	RP $\uparrow$	PPL $\downarrow$	
				OBQA	PIQA						C4	Wiki-2
1	35.24	54.46	56.56	35.40	68.88	<b>61.17</b>	25.50	48.17	76.47	<b>19.93</b>	<b>16.53</b>	
2	<b>35.75</b>	<b>54.84</b>	56.27	35.40	68.93	60.77	25.30	<b>48.18</b>	<b>76.51</b>	20.70	16.57	
4	34.98	54.76	<b>57.86</b>	36.40	67.79	59.75	24.22	47.97	76.12	19.87	17.40	
8	34.22	54.17	56.41	<b>36.60</b>	68.12	58.48	26.47	47.78	76.07	20.94	19.19	
16	32.08	53.62	56.07	33.60	68.34	60.62	26.81	47.31	74.86	20.30	17.82	
32	30.89	52.78	55.87	33.40	68.77	59.43	<b>26.93</b>	46.87	74.10	21.43	18.64	
64	32.34	53.28	56.31	34.60	67.95	59.91	26.91	47.33	75.06	21.69	18.01	
128	32.25	52.48	56.51	33.20	<b>69.31</b>	57.38	27.47	46.94	74.36	23.29	18.64	
256	32.00	50.80	55.99	34.80	67.74	60.22	27.16	46.96	74.58	23.97	19.72	
512	31.66	49.33	55.79	33.20	67.79	59.04	26.82	46.23	73.30	25.03	20.97	
inf	31.31	49.07	55.69	32.80	68.01	60.30	26.63	46.26	73.24	25.50	21.21	

Table 12: The detail experiment result of Fig. 10. Effect of calibration data scale.

Num	ARC-c	ARC-e	HellaS	Benchmark $\uparrow$		WinoG	MMLU (5)	Avg $\uparrow$	RP $\uparrow$	PPL $\downarrow$	
				OBQA	PIQA					C4	Wiki-2
2	33.87	52.40	56.42	36.60	66.97	59.19	25.27	47.25	75.19	29.35	43.70
4	<b>35.58</b>	54.25	56.45	<b>37.00</b>	68.23	59.59	27.05	<b>48.31</b>	<b>77.07</b>	23.45	31.42
8	33.28	53.16	54.85	<b>37.00</b>	67.63	59.98	26.95	47.55	75.79	21.08	21.01
16	35.41	55.93	<b>56.83</b>	36.20	66.97	60.14	26.30	48.25	76.80	20.26	16.99
32	33.36	53.49	55.64	36.00	66.54	61.33	<b>27.38</b>	47.68	75.92	22.67	17.26
64	33.19	53.03	56.58	35.80	67.46	60.46	26.87	47.63	75.72	20.61	<b>16.11</b>
128	34.56	53.96	56.37	36.80	68.44	<b>61.56</b>	25.22	48.13	76.49	20.88	17.03
<b>256</b>	35.24	<b>54.46</b>	56.56	35.40	<b>68.88</b>	61.17	25.50	48.17	76.47	19.93	16.53
512	33.79	54.04	56.79	36.60	67.52	60.14	27.23	48.02	76.45	<b>19.60</b>	16.18

Table 13: The detail experiment result of Fig. 8. Impact of merge step granularity.

$m$	ARC-c	ARC-e	HellaS	Benchmark $\uparrow$		WinoG	MMLU (5)	Avg $\uparrow$	RP $\uparrow$	PPL $\downarrow$	
				OBQA	PIQA					C4	Wiki-2
<b>2</b>	<b>35.24</b>	<b>54.46</b>	<b>56.56</b>	<b>35.40</b>	<b>68.88</b>	61.17	25.50	48.17	76.47	<b>19.93</b>	<b>16.53</b>
3	34.30	48.48	54.74	33.60	65.07	<b>63.69</b>	<b>37.69</b>	<b>48.22</b>	<b>77.76</b>	29.56	45.18
4	32.59	48.19	46.94	34.80	65.29	63.22	24.78	45.12	72.00	53.32	109.74
5	32.34	42.21	38.03	34.00	59.30	57.70	28.12	41.67	67.66	91.96	668.17
6	30.63	43.10	40.36	35.00	60.39	58.64	22.57	41.53	66.72	128.37	551.57
7	32.00	41.25	37.14	33.40	58.22	59.51	29.73	41.61	67.70	97.65	268.47
8	33.11	47.22	40.69	<b>35.40</b>	63.49	58.41	28.76	43.87	70.92	96.44	276.09
9	29.69	43.39	41.67	33.80	63.60	57.46	25.56	42.17	67.62	65.33	193.89
10	30.29	43.18	41.13	33.60	62.89	57.06	28.38	42.36	68.27	78.85	303.17
11	30.46	41.58	40.24	34.20	61.37	59.91	24.71	41.78	67.20	104.82	470.91

Table 14: The detail experiment result of Fig. 9. Impact of calibration datasets.

Dataset	ARC-c	ARC-e	HellaS	Benchmark $\uparrow$		WinoG	MMLU (5)	Avg $\uparrow$	RP $\uparrow$	PPL $\downarrow$	
				OBQA	PIQA					C4	Wiki-2
wiki2	<b>35.24</b>	<b>54.46</b>	56.56	35.40	68.88	61.17	25.50	48.17	76.47	19.93	<b>16.53</b>
C4	34.30	52.95	<b>57.25</b>	<b>35.60</b>	<b>69.80</b>	62.04	26.39	<b>48.33</b>	<b>76.71</b>	<b>19.32</b>	21.05
PG19 (2)	34.39	50.21	53.74	34.00	67.25	60.69	27.90	46.88	74.77	21.28	21.15
MMLU	33.70	50.80	54.13	33.20	66.00	<b>63.77</b>	27.20	46.97	74.67	28.12	35.79
Aplaca	34.39	51.77	53.54	35.00	67.25	<b>63.77</b>	<b>29.87</b>	47.94	76.61	25.65	30.57

---

**Algorithm 1** Progressive Concatenation-based Layer Merging Strategy (CoMe)

---

**Input:** calibration dataset  $\mathcal{D}$ , original model  $M$ , the number of layers skipped in SBI  $m$ , skewness exponent  $p$ , target layer number  $L$ , minimum retention ratio  $\rho \in (0, 1)$

**Output:** Pruned model  $M'$ , layer mapping  $\mathcal{P} = [\{a_1, b_1\}, \dots, \{a_N, b_N\}]$

```
1: Initialize  $\mathcal{P} \leftarrow \emptyset, M' \leftarrow M$ 
2: while NUMLAYERS( $M'$ )  $> L$  do
3:    $m' \leftarrow \min(m, \text{NUMLAYERS}(M') - L)$ 
4:    $\{S^{(l)}\} \leftarrow \text{COMPUTECHANNELSENSITIVITY}(M', \mathcal{D})$ 
5:    $\{BI_l\} \leftarrow \text{COMPUTEBI SCORES}(\{\mathbf{H}^{(l)}\})$ 
6:    $\{SBI_{l:l+m'}\} \leftarrow \text{COMPUTESBI SCORES}(\{\mathbf{H}^{(l)}\}, m')$ 
7:    $(l^*, l^* + m') \leftarrow \arg \min SBI_{l:l+m}$ 
8:    $\{r_t\} \leftarrow \text{ADJUSTRETENTIONRATIOS}(\{BI_t\}_{t=l^*}^{l^*+m'}, p, \rho)$ 
9:    $W^{(\text{merge})} \leftarrow \text{CONCATENATIONBASEDLAYERMERGE}(\{W^{(t)}, S^{(t)}, r_t\}_{t=l^*}^{l^*+m'})$ 
10:  Replace layers  $[l^*, \dots, l^* + m']$  with  $W^{(\text{merge})}$  in  $M'$ 
11:   $\mathcal{P} \leftarrow \mathcal{P} \cup [\{l^* + m', \text{new layer index in } M'\}]$ 
12: end while
13: return  $M', \mathcal{P}$ 
14:
15: function COMPUTECHANNELSENSITIVITY( $M', \mathcal{D}$ )
16:   for each layer  $l \in M'$  do
17:      $\mathbf{H}^{(l)} \leftarrow \text{FORWARDPASS}(M', \mathcal{D}, l)$ 
18:      $S^{(l)} \leftarrow \{\mathbb{E}_{\mathcal{D}}[|x_i| \sum_k |w_{i,k}|]\}_{i=1}^u$  ▷ Eq. (1)
19:   end for
20:   return  $\{S^{(l)}\}_{l=1}^L$ 
21: end function
22: function COMPUTEBI SCORES( $\{\mathbf{H}^{(l)}\}$ )
23:   for  $l = 1$  to NUMLAYERS( $M'$ ) do
24:      $BI_l \leftarrow 1 - \mathbb{E}_{\mathcal{D}} \left[ \frac{\mathbf{H}^{(l-1)\top} \mathbf{H}^{(l)}}{\|\mathbf{H}^{(l-1)}\|_2 \|\mathbf{H}^{(l)}\|_2} \right]$  ▷ Eq. (2)
25:   end for
26:   return  $\{BI_l\}$ 
27: end function
28: function COMPUTESBI SCORES( $\{\mathbf{H}^{(l)}\}, m'$ )
29:   for  $l = 1$  to NUMLAYERS( $M'$ )  $- m'$  do
30:      $SBI_{l:l+m'} \leftarrow 1 - \mathbb{E}_{\mathcal{D}} \left[ \frac{\mathbf{H}^{(l-1)\top} \mathbf{H}^{(l+m')}}{\|\mathbf{H}^{(l-1)}\|_2 \|\mathbf{H}^{(l+m')}\|_2} \right]$  ▷ Eq. (3)
31:   end for
32:   return  $\{SBI_{l:l+m'}\}$ 
33: end function
34: function ADJUSTRETENTIONRATIOS( $\{BI_t\}_{t=l^*}^{l^*+m'}, p, \rho$ )
35:    $r_t \leftarrow BI_t^p / \sum_i BI_i^p$  for  $t \in [l^*, l^* + m']$  ▷ Eq. (5)
36:   if  $\max r_t < \rho$  then
37:      $r_{\arg \max BI_t} \leftarrow \rho$ 
38:      $t^* \leftarrow \arg \max BI_t$ 
39:      $\sum' \leftarrow \sum_{t \neq t^*} r_t$ 
40:      $r_t \leftarrow (1 - \rho)r_t / \sum'$  for  $t \neq t^*$ 
41:   end if
42:   return  $\{r_t\}$  normalized to  $\sum r_t = 1$ 
43: end function
44: function CONCATENATIONBASEDLAYERMERGE( $\{W^{(t)}, S^{(t)}, r_t\}_{t=l^*}^{l^*+m'}$ )
45:   for each layer  $t \in [l^*, l^* + m']$  do
46:      $k_t \leftarrow r_t \times |S^{(t)}|$ 
47:      $\mathcal{T}_t \leftarrow \text{Top-}k_t \text{ indices sorted by } S^{(t)}$ 
48:   end for
49:    $W^{(\text{merge})} \leftarrow \bigoplus_{t=l^*}^{l^*+m'} W^{(t)}[:, \mathcal{T}_t]$  ▷ Eq. (4)
50:   return  $W^{(\text{merge})}$ 
51: end function
```

---

---

**Algorithm 2** Progressive Posterior-based CoMe (CoMe-P)

---

**Input:** Calibration dataset  $\mathcal{D}$ , original model  $M$ , target layers number  $L$ , search granularity  $n$

**Output:** Pruned model  $M'$ , layer mapping  $\mathcal{P} = [\{a_1, b_1\}, \dots, \{a_N, b_N\}]$

```
1: Initialize  $\mathcal{P} \leftarrow \emptyset$ ,  $M' \leftarrow M$ ,
2: Generate parameter preservation ratio candidate set  $\Gamma = \{\frac{i}{n} \mid i = 0, 1, 2, \dots, n\}$ 
3: while NUMLAYERS( $M'$ )  $> L$  do
4:    $\{S^{(l)}\} \leftarrow \text{COMPUTECHANNELSENSITIVITY}(M', \mathcal{D})$ 
5:    $\{SBI_{l:l+1}\} \leftarrow \text{COMPUTESBISCORES}(\{H^{(l)}\}, 1)$ 
6:    $(l^*, l^* + 1) \leftarrow \arg \min SBI_{l:l+1}$ 
7:   for each  $r$  in  $\Gamma$  do
8:      $W^{(\text{merge})} \leftarrow \text{CONCATENATIONBASEDLAYERMERGE}(\{W^{(t)}, S^{(t)}, r_t\}_{t=l^*}^{l^*+1})$ 
9:      $M'' \leftarrow \text{Replace layers } [l^*, l^* + 1] \text{ with } W^{(\text{merge})} \text{ in } M'$ 
10:     $ppl \leftarrow PPL(M'')$ 
11:  end for
12:   $M' \leftarrow \text{The } M'' \text{ has the lower } ppl$ 
13:   $\mathcal{P} \leftarrow \mathcal{P} \cup [\{l^* + 1, \text{new layer index in } M'\}]$ 
14: end while
15: return  $M', \mathcal{P}$ 
```

---

---

**Algorithm 3** CoMe Single-Process Post-training (CoMe-sp)

---

**Input:** training data  $\mathcal{D}_{\text{train}}$ , layer mapping  $\mathcal{P} = [\{a_1, b_1\}, \dots, \{a_N, b_N\}]$ , teacher model  $M$ , student model  $M'$ , learning rate  $\eta$ , optimizer  $\Omega$ , batch size  $B$

**Output:** Optimized student model  $M'$

```
1: Initialize  $\Omega \leftarrow \text{ADAM}(\{\theta_{b_i} \mid \{a_i, b_i\} \in \mathcal{P}\}, \eta)$  ▷ Optimize merged layers only
2: for epoch = 1 to  $E_{\text{global}}$  do
3:   for  $\mathcal{B} \leftarrow \text{BATCHLOADER}(\mathcal{D}_{\text{train}}, B)$  do
4:      $\{H^{(t, a_i)}\}_{i=1}^N \leftarrow \text{GETACTIVATIONS}(M, x, \{a_i \mid \{a_i, b_i\} \in \mathcal{P}\})$ 
5:      $\{H^{(s, b_i)}\}_{i=1}^N \leftarrow \text{GETACTIVATIONS}(M', x, \{b_i \mid \{a_i, b_i\} \in \mathcal{P}\})$ 
6:     return  $\{H^{(t, a_i)}\}_{i=1}^N, \{H^{(s, b_i)}\}_{i=1}^N$ 
7:    $\mathcal{L}_{\text{total}} \leftarrow 0$ 
8:   for  $i = 1$  to  $N$  do
9:      $\mathcal{L}_{\text{KL}}^{(i)} \leftarrow \frac{1}{N} D_{\text{KL}}(\sigma(H^{(t, a_i)}) \parallel \sigma(H^{(s, b_i)}))$  ▷ Eq. (6)
10:     $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{total}} + \mathcal{L}_{\text{KL}}^{(i)}$  ▷ Eq. (8)
11:  end for
12:   $\Omega.\text{zero\_grad}()$ 
13:   $\mathcal{L}_{\text{total}}.\text{BACKWARD}()$ 
14:   $\Omega.\text{step}()$ 
15: end for
16: end for
```

---

---

**Algorithm 4** CoME Multi-Process Post-training (CoMe-mp)

---

**Input:** training data  $\mathcal{D}_{\text{train}}$ , layer mapping  $\mathcal{P} = [\{a_1, b_1\}, \dots, \{a_N, b_N\}]$ , teacher model  $M$ , student model  $M'$ , learning rate  $\{\eta_1, \dots, \eta_{|\mathcal{P}|}\}$ , optimizer  $\Omega$ , batch size  $B$

```
1: for  $k = 1$  to  $|\mathcal{P}|$  do ▷ Layerwise progression
2:    $\{a_k, b_k\} \leftarrow \mathcal{P}[k]$ 
3:    $\Omega_k \leftarrow \text{ADAM}(\theta_{b_k}, \eta_k = \eta_k)$ 
4:   for epoch = 1 to  $E_{\text{local}}$  do
5:     for  $\mathcal{B} \leftarrow \text{BATCHLOADER}(\mathcal{D}_{\text{train}}, B)$  do
6:        $\mathbf{H}^{(t, a_k)} \leftarrow \text{GETSINGLEACTIVATION}(M, x, a_k)$ 
7:        $\mathbf{H}^{(s, b_k)} \leftarrow \text{FORWARDTOLAYER}(M', x, b_k)$ 
8:        $\mathcal{L}_{\text{KL}} \leftarrow D_{\text{KL}}(\sigma(\mathbf{H}^{(t, a_k)}) \parallel \sigma(\mathbf{H}^{(s, b_k)}))$  ▷ Eq. (6)
9:        $\Omega_k.\text{zero\_grad}()$ 
10:       $\mathcal{L}_{\text{KL}}.\text{BACKWARD}()$ 
11:       $\Omega_k.\text{step}()$ 
12:     end for
13:   end for
14: end for
15: return  $M'$ 
```

---