

Small Models, Fewer Heads, More Forgetting: Sparse Attention Fine-Tuning for SLM Continual Learning

Hao Xu¹, Shezheng Song^{2*}, Long Peng^{1*}, Xiaodong Liu¹,
Xiaopeng Li¹, Ma Jun¹, Shasha Li¹, Jie Yu¹

¹College of Computer Science and Technology, National University of Defense Technology

²Hefei University of Technology,

xuhao19@nudt.edu.cn, betterzsong@gmail.com,

{penglong, liuxiaodong, xiaopengli, majun, shashali, yj}@nudt.edu.cn

Abstract

Small Language Models (SLMs) have been widely deployed in edge computing and mobile devices. Generally, these models need to be continuously updated to adapt to dynamic environments and evolving user needs. Therefore, the performance of SLMs in continual learning has become a critical research focus. Compared to LLMs with rich redundant representation spaces, SLMs are more likely to overwrite previous representations during new task learning due to their fewer attention heads and layers. To address this issue, we propose LightLoRA, a sparse fine-tuning method specifically designed for SLMs. This method constructs a conflict-aware importance mechanism by combining raw importance from SVD decomposition with task conflict degrees based on gradient direction consistency. By selecting and updating heads that provide **high contributions** and **low conflicts**, LightLoRA successfully maintains sufficient space for knowledge preservation. Experimental results show that the proposed method reaches state-of-the-art performance across multiple benchmarks without adding extra parameters. Code is released on <https://github.com/season1blue/lightlora>.

1 Introduction

The high computational cost of Large Language Models (LLMs) (Brown et al., 2020; Touvron et al., 2023; Achiam et al., 2023) has driven growing interest in **Small Language Models (SLMs)** (Lu et al., 2024; Belcak et al., 2025), typically defined as models with fewer than 1B parameters that can be deployed efficiently on personal devices (Dubiel et al., 2024). In practical applications, SLMs still need to be continually adapted to evolving tasks and user needs through fine-tuning, naturally giving rise to a continual learning setting where new knowledge must be acquired without losing previously learned capabilities.

*Corresponding authors.

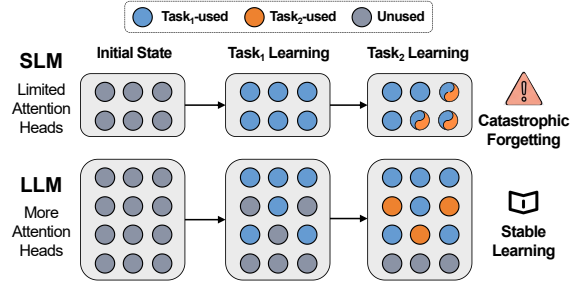


Figure 1: SLMs are more prone to catastrophic forgetting because they have fewer attention heads than LLMs, limiting available representational space for new tasks.

A central challenge in this process is **catastrophic forgetting**. Based on our empirical study, forgetting caused by fine-tuning is more pronounced in SLMs than in larger models, which significantly constrains their practical use in dynamic environments. One important reason is the limited representational capacity of SLMs. As shown in Figure 1, Qwen1.5-0.5B has only half as many attention heads per layer as Qwen1.5-7B, leaving less representational space to accommodate new knowledge without overwriting old knowledge. Despite this challenge, existing studies on SLMs have mainly focused on model compression, structural efficiency, and distillation training (Zhai, 2024; Wang et al., 2025a), while fine-tuning-induced catastrophic forgetting in SLMs remains largely underexplored.

Existing methods for mitigating catastrophic forgetting mainly include regularization-based (Kirkpatrick et al., 2017; Wang et al., 2023, 2022; He et al., 2023), replay-based (Rebuffi et al., 2017; Wang et al., 2024a), parameter-expansion approaches (Wu et al., 2024; Razdaibiedina et al., 2023), and parameter-isolation methods that maintain separate adapters for different tasks (Wang et al., 2024b; Huang et al., 2023). These methods, however, are not ideal for SLMs. Replay-based methods require storing historical data, increasing memory usage. Parameter-expansion meth-

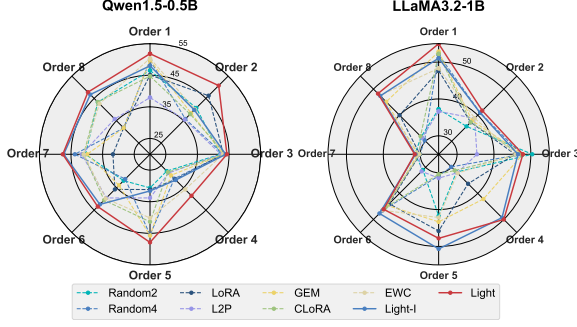


Figure 2: Radar-chart comparison between our method (denoted as *Light*) and baselines across task orders.

ods add new trainable components as tasks accumulate. Parameter-isolation methods maintain separate adapters for different tasks, which increases storage cost and requires selecting the correct adapter at inference time.

We observe that activating only a few key attention heads during single-task fine-tuning achieves performance comparable to fine-tuning all heads. Therefore, we can fine-tune only a subset of attention heads for each task during continual learning to reserve capacity for subsequent tasks. There are several challenges: (1) How to determine which attention heads are **important** for the current task. (2) How to select attention heads when **conflicts** arise between historical and current tasks.

Therefore, we propose **LightLoRA**, a sparse attention fine-tuning method for SLM continual learning. Rather than updating all attention heads, LightLoRA aims to identify heads that are most beneficial for the current task while causing minimal interference to previous tasks. We first define **raw importance** from task-specific parameter variation: by applying SVD to the incremental LoRA update of each head, we use its singular values to capture how strongly that head participates in learning the current task. Since a highly active head may still conflict with historical knowledge, we further measure **inter-task conflict** through the directional consistency between current and historical gradients, where orthogonal directions imply low interference. We then combine raw importance and conflict into a **conflict-aware importance** score for selective head activation. In this way, LightLoRA updates only a small subset of heads that are both useful and low-conflict, enabling effective new-task learning while mitigating catastrophic forgetting. As shown in Figure 2, LightLoRA (marked by the red curve) consistently outperforms previous methods.

Our contributions are as follows:

- From the perspective of catastrophic forgetting, we provide the first systematic analysis of the vulnerability of SLMs in continual learning caused by insufficient attention heads.
- We propose a gradient conflict-aware sparse attention fine-tuning mechanism to reduce the forgetting of historical tasks while effectively learning the current task.
- Experimental results show that our method achieves state-of-the-art (SOTA) performance on benchmarks consisting of two models and eight datasets across multiple training orders. It serves as a lightweight strategy since it requires no extra parameter modules.

2 Motivation

2.1 SLM is Prone to Forgetting

Intuitively, models with fewer parameters are more likely to suffer from catastrophic forgetting due to their limited capacity. To verify this, we compared two representative models: Qwen1.5-0.5B and Qwen1.5-7B. We first fine-tuned each model on 10 individual tasks separately (FT) to establish a performance baseline. Then, we sequentially fine-tuned both models on the same 10 tasks (Continual FT). Based on LoRA (Hu et al., 2022) research, which shows that tuning only the query (W_q) and value (W_v) attention matrices can achieve performance close to full fine-tuning, we update only the attention weights in our experiments for both simplicity and parameter efficiency. Details are provided in Appendix A.3.

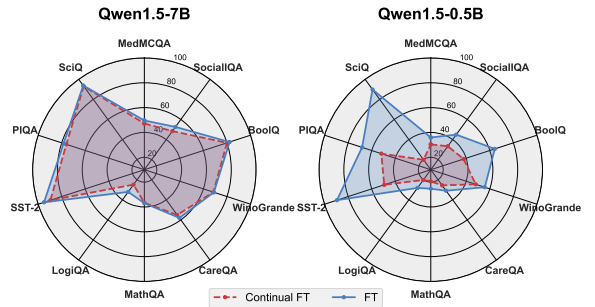


Figure 3: Task accuracy of Qwen1.5-7B and Qwen1.5-0.5B after fine-tuning on 10 tasks.

As shown in Figure 3, the 0.5B model shows much more severe forgetting than the 7B model. Compared to the 0.5B model, the 7B model has more attention heads and a larger representational space. When learning new tasks, larger models

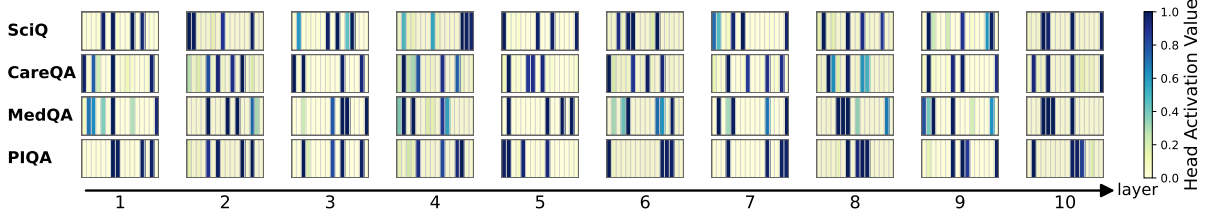


Figure 4: Layer-wise activation frequency of attention heads across four distinct tasks during fine-tuning. For brevity, only the first 10 Transformer layers are displayed.

can use inactive or less important heads, reducing interference with old knowledge. In contrast, SLMs have fewer heads and must often modify those already storing information from previous tasks. Prior work (Michel et al., 2019) suggests that removing several attention heads from a trained Transformer model does not significantly hurt performance. We further explore whether this property holds during the fine-tuning stage.

2.2 Only a Few Heads Matter

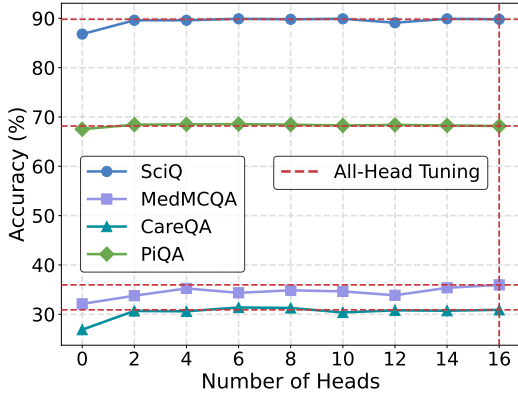


Figure 5: Impact of the number of activated attention heads on accuracy across four tasks.

Using the Qwen1.5-0.5B, we conducted fine-tuning experiments on four datasets for one epoch each. During fine-tuning, we defined a measure of attention raw importance (see Section 3.1). We only applied LoRA updates to the most important heads for the current task. We varied the number of active heads from 0 to 16 (where 16 represents standard LoRA). The results in Figure 5 show that **fine-tuning only a few critical heads performs almost as well as fine-tuning all heads**. In some cases, selective fine-tuning even works better. This suggests that not all attention heads are equally important during fine-tuning.

Additionally, we recorded the activation status of attention heads in each Transformer layer across various tasks during the fine-tuning process. Figure 4 shows that different tasks rely on different

attention heads. Consequently, we can update only the attention heads that are important to the current task during fine-tuning to reserve learning capacity for subsequent tasks.

3 Methodology

We follow the standard continual learning setting, where a model learns a sequence of tasks $\mathcal{T} = \{T_1, T_2, \dots, T_N\}$. During the training of task T_t , rather than utilizing all attention heads, we activate only a subset for fine-tuning. This involves three steps: **(1) Raw importance calculation**: assessing the contribution of each head to the current task T_t . **(2) Inter-task gradient conflict assessment**: measuring the consistency between the gradient direction of the current task and that of historical tasks. **(3) Conflict-aware head activation**: combining raw importance and inter-task gradient conflict to compute a conflict-aware importance for each head and activating the k highest-scoring heads in each Transformer layer. The overall framework is shown in Figure 6.

3.1 Raw Importance Calculation

To quantify the contribution of each attention head h to task T_t , we define the **raw importance** $I_t(h)$. This measures the change in LoRA parameters corresponding to the head h since the completion of the previous task T_{t-1} .

Specifically, LoRA approximates weight updates using low-rank matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times d}$, where the update is represented as $\Delta W = BA$. During the training of task T_t , let $W_t = B_t A_t$ be the update matrix at the current time step, and $W_{t-1} = B_{t-1} A_{t-1}$ be the matrix at the end of task T_{t-1} . The incremental update is defined as $\Delta W_t = W_t - W_{t-1}$, which can be viewed as the representation of new knowledge learned specifically for task T_t .

For the h -th head in the l -th layer, let $\Delta W_t^{(l,h)}$ be its corresponding parameter update sub-matrix. Its

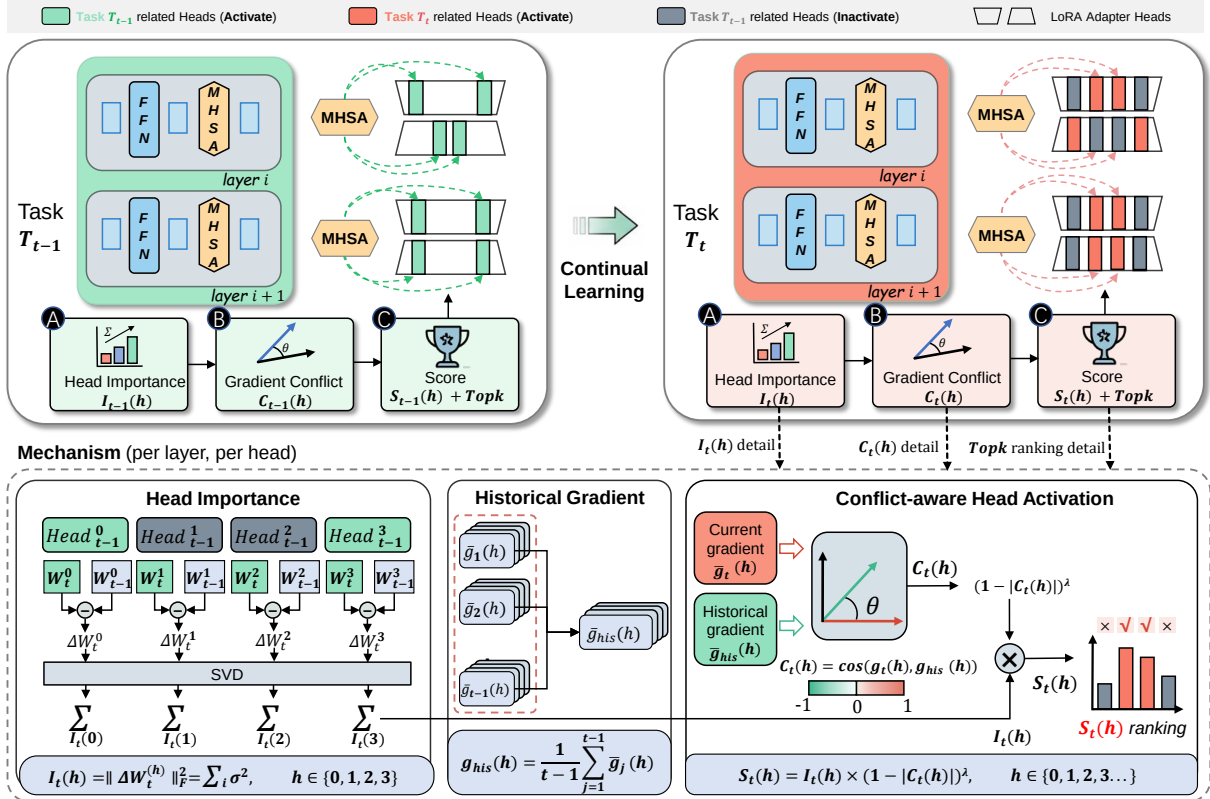


Figure 6: Selective head activation and preservation in task-aware continual fine-tuning. During task T_{t-1} fine-tuning (left), only a subset of heads (green) is activated. As training continues on task T_t (right), a new set of heads (red) becomes active, while previously used heads from task T_{t-1} (gray) are preserved and inactivated.

importance is defined by squared Frobenius norm:

$$I_t(l, h) = \|\Delta W_t^{(l, h)}\|_F^2 = \sum_i \sigma_i^2, \quad (1)$$

where σ_i represents the singular values of $\Delta W_t^{(l, h)}$. This metric reflects the intensity of parameter changes when the head adapts to the new task. Note that for the first task T_1 , we use W_1 as the importance as there is no prior task.

3.2 Inter-task Gradient Conflict Assessment

To mitigate catastrophic forgetting, it is essential to ensure that parameter updates for the new task do not conflict with the optimization directions of previous tasks. We achieve this by comparing gradient vectors across tasks.

First, for each head h , we define its gradient vector $\bar{g}(h)$ by flattening and concatenating the gradients of its LoRA matrices, ∇A and ∇B . This vector $\bar{g}(h)$ therefore characterizes the gradient direction in the LoRA parameter space associated with head h , reflecting how the head's low-rank adaptation parameters are being updated.

During training for task T_t , we compute the average gradient vector $\bar{g}_t(h)$ within a sliding window

of recent batches to represent the current task's direction. Meanwhile, after each historical task $T_j (j < t)$ is completed, we store its average gradient vector $\bar{g}_j(h)$. The overall historical gradient direction is:

$$\bar{g}_{his}(h) = \frac{1}{t-1} \sum_{j=1}^{t-1} \bar{g}_j(h). \quad (2)$$

We measure the **inter-task gradient conflict** $C_t(h)$ between the current and historical directions:

$$C_t(h) = \frac{\bar{g}_t(h) \cdot \bar{g}_{his}(h)}{|\bar{g}_t(h)| \cdot |\bar{g}_{his}(h)|}. \quad (3)$$

When $C_t(h)$ approaches 0, it indicates that the current task's gradient direction is nearly orthogonal to that of previous tasks in the LoRA parameter space, implying that the corresponding head learns task-specific features without disturbing previously acquired knowledge.

3.3 Conflict-Aware Head Activation

We combine raw importance $I_t(h)$ and inter-task gradient conflict $C_t(h)$ into a **conflict-aware importance** $S_t(h)$:

$$S_t(h) = I_t(h) \times (1 - |C_t(h)|)^\lambda, \quad (4)$$

where $\lambda > 1$ controls the sensitivity to gradient conflict. The raw importance $I_t(h)$ quantifies the contribution of each head to the current task, while the conflict term $(1 - |C_t(h)|)^\lambda$ (ranges from 0 to 1) adjusts this score based on the relationship between the current and historical gradient.

Following the principle of orthogonality in continual learning (Wang et al., 2023), we prioritize heads where the current gradient is nearly orthogonal to the historical gradient ($C_t(h) \approx 0$). This allows the model to learn new knowledge in a parameter subspace that does not interfere with previous tasks. In each Transformer layer, we rank heads by $S_t(h)$ and select the top- k heads to form a binary mask. During training, only the LoRA parameters corresponding to the selected heads are updated, while the others remain frozen.

4 Experiments and Analysis

4.1 Experiment Setup

4.1.1 Models and Datasets

We employ *Qwen1.5-0.5B* (Bai et al., 2023) and *LLaMA3.2-1B* (Grattafiori et al., 2024) as representative SLMs. Our experiments are conducted on two benchmarks: Standard CL Benchmark and Large Number of Tasks Benchmark. Our experimental datasets cover a diverse range of tasks, including text classification, medical QA, mathematical and logical reasoning, as well as commonsense and scientific reasoning. Details on datasets and task order can be found in Appendix A.4. All experiments were conducted on NVIDIA A6000 GPUs. Following Hu et al. (2022) and Wang et al. (2023), LoRA is applied to the attention weights of queries (W_q) and values (W_v). More implementation details can be found in the Appendix A.3.

4.1.2 Baselines and Metrics

The proposed method is compared with several representative continual learning baselines, whose detailed descriptions are provided in Appendix A.2. Results of multi-task learning (MTL) are also included as an upper-bound reference. As lightweight baselines, two or four attention heads are randomly selected for updates during training, denoted as *Random(2)* and *Random(4)*. These settings are chosen because LightLoRA updates only a small subset of heads (e.g., $k=4$) while freezing the rest. In addition, *Light-I* is a simplified variant of our method that selects heads solely based on raw importance scores.

We report the average accuracy (**AA**↑) and backward transfer (**BWT**↑) (Lopez-Paz and Ranzato, 2017) after training on the last task. AA is the mean test accuracy over all tasks evaluated after finishing the final task, and BWT measures how performance on previous tasks changes after learning new tasks (higher is better, indicating less forgetting). Let $a_{i,j}$ denote the accuracy of the model on the i -th task after completing training on the j -th task. The equation are as follows:

$$AA = \frac{1}{T} \sum_{i=1}^T a_{i,T} \quad (5)$$

$$BWT = \frac{1}{T-1} \sum_{i=1}^{T-1} (a_{i,T} - a_{i,i}) \quad (6)$$

4.2 Main Results

We conduct experiments on the Qwen and LLaMA models, and report the results in Tables 1 and 2.

Standard CL Benchmark The results show that LightLoRA consistently outperforms all other continual learning baselines across both models and all task orders. We attribute this success to our strategy of only updating the most important attention heads. This approach allows the model to learn new knowledge effectively while reserving sufficient parameter space for future tasks, thereby preventing catastrophic forgetting.

Large Number of Tasks Benchmark The performance gap between Light and other baseline methods becomes even more obvious in this setting. In Table 1, for example, traditional SeqLoRA only achieves an average accuracy of 32.47%, while Light reaches 42.79%. To further verify the effectiveness of our selection mechanism, we compared our method with Random(2) and Random(4). As shown in Table 1 and Table 2, these random selection baselines perform significantly worse than LightLoRA. This confirms that simply updating a small subset of attention heads is insufficient.

4.3 Effect of Conflict-Aware Head Activation

Figure 7 shows how LightLoRA selects attention heads during a training batch. We can see that the heads chosen based on the raw importance are different from those selected using the conflict-aware strategy. This discrepancy is expected because our **conflict-aware importance explicitly incorporates information from previous tasks**. Specifically, the score $S_t(h) = I_t(h) \times (1 - |C_t(h)|)^\lambda$ not

<i>Qwen0.5B</i>	Standard CL Benchmark				Large Number of Tasks					
	Order 1	Order 2	Order 3	Average	Order 4	Order 5	Order 6	Order 7	Order 8	Average
MTL	54.72				51.58					
Random(2)	46.50	40.77	42.27	43.18	31.46	30.56	31.71	40.56	47.04	33.57
	<i>-7.51</i>	<i>-5.67</i>	<i>-14.88</i>	<i>-9.35</i>	<i>-10.68</i>	<i>-4.92</i>	<i>-8.71</i>	<i>-8.44</i>	<i>-5.08</i>	<i>-7.57</i>
Random(4)	49.81	35.78	44.17	43.25	30.92	31.05	31.08	43.80	31.84	38.06
	<i>-5.60</i>	<i>-11.56</i>	<i>-12.85</i>	<i>-10.00</i>	<i>-20.79</i>	<i>-16.82</i>	<i>-5.72</i>	<i>-8.59</i>	<i>-5.55</i>	<i>-11.49</i>
SeqLoRA	45.08	46.23	43.23	44.85	31.28	31.16	35.71	31.73	31.82	32.47
	<i>-12.53</i>	<i>-8.04</i>	<i>-15.20</i>	<i>-11.92</i>	<i>-17.31</i>	<i>-4.51</i>	<i>-15.88</i>	<i>-4.43</i>	<i>-6.76</i>	<i>-9.78</i>
L2P	37.95	35.50	42.43	38.63	28.35	31.25	39.55	41.61	35.79	35.19
	<i>-3.03</i>	<i>-18.18</i>	<i>-10.18</i>	<i>-10.46</i>	<i>-17.34</i>	<i>-3.49</i>	<i>-5.75</i>	<i>-4.81</i>	<i>-5.17</i>	<i>-7.31</i>
GEM	49.81	38.39	43.21	43.80	29.41	43.42	42.30	39.88	31.77	36.65
	<i>-4.46</i>	<i>-8.55</i>	<i>-10.15</i>	<i>-7.72</i>	<i>-19.02</i>	<i>-2.09</i>	<i>-14.57</i>	<i>-12.39</i>	<i>-6.48</i>	<i>-10.91</i>
CLoRA	44.72	37.97	43.84	42.17	27.98	42.58	40.54	41.02	43.40	38.03
	<i>-8.07</i>	<i>-7.91</i>	<i>-7.85</i>	<i>-7.94</i>	<i>-22.76</i>	<i>-5.91</i>	<i>-10.92</i>	<i>-6.42</i>	<i>-4.65</i>	<i>-10.13</i>
EWC	50.35	40.11	43.13	44.53	35.74	40.16	39.82	40.92	36.37	39.16
	<i>-1.45</i>	<i>-5.67</i>	<i>-10.04</i>	<i>-5.72</i>	<i>-14.29</i>	<i>-9.54</i>	<i>-8.29</i>	<i>-9.77</i>	<i>-17.21</i>	<i>-11.82</i>
OLoRA	43.55	47.06	42.85	44.49	37.35	43.23	43.07	44.54	44.90	42.05
	<i>-1.40</i>	<i>-5.44</i>	<i>-2.23</i>	<i>-3.02</i>	<i>-8.87</i>	<i>-6.13</i>	<i>-4.49</i>	<i>-7.59</i>	<i>-2.45</i>	<i>-5.91</i>
Light-I	47.99	39.83	43.95	43.92	31.52	31.60	42.44	47.31	46.73	38.22
	<i>-8.90</i>	<i>-6.32</i>	<i>-13.42</i>	<i>-9.55</i>	<i>-9.41</i>	<i>1.31</i>	<i>-10.36</i>	<i>-4.71</i>	<i>-2.23</i>	<i>-5.08</i>
Light	51.82	50.73	44.34	48.96	40.17	47.88	43.40	47.61	47.79	42.79
	<i>-0.08</i>	<i>-3.74</i>	<i>-2.06</i>	<i>-1.96</i>	<i>-5.41</i>	<i>2.81</i>	<i>-0.22</i>	<i>-4.35</i>	<i>-1.52</i>	<i>-1.74</i>

Table 1: Results on two CL benchmarks with **Qwen1.5-0.5B**. Averaged accuracy (AA $\uparrow$) (in black font) after training on the last task and *backward transfer (BWT $\uparrow$)* (in blue italic) is reported.

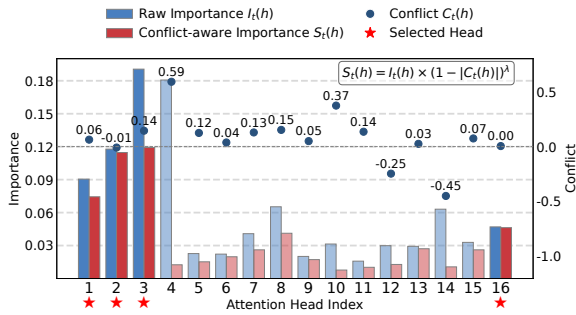


Figure 7: Raw importance, conflict-aware importance, and conflict degree for attention heads selection.

only measures how much a head contributes to the current task ($I_t(h)$), but also penalizes heads whose current update direction is highly aligned with historical optimization directions (large $|C_t(h)|$). As a result, a head that seems important for the current batch may be scored lower if its update is likely to interfere with prior knowledge. Comparative experiments on different designs of $S_t(h)$ are presented in Appendix A.6.

Figure 8 further visualizes the resulting head activation patterns under raw importance and conflict-aware importance when fine-tuning SST-2 in Order 5. Raw-importance selection produces highly concentrated activations, indicating that updates are dominated by a small set of heads (and often a

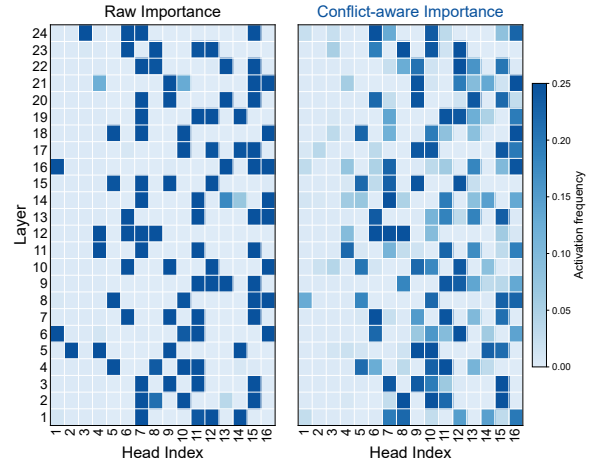


Figure 8: Head activation distribution across layers.

few layers). In contrast, the conflict-aware strategy yields a more distributed activation map, suggesting more **balanced use** of update capacity across heads, which is generally associated with improved robustness and generalization.

4.4 Overhead Analysis

Figure 9 compares the training time of LightLoRA and standard LoRA. LightLoRA adds only a small overhead across both Qwen and LLaMA models due to the calculation of head importance and conflicts. In most cases, the time difference is minimal.

<i>LLaMA1B</i>	Standard CL Benchmark				Large Number of Tasks					
	Order 1	Order 2	Order 3	Average	Order 4	Order 5	Order 6	Order 7	Order 8	Average
MTL	52.88				54.91					
Random(2)	37.25	35.70	45.39	40.28	31.83	41.71	31.03	31.45	31.71	33.54
	<i>-8.52</i>	<i>-16.66</i>	<i>-5.60</i>	<i>-10.26</i>	<i>-16.03</i>	<i>-3.98</i>	<i>-10.63</i>	<i>-5.41</i>	<i>-15.44</i>	<i>-10.30</i>
Random(4)	52.00	40.93	47.46	46.80	29.83	30.87	46.65	28.81	30.37	33.31
	<i>-8.29</i>	<i>-14.17</i>	<i>-3.32</i>	<i>-8.59</i>	<i>-26.15</i>	<i>-5.58</i>	<i>-6.36</i>	<i>-13.80</i>	<i>-9.38</i>	<i>-12.25</i>
SeqLoRA	47.66	37.28	45.73	43.56	36.30	45.81	43.98	30.07	39.98	39.23
	<i>-12.16</i>	<i>-14.35</i>	<i>-12.88</i>	<i>-13.13</i>	<i>-11.52</i>	<i>-6.86</i>	<i>-6.23</i>	<i>-6.06</i>	<i>-7.08</i>	<i>-7.55</i>
L2P	36.46	38.78	35.39	36.87	32.12	31.47	31.54	30.53	31.37	31.41
	<i>-9.10</i>	<i>-13.17</i>	<i>-3.40</i>	<i>-8.56</i>	<i>-17.04</i>	<i>-13.36</i>	<i>-6.38</i>	<i>-6.17</i>	<i>-6.47</i>	<i>-9.88</i>
GEM	52.82	37.87	45.34	45.34	42.17	43.33	44.63	30.67	45.11	41.18
	<i>-8.63</i>	<i>-15.25</i>	<i>-10.37</i>	<i>-11.42</i>	<i>-11.53</i>	<i>-12.23</i>	<i>-10.68</i>	<i>-7.65</i>	<i>-2.78</i>	<i>-8.97</i>
CLoRA	52.26	37.30	47.12	45.56	31.21	30.42	45.55	30.21	31.95	33.87
	<i>-8.54</i>	<i>-17.25</i>	<i>-4.39</i>	<i>-10.06</i>	<i>-17.98</i>	<i>-6.16</i>	<i>-2.48</i>	<i>-7.12</i>	<i>-9.06</i>	<i>-8.56</i>
EWC	48.32	38.82	45.86	44.33	32.26	42.16	46.47	30.23	47.55	39.74
	<i>-11.10</i>	<i>-14.65</i>	<i>-4.78</i>	<i>-10.18</i>	<i>-20.22</i>	<i>-6.08</i>	<i>-7.92</i>	<i>-18.44</i>	<i>-7.43</i>	<i>-12.02</i>
OLoRA	49.50	38.27	47.39	45.05	46.05	46.39	48.16	41.91	46.37	45.78
	<i>-9.34</i>	<i>-14.00</i>	<i>-6.73</i>	<i>-10.02</i>	<i>-4.16</i>	<i>-5.34</i>	<i>1.13</i>	<i>-1.06</i>	<i>-8.95</i>	<i>-3.68</i>
Light-I	51.07	40.90	46.51	46.16	49.33	50.68	47.71	31.28	47.39	45.28
	<i>-8.29</i>	<i>-12.17</i>	<i>-3.58</i>	<i>-8.01</i>	<i>-3.22</i>	<i>-3.65</i>	<i>2.50</i>	<i>-1.10</i>	<i>-1.34</i>	<i>-1.36</i>
Light	54.97	41.74	47.85	48.19	47.26	49.56	46.56	44.49	48.97	47.39
	<i>-6.85</i>	<i>-11.40</i>	<i>-2.62</i>	<i>-6.96</i>	<i>-4.04</i>	<i>-4.67</i>	<i>2.09</i>	<i>-0.32</i>	<i>-4.81</i>	<i>-2.35</i>

Table 2: Results on two CL benchmarks with **LLaMA3.2-1B**. Averaged accuracy (AA $\uparrow$) (in black font) after training on the last task and *backward transfer (BWT $\uparrow$)* (in blue italic) is reported.

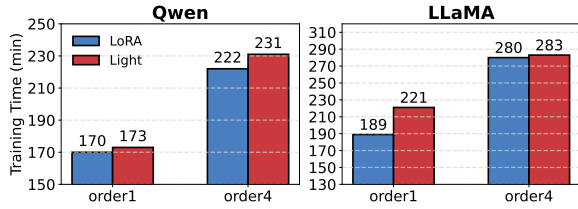


Figure 9: Time overhead.

For example, in LLaMA under order 4, the overhead is only about 1%. These results show that LightLoRA is efficient and adds **minimal computational cost**, making it ideal for SLMs.

4.5 Parameter Sensitivity Analysis

We investigate the sensitivity of LightLoRA to key hyperparameters (LoRA rank r , the conflict penalty λ , and the number of activated heads k).

Effect of LoRA Rank. Figure 10 shows how the LoRA rank affects our method’s performance across three different task orders. Overall, the average accuracy does not simply go up as the rank increases. Instead, it usually follows a parabolic trend, peaking at $r = 8$ or $r = 12$ depending on the task order. Specifically, a rank of 8 consistently gives good results across all sequences; therefore, we set $r = 8$ in experiments. When the rank is too low (e.g., $r = 4$), the model struggles to learn new

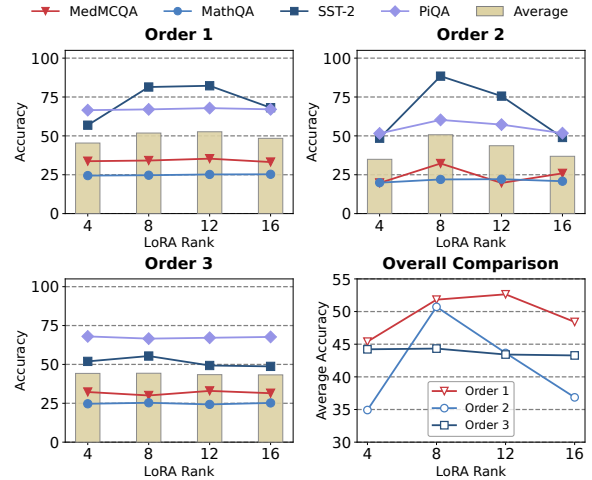


Figure 10: Effect of LoRA rank on accuracy across continual learning orders.

tasks effectively, resulting in poor performance. In contrast, when the rank is too high (e.g., $r = 16$), performance drops significantly, particularly in Order 2. This is because a larger rank increases the risk of overfitting.

Effect of λ . Figure 11 shows the impact of the hyperparameter λ on performance across different task orders. We observe that the effect of λ varies significantly between tasks. For tasks like MathQA and MedMCQA, the accuracy remains relatively stable as λ changes. And the tasks like CareQA

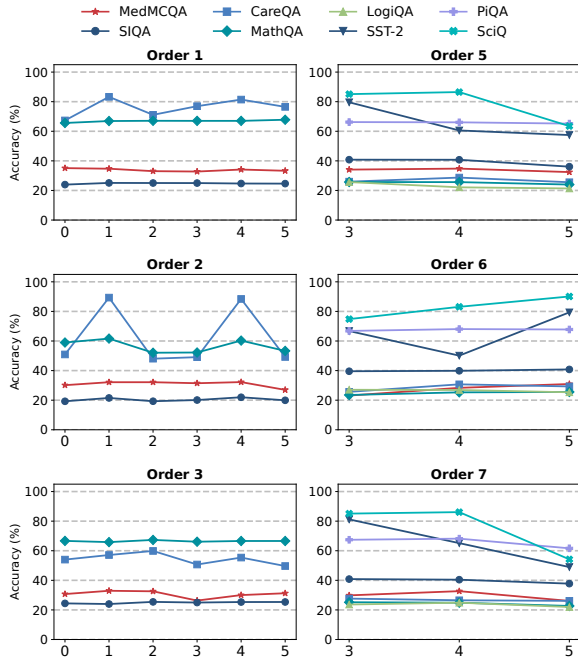


Figure 11: Sensitivity analysis of the hyperparameter λ across different task orders.

and SST-2 are sensitive to λ . Additionally, in Order 5 and 7, we see a clear downward trend for SciQ and SST-2 when λ increases from 4 to 5. This suggests that while penalizing gradient conflict helps keep old knowledge, a λ that is too large may be too restrictive. It might prevent the model from activating enough "important" heads for the current task, even if those heads have only minor conflicts with the past.

Effect of the number of activated heads. Figure 12 shows how the number of activated heads, k , affects performance. We observe that accuracy does not simply increase as k gets larger. For instance, in Order 6 and Order 7, the performance of tasks like SST-2 and SciQ drops significantly when k reaches 12. This is because updating too many heads makes the training process similar to standard full fine-tuning, which causes interference with past knowledge and leads to forgetting. Accordingly, we set $k = 4$ in all experiments, which keeps the number of updated heads minimal while incurring only a marginal performance drop.

5 Related Work

Continual Learning Research in continual learning can be categorized into: (1) Regularization-based methods (Kirkpatrick et al., 2017; Zenke et al., 2017; Li and Hoiem, 2017; Aljundi et al., 2018; Song et al., 2025) mitigate catastrophic forgetting by penalizing changes to model parameters

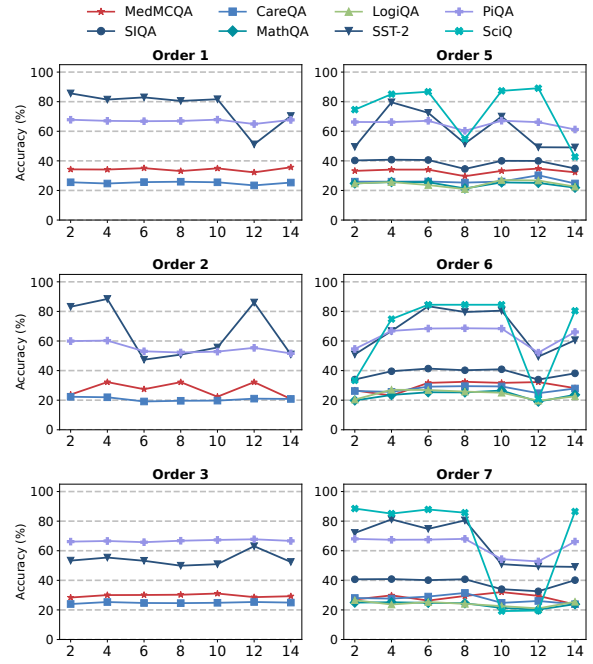


Figure 12: Effect of the number of activated heads (k) on performance.

to balance plasticity and stability. (2) Replay-based methods (Lopez-Paz and Ranzato, 2017; Chaudhry et al., 2018; Ramakrishnan et al., 2020; Huang et al., 2024) retain memory of old tasks by reloading or generating historical samples while learning new tasks. (3) Architecture-based methods (Wu et al., 2024; Jin and Kim, 2022; Aljundi et al., 2017; Ren et al., 2024) avoid forgetting by expanding the model structure and assigning dedicated parameter spaces to different tasks.

Small Language Models Research about SLMs have concentrated on the following areas. (1) Quantization and efficient inference. (Wang et al., 2025b) proposed the first systematic benchmarking framework, which revealed that conventional LLM quantization methods are ineffective for smaller models. (Zhou et al., 2025) systematically evaluated the effectiveness of pruning methods such as SparseGPT and Wanda, as well as quantization methods like GPTQ and AWQ, on SLMs. (2) Safety. (Zhang et al., 2025) performed the first systematic evaluation of jailbreak attack vulnerabilities in SLMs to uncover underlying security risks. (3) Collaboration between SLMs and LLMs. (Tan et al., 2024) proposed that an SLM can act as a guide to determine when and what to retrieve, thereby enhancing question-answering efficiency. (Vernikos et al., 2024) proposed that SLMs can refine multiple candidate outputs generated by LLMs. This approach significantly improves the overall

performance of natural language generation tasks.

6 Conclusion

In this paper, we address the challenge of catastrophic forgetting of Small Language Model (SLM) under continual learning. We found that SLMs are more susceptible to catastrophic forgetting because of their limited number of attention heads. To mitigate this issue, we propose LightLoRA, a framework that selectively updates task-important and low-conflict attention heads in the fine-tuning process while keeping the others frozen. Experimental results on Qwen1.5-0.5B and LLaMA3.2-1B demonstrate that LightLoRA significantly outperforms existing methods across multiple benchmarks.

7 Limitations

While LightLoRA demonstrates clear benefits, it has certain constraints. Currently, the method focuses exclusively on attention head selection. Given that Feed-Forward Networks (FFNs) also account for a substantial parameter count and contribute to knowledge loss, extending our framework to FFN adaptation remains a key priority for future exploration.

8 Acknowledgements

We sincerely thank the anonymous reviewers for their insightful and detailed comments. This work is supported by the Science and Technology Innovation Program of Hunan Province under Grants 2025RC3121 and 2024RC1048.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154.
- Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. 2017. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3366–3375.
- Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 2357–2367.
- Anna Arias-Duart, Pablo Agustin Martin-Torres, Daniel Hinjos, Pablo Bernabeu-Perez, Lucia Urcelay Ganza-bal, Marta Gonzalez Mallo, Ashwin Kumar Gururajan, Enrique Lopez-Cuena, Sergio Alvarez-Napagao, and Dario Garcia-Gasulla. 2025. Automatic evaluation of healthcare llms beyond question-answering. *arXiv preprint arXiv:2502.06666*.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small language models are the future of agentic ai. *arXiv preprint arXiv:2506.02153*.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Arslan Chaudhry, Marc’Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. 2018. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*.
- Mateusz Dubiel, Yasmine Barghouti, Kristina Kudryavtseva, and Luis A Leiva. 2024. On-device query intent prediction with lightweight llms to support ubiquitous conversations. *Scientific reports*, 14(1):12731.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. [The language model evaluation harness](#).
- 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*.

- Jinghan He, Haiyun Guo, Ming Tang, and Jinqiao Wang. 2023. Continual instruction tuning for large multi-modal models. *arXiv preprint arXiv:2311.16206*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023. Lorahub: Efficient cross-task generalization via dynamic lora composition. *arXiv preprint arXiv:2307.13269*.
- Jianheng Huang, Leyang Cui, Ante Wang, Chengyi Yang, Xinting Liao, Linfeng Song, Junfeng Yao, and Jinsong Su. 2024. Mitigating catastrophic forgetting in large language models with self-synthesized rehearsal. *arXiv preprint arXiv:2403.01244*.
- Hyundong Jin and Eunwoo Kim. 2022. Helpful or harmful: Inter-task association in continual learning. In *European conference on computer vision*, pages 519–535. Springer.
- William B Johnson, Joram Lindenstrauss, and 1 others. 1984. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.
- Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.
- Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.
- David Lopez-Paz and Marc’Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.
- Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. 2025. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19165–19181.
- Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D Lane, and Mengwei Xu. 2024. Small language models: Survey, measurements, and insights. *arXiv preprint arXiv:2409.15790*.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32.
- Ankit Pal, Logesh Kumar Umapathi, and Malaikanan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR.
- Kandan Ramakrishnan, Rameswar Panda, Quanfu Fan, John Henning, Aude Oliva, and Rogerio Feris. 2020. Relationship matters: Relation guided knowledge transfer for incremental learning of object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 250–251.
- Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madihan Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. *arXiv preprint arXiv:2301.12314*.
- Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.
- Weijie Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. 2024. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. *arXiv preprint arXiv:2402.18865*.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialliqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
- Shezheng Song, Hao Xu, Jun Ma, Shasha Li, Long Peng, Qian Wan, Xiaodong Liu, and Jie Yu. 2025. How to alleviate catastrophic forgetting in llms fine-tuning? hierarchical layer-wise and element-wise regularization. *arXiv preprint arXiv:2501.13669*.
- Jiejun Tan, Zhicheng Dou, Yutao Zhu, Peidong Guo, Kun Fang, and Ji-Rong Wen. 2024. Small models, big insights: Leveraging slim proxy models to decide when and what to retrieve for llms. *arXiv preprint arXiv:2402.12052*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutika Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

- Giorgos Vernikos, Arthur Bražiškas, Jakub Adamek, Jonathan Mallinson, Aliaksei Severyn, and Eric Malmi. 2024. Small language models improve giants by rewriting their outputs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2703–2718.
- Elena Voita, David Talbot, Fedor Moiseev, Rico Senrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 5797–5808.
- Chengyu Wang, Junbing Yan, Yuanhao Yue, and Jun Huang. 2025a. Distilqwen2. 5: Industrial practices of training distilled open lightweight language models. *arXiv preprint arXiv:2504.15027*.
- Jiacheng Wang, Yejun Zeng, Jinyang Guo, Yuqing Ma, Aishan Liu, and Xianglong Liu. 2025b. Slmquant: Benchmarking small language model quantization for practical deployment. In *Proceedings of the 3rd International Workshop on Rich Media With Generative AI*, pages 2–10.
- Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. 2023. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671.
- Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024a. In-scl: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. *arXiv preprint arXiv:2403.11435*.
- Zhiqi Wang, Shizhu He, Kang Liu, and Jun Zhao. 2024b. Instance-level dynamic loras composition for cross-task generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5699–5708.
- Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 139–149.
- Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*.
- Chengyue Wu, Yukang Gan, Yixiao Ge, Zeyu Lu, Jiahao Wang, Ye Feng, Ying Shan, and Ping Luo. 2024. Llama pro: Progressive llama with block expansion. *arXiv preprint arXiv:2401.02415*.
- Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR.
- Keke Zhai. 2024. A post-training enhanced optimization approach for small language models. *arXiv preprint arXiv:2411.02939*.
- Wenhui Zhang, Huiyu Xu, Zhibo Wang, Zeqing He, Ziqi Zhu, and Kui Ren. 2025. Can small language models reliably resist jailbreak attacks? a comprehensive evaluation. *arXiv preprint arXiv:2503.06519*.
- Zihan Zhou, Simon Kurz, and Zhixue Zhao. 2025. Revisiting pruning vs quantization for small language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12055–12070.

A Appendix

A.1 Theoretical Study on How Attention Heads Affect Continual Learning

The empirical results in Section 2.1 show that SLMs suffer from more severe forgetting. We now explain *why* from a geometric perspective.

A.1.1 Heads as Independent Learning Slots

In multi-head attention, each head h projects the input into its own d_h -dimensional subspace via $W_Q^h, W_K^h \in \mathbb{R}^{d_{\text{model}} \times d_h}$, where $d_h = d_{\text{model}}/H$. Under LoRA fine-tuning, the adaptation $\Delta W^h = B^h A^h$ is applied independently per head. This means each head acts as an independent “learning slot” that can be tuned for a specific task without directly affecting other heads.

Prior work has shown that different heads naturally specialize in different linguistic functions (Voita et al., 2019; Michel et al., 2019). In continual learning, this specialization becomes critical: ideally, different tasks can occupy different heads, avoiding mutual interference.

A.1.2 The Head Sharing Problem

Suppose each task requires modifying at least κ heads to achieve acceptable performance. With H heads in total, the maximum number of tasks that can be learned without any head overlap is simply:

$$T_{\max} = \left\lfloor \frac{H}{\kappa} \right\rfloor \quad (7)$$

For example, with $\kappa = 4$: Qwen1.5-0.5B ($H = 16$) gives $T_{\max} = 4$, while Qwen1.5-7B ($H = 32$) gives $T_{\max} = 8$. Once the number of tasks exceeds $T_{\max}$, some heads must be shared across tasks, and modifying a shared head for a new task risks overwriting what it learned for previous tasks.

The reason is straightforward: when there are fewer available heads, different tasks are forced to share the same heads earlier, and forgetting begins sooner.

A.1.3 Why Sharing Causes Interference

When two tasks share a head, their respective LoRA updates ΔW_i^h and ΔW_j^h coexist in the same d_h -dimensional subspace. The question becomes: how many task-specific representations can a single d_h -dimensional space accommodate before they start interfering with each other.

The Johnson-Lindenstrauss (JL) lemma (Johnson et al., 1984) provides a well-known answer

to this type of question. It states that preserving pairwise distances among n points with distortion at most ε requires a space of dimension at least $d = \Omega(\varepsilon^{-2} \log n)$. Applying this to our setting, where each task’s adaptation defines a characteristic direction in the head’s subspace, the number of task representations that can coexist in a d_h -dimensional head without significant distortion is bounded by:

$$T_{\text{per-head}} \leq \exp(c \cdot d_h \cdot \varepsilon^2) \quad (8)$$

where c is a constant and ε controls the tolerable distortion.

Since $d_h = d_{\text{model}}/H$, increasing the number of heads H (as in LLMs) reduces d_h and thus reduces per-head capacity. However, the total number of tasks the model can handle is approximately $H \times T_{\text{per-head}}$. More heads means more independent slots, even though each slot is smaller. The net effect favors more heads: the linear gain from having more independent slots outweighs the logarithmic loss in per-slot capacity.

For SLMs with small H , both factors work against them: fewer independent slots *and* each slot must accommodate more tasks, leading to a compounding disadvantage in continual learning.

A.1.4 Implication for LightLoRA

The above analysis establishes a structural connection between attention head count and continual learning capacity. Specifically, the limited number of heads in SLMs constrains the availability of independent representational subspaces, which in turn forces premature sharing of heads across tasks and amplifies inter-task interference. This theoretical understanding directly motivates the two core design principles of LightLoRA:

- **Sparse activation.** By updating only k out of H heads per task, we effectively increase $T_{\max}$ in Eq. (7). The remaining $H - k$ heads are preserved as protected slots for previously learned knowledge.
- **Conflict-aware selection.** Among candidate heads, we prefer those whose gradient directions are orthogonal to historical gradients. Orthogonal updates utilize unused directions in the subspace, minimizing distortion to existing task representations. This is precisely the condition under which the JL bound is most favorable.

A.2 Baselines

We compare our proposed method against various representative continual learning baselines.

- **SeqLoRA** (Hu et al., 2022): Performs LoRA fine-tuning sequentially according to the task order, which is the most basic continual learning setting.
- **L2P** (Wang et al., 2022): A continual learning method based on prompt learning that mitigates forgetting of old knowledge by learning a set of adjustable prompts for each task.
- **GEM** (Lopez-Paz and Ranzato, 2017): Ensures that current task updates do not increase the loss of old tasks by storing old task samples and computing gradient constraints, thereby preserving old knowledge.
- **CLoRA** (Lu et al., 2025): Introduces subspace regularization on top of LoRA to balance new and old knowledge by constraining the direction of parameter updates.
- **EWC** (Kirkpatrick et al., 2017): Adds an elastic regularization term to important parameters to keep them close to their optimal values for old tasks during new task training, thereby mitigating forgetting.
- **OLoRA** (Wang et al., 2023): Restricts learning updates for new tasks to a subspace that is orthogonal to the representation space of previous tasks, which effectively eliminates inter-task interference and mitigates catastrophic forgetting.

Hyperparameter	
LoRA α	32
Dropout	0.1
Weight decay rate	0
Optimizer	AdamW
Learning Rate	8e-4
LR Scheduler	Linear
Batch Size	20
Warmup Steps	0
Epochs	1
LoRA target modules	query, value

Table 3: Hyperparameters for experiment settings

A.3 Implementation Details

Table 3 shows our detailed experiment hyperparameters. We used the Huggingface *peft* library and *Accelerate* library for training. For the Qwen model, we set the λ values to 4, 4, 4, 3, 3, 3, 3, and 3 for the sequences from order1 to order8. For the LLaMA model, we set the values of λ to 4, 4, 4, 1, 2, 1, 2, and 2 for order1 through order8. In our experiments, we set the number of selected attention heads to four and used a selection interval of 50 batches.

A.4 Datasets and Task Sequence Orders

Table 4 shows detailed information about the eight datasets used in our experiments, as well as the evaluation metrics. We report orders used for our Continual Learning experiments in Table 5.

Standard CL Benchmark. In standard Continual Learning evaluations, we selected four representative tasks: MathQA (mathematical reasoning), SST-2 (sentiment classification), PIQA (physical commonsense reasoning), and MedMCQA (medical question answering).

Large Number of Tasks. We introduced eight datasets covering various types. These include classification (SST-2), medical QA (CareQA, MedMCQA), math and logic (MathQA, LogiQA), and commonsense reasoning (SocialIQA, PIQA, SciQ).

We conduct tests on these tasks based on the *lm-eval* (Gao et al., 2024) framework. Accordingly, we report task accuracy as the primary metric.

A.5 Comparison with Parameter-Isolated Approaches

To validate the advantage of our unified adapter approach over parameter-isolated LoRA routing, we conduct comparisons under three realistic deployment scenarios on LLaMA-3.2-1B.

Scenario 1: Known Task Identity. Even when task identity is perfectly known, LoRA routing incurs significant overhead. As shown in Table 6, the mean per-sample inference time increases from 1.81s to 2.23s (a 23.2% increase) due to adapter switching. Additionally, maintaining 8 task-specific adapters requires 352MB of storage, which is 14.6% of the 2.4GB base model.

Scenario 2: Inferred Task Identity. We evaluate a classify-then-answer pipeline where a domain classifier (85.27% accuracy) routes queries to task-specific adapters. Misclassification directly degrades downstream performance, as shown in

Dataset Name	# Train	# Test	Domain	Metric
MathQA (Amini et al., 2019)	29,837	2,985	mathematical reasoning	accuracy
SST-2 (Socher et al., 2013)	67,349	1,821	sentiment analysis	accuracy
PIQA (Bisk et al., 2020)	16,113	3,003	physical commonsense	accuracy
MedMCQA (Pal et al., 2022)	182,822	6,150	medical QA	accuracy
LogiQA (Liu et al., 2020)	7,376	651	logical reasoning	accuracy
SocialIQA (Sap et al., 2019)	33,410	1,954	social commonsense	accuracy
CareQA (Arias-Duart et al., 2025)	5,621	1,000	healthcare	accuracy
SciQ (Welbl et al., 2017)	11,679	1,000	science	accuracy

Table 4: The details of 8 datasets used in our CL experiments. First four tasks correspond to the standard CL benchmark, all other tasks are used in long-sequence experiments.

Order	Task Sequence
1	mathqa → sst2 → piqa → medmcqa
2	piqa → mathqa → medmcqa → sst2
3	sst2 → medmcqa → piqa → mathqa
4	socialiqa → piqa → logiqa → medmcqa → careqa → mathqa → sciq → sst2
5	socialiqa → careqa → mathqa → logiqa → sst2 → piqa → sciq → medmcqa
6	careqa → mathqa → socialiqa → sst2 → medmcqa → sciq → piqa → logiqa
7	logiqa → careqa → socialiqa → medmcqa → sst2 → sciq → piqa → mathqa
8	mathqa → socialiqa → careqa → logiqa → piqa → sst2 → medmcqa → sciq

Table 5: Eight different orders of task sequences used for continual learning experiments. Orders 1-3 correspond to the standard CL benchmark. Orders 4-8 are large number of tasks benchmark.

Table 6: Inference cost comparison on LLaMA-3.2-1B (SciQ task).

Time (s)	Ours (CL)	LoRA Routing
Mean	1.81	2.23
P50	1.79	2.31
P95	1.83	2.32
Max	1.84	2.54

Table 7.

Table 7: Task accuracy (%) under inferred task identity via domain classifier routing on LLaMA-3.2-1B.

Method	MathQA	PiQA	MedMCQA	SciQ
LoRA Routing	27.04	63.88	40.21	81.29
Ours (CL)	32.92	75.21	46.44	95.00

Scenario 3: Cross-Domain Generalization.

We evaluate generalization to SciQ (unseen task) by comparing individual task-specific adapters against our continual learning adapter trained sequentially on MathQA, PiQA, and MedMCQA. As shown in Table 8, sequential learning enables beneficial knowledge accumulation that isolated adapters cannot achieve.

Table 8: Cross-domain transfer to unseen task SciQ on LLaMA-3.2-1B.

Method	SciQ Acc (%)
MathQA Adapter	79.3
PiQA Adapter	81.8
MedMCQA Adapter	83.1
Ours (CL)	92.8

A.6 Analysis on Conflict Score Design

A key design choice in our conflict-aware importance mechanism (eq. (4)) is the use of absolute value $|C_t(h)|$, which treats all non-orthogonal gradient directions equally without distinguishing positive from negative transfer. An alternative sign-aware formulation that explicitly models the transfer-interference tradeoff is:

$$S_t(h)_{\text{sign}} = I_t(h) \times \max(0, 1 + \lambda \cdot C_t(h)) \quad (9)$$

This variant encourages updating heads with positive transfer ($C_t(h) > 0$) while suppressing those with negative transfer ($C_t(h) < 0$).

We compare both formulations on Qwen1.5-0.5B across three task orders. As shown in Table 9, our absolute-value formulation consistently

outperforms the sign-aware variant, with a substantial average improvement of 3.98 points (48.96 vs. 44.98). The advantage is most pronounced on Order 2 (+12.12), where the sign-aware variant suffers from aggressive head reuse driven by perceived positive transfer.

Table 9: Study on conflict score design on Qwen. “Sign-Aware” preserves gradient sign; “Ours” uses absolute value $|C_t(h)|$.

Qwen	Order1	Order2	Order3	Average
Sign-Aware	51.65	38.61	43.69	44.98
Ours	51.82	50.73	44.34	48.96

This result validates our design rationale: in SLMs with severely limited attention heads, *capacity preservation* through orthogonality is more effective than attempting to exploit positive transfer. Even beneficial gradient alignment consumes scarce representational capacity, and continually modifying a head, even in a positive direction, leads to parameter drift that gradually erodes previously learned knowledge. By treating all non-orthogonal directions equally, our formulation maximally preserves unused parameter space for future tasks.

A.7 Visualization

To illustrate our attention head selection process in more detail, Figure 13 shows the specific selection patterns for each task within the Order 5 sequence using our method. We can observe that the activation patterns change significantly from task to task. Instead of repeatedly using the same set of heads, the model dynamically selects different subsets of heads in each layer for different tasks. This behavior suggests that LightLoRA can effectively identify and utilize available parameter subspaces for new tasks, which helps minimize interference with previously learned knowledge and reduces the risk of catastrophic forgetting.

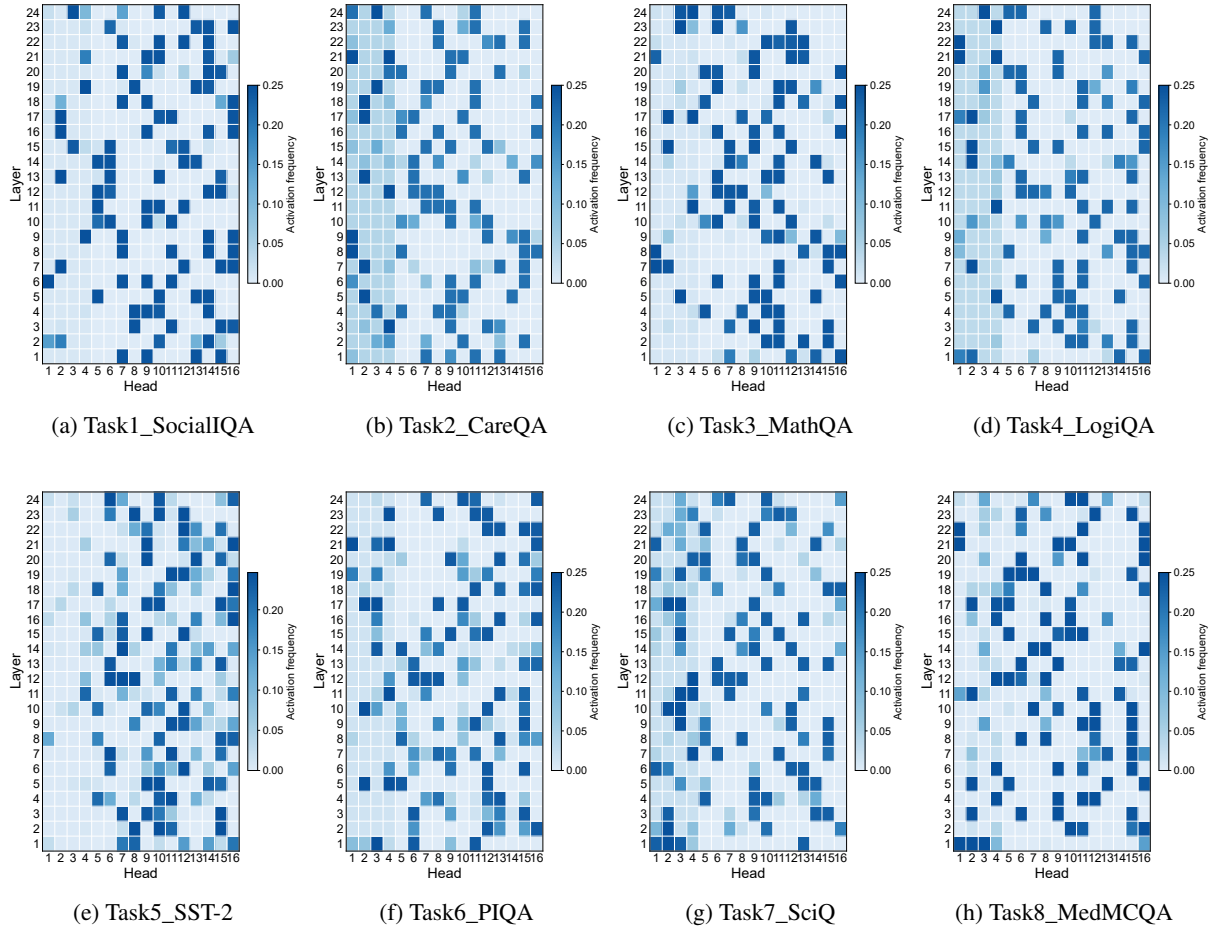


Figure 13: When using the LightLoRA method to learn the order5 sequence, the activation frequency of attention heads in each Transformer layer is recorded after completing each task. In Task 1, attention heads are selected solely based on the raw importance.

Qwen-1.5-0.5B		MedMCQA	SiQA	CareQA	MathQA	LogiQA	SST2	PiQA	SciQ	Average
Random2	Order 1	34.28	-	-	25.63	-	58.14	67.95	-	46.50
	Order 2	32.23	-	-	20.80	-	54.70	55.33	-	40.77
	Order 3	27.42	-	-	25.26	-	49.20	67.19	-	42.27
	Order 4	27.80	33.11	25.53	21.51	20.12	51.49	52.18	19.90	31.45
	Order 5	21.73	32.80	26.29	20.60	20.43	52.18	51.47	19.00	30.56
	Order 6	31.87	34.39	24.89	21.68	20.58	49.77	52.18	18.30	31.71
	Order 7	30.29	35.77	27.02	23.25	24.88	49.20	65.13	68.90	40.56
	Order 8	26.85	41.40	26.31	26.13	26.42	70.18	68.01	91.00	47.04
Random4	Order 1	34.19	-	-	24.82	-	71.90	68.34	-	49.81
	Order 2	19.72	-	-	19.83	-	50.69	52.88	-	35.78
	Order 3	32.92	-	-	25.19	-	50.92	67.63	-	44.17
	Order 4	27.97	32.70	25.23	19.50	18.74	50.92	52.39	19.90	30.92
	Order 5	25.94	33.93	26.19	20.47	20.43	49.20	51.90	20.30	31.05
	Order 6	25.60	33.83	26.29	20.64	20.74	49.89	51.47	20.20	31.08
	Order 7	28.50	41.15	28.77	25.53	25.96	51.61	67.08	81.80	43.80
	Order 8	32.23	34.29	24.71	19.33	20.89	51.72	52.12	19.40	31.84
LORA	Order 1	35.14	-	-	25.33	-	53.10	66.76	-	45.08
	Order 2	20.44	-	-	22.85	-	80.62	60.99	-	46.23
	Order 3	25.72	-	-	26.33	-	53.78	67.08	-	43.23
	Order 4	30.36	33.37	25.60	19.43	20.28	49.43	51.90	19.90	31.28
	Order 5	27.95	33.52	25.28	19.80	19.51	50.92	51.69	20.60	31.16
	Order 6	31.41	34.54	24.82	21.14	22.89	51.15	56.80	42.90	35.71
	Order 7	32.13	33.62	24.66	21.78	20.74	48.51	52.29	20.10	31.73
	Order 8	32.23	32.86	24.71	20.17	19.97	50.46	52.39	21.80	31.82
CLoRA	Order 1	34.86	-	-	25.39	-	61.31	57.30	-	44.72
	Order 2	32.23	-	-	19.13	-	48.51	52.01	-	37.97
	Order 3	29.98	-	-	26.00	-	67.19	52.18	-	43.84
	Order 4	30.62	34.19	25.42	17.69	21.51	40.92	32.88	20.60	27.98
	Order 5	34.07	39.87	27.59	24.49	26.88	54.24	56.27	77.20	42.58
	Order 6	30.55	30.28	30.07	25.23	27.04	54.66	68.99	57.50	40.54
	Order 7	29.88	30.84	29.99	26.70	25.96	55.28	66.70	62.80	41.02
	Order 8	31.15	41.91	25.05	26.06	25.19	54.13	65.23	78.50	43.40
GEM	Order 1	36.12	-	-	24.96	-	70.62	67.52	-	49.81
	Order 2	31.99	-	-	18.63	-	50.92	52.01	-	38.39
	Order 3	26.70	-	-	25.23	-	54.24	66.65	-	43.21
	Order 4	20.08	33.47	23.48	20.80	19.97	41.24	43.70	32.50	29.41
	Order 5	34.71	40.63	27.74	24.42	24.73	49.66	67.46	78.00	43.42
	Order 6	29.55	40.02	28.27	24.99	27.19	53.33	67.08	68.00	33.89
	Order 7	32.85	41.86	26.81	25.09	22.73	59.27	46.65	63.80	39.88
	Order 8	32.23	34.08	24.71	19.63	20.89	48.97	51.47	22.20	31.77
L2P	Order 1	32.23	-	-	18.73	-	49.08	51.74	-	37.95
	Order 2	19.72	-	-	20.64	-	49.08	52.56	-	35.50
	Order 3	24.46	-	-	27.04	-	69.95	48.28	-	42.43
	Order 4	26.92	33.27	25.53	18.32	21.20	39.08	43.26	19.20	28.35
	Order 5	32.23	33.57	24.71	19.13	20.74	47.25	52.77	19.60	31.25
	Order 6	32.35	37.77	24.62	23.25	25.19	50.92	63.71	58.60	39.55
	Order 7	29.57	41.56	25.74	26.93	26.11	59.29	67.68	56.00	41.61
	Order 8	32.11	34.54	24.80	19.56	21.35	49.31	55.55	49.10	35.79
EWC	Order 1	33.28	-	-	24.59	-	75.57	67.95	-	50.35
	Order 2	32.23	-	-	22.58	-	51.15	54.46	-	40.11
	Order 3	26.78	-	-	25.13	-	62.25	58.34	-	43.13
	Order 4	32.23	37.72	24.71	23.65	24.73	49.93	61.21	31.70	35.74
	Order 5	34.35	41.97	29.07	25.56	25.35	60.89	56.92	47.20	40.16
	Order 6	32.58	41.97	29.57	23.62	26.11	50.23	57.30	57.20	39.82
	Order 7	31.34	39.66	26.70	24.62	23.20	48.97	67.36	65.50	40.92
	Order 8	33.25	34.21	25.08	21.94	18.75	36.22	54.83	66.69	36.37
OLoRA	Order 1	30.72	-	-	25.36	-	51.26	66.87	-	43.55
	Order 2	26.18	-	-	23.89	-	70.99	67.19	-	47.06
	Order 3	26.68	-	-	26.33	-	51.72	66.65	-	42.85
	Order 4	26.85	42.37	28.30	24.82	23.04	44.36	64.25	44.80	37.35
	Order 5	29.88	42.78	28.95	26.16	24.27	52.06	67.36	74.40	43.23
	Order 6	30.48	42.58	27.61	24.42	23.81	52.87	66.81	76.00	43.07
	Order 7	30.74	42.07	28.96	26.60	24.12	52.41	67.52	83.90	44.54
	Order 8	31.94	43.14	26.90	24.32	25.35	52.98	67.08	87.50	44.90
Light (Ours)	Order 1	34.14	-	-	24.69	-	81.42	67.03	-	51.82
	Order 2	32.23	-	-	21.98	-	88.42	60.28	-	50.73
	Order 3	30.03	-	-	25.33	-	55.39	66.59	-	44.34
	Order 4	26.15	39.97	25.51	23.12	25.81	52.18	62.46	66.20	40.18
	Order 5	34.09	40.79	25.94	25.70	25.65	79.59	66.21	85.10	47.88
	Order 6	23.12	39.56	25.64	23.52	27.04	66.74	66.81	74.80	43.40
	Order 7	29.84	40.84	27.65	25.19	23.66	81.19	67.41	85.10	47.61
	Order 8	32.61	42.73	27.82	25.70	26.11	67.89	68.28	91.20	47.79

Table 10: Qwen Detailed performance.

LLaMA-3.2-1B		MedMCQA	SiQA	CareQA	MathQA	LogiQA	SST2	PiQA	SciQ	Average
Random2	Order 1	26.13	-	-	19.66	-	50.92	52.29	-	37.25
	Order 2	26.15	-	-	19.83	-	45.41	51.41	-	35.70
	Order 3	34.69	-	-	30.42	-	44.11	72.36	-	45.39
	Order 4	30.79	33.67	24.71	19.70	20.28	49.31	52.77	23.40	31.83
	Order 5	27.28	39.87	25.81	23.79	22.89	54.13	69.42	70.50	41.71
	Order 6	23.57	32.96	25.89	19.26	21.35	52.98	52.83	19.40	31.03
	Order 7	32.18	32.96	24.71	20.27	21.35	49.08	52.01	19.00	31.45
	Order 8	32.23	33.47	24.71	18.66	21.51	49.08	52.61	21.40	31.71
Random4	Order 1	42.12	-	-	29.95	-	72.82	63.12	-	52.00
	Order 2	32.23	-	-	20.37	-	57.45	53.65	-	40.93
	Order 3	36.46	-	-	30.45	-	61.12	61.82	-	47.46
	Order 4	19.94	34.14	22.84	19.16	20.12	50.92	52.61	18.90	29.83
	Order 5	23.26	33.88	25.87	20.30	22.27	50.46	52.83	18.10	30.87
	Order 6	32.20	40.89	30.14	27.04	25.50	56.54	71.71	89.20	46.65
	Order 7	22.19	32.65	23.13	20.40	21.97	49.08	41.96	19.10	28.81
	Order 8	20.68	33.11	24.02	17.69	20.28	54.01	52.29	20.90	30.37
LORA	Order 1	44.35	-	-	30.75	-	52.64	62.91	-	47.66
	Order 2	29.93	-	-	18.53	-	48.74	51.90	-	37.28
	Order 3	36.74	-	-	29.38	-	55.16	61.65	-	45.73
	Order 4	22.54	41.30	34.78	19.61	23.35	49.08	51.76	48.00	36.30
	Order 5	42.55	42.68	33.96	31.62	25.50	44.45	73.29	72.40	45.81
	Order 6	29.88	42.78	30.26	28.71	21.51	49.08	70.02	79.60	43.98
	Order 7	28.16	33.42	25.33	19.97	19.51	40.69	52.50	21.00	30.07
	Order 8	37.87	42.94	32.63	32.09	26.57	39.86	53.56	54.30	39.98
CLoRA	Order 1	43.87	-	-	31.62	-	59.98	73.56	-	52.26
	Order 2	25.94	-	-	19.10	-	51.26	52.88	-	37.30
	Order 3	36.03	-	-	31.56	-	59.29	61.60	-	47.12
	Order 4	28.85	33.93	24.69	18.56	20.43	50.69	51.41	21.10	31.21
	Order 5	22.38	34.34	26.19	20.27	20.28	47.82	51.90	20.20	30.42
	Order 6	32.54	41.04	31.35	29.75	21.97	50.69	71.33	85.70	45.55
	Order 7	31.89	26.57	24.82	19.36	19.51	45.92	52.72	20.90	30.21
	Order 8	32.23	34.85	24.71	18.49	19.97	51.03	52.23	22.10	31.95
GEM	Order 1	44.23	-	-	30.75	-	62.96	73.34	-	52.82
	Order 2	32.20	-	-	18.36	-	49.08	51.85	-	37.87
	Order 3	37.34	-	-	29.38	-	52.64	61.98	-	45.34
	Order 4	32.27	42.78	30.14	30.99	25.19	40.34	62.74	72.90	42.17
	Order 5	32.82	32.63	34.01	31.73	25.65	43.85	73.56	72.40	43.33
	Order 6	32.97	40.84	32.18	29.72	23.81	55.60	72.14	69.80	44.63
	Order 7	25.23	32.55	24.66	19.73	21.20	48.74	52.45	20.80	30.67
	Order 8	35.48	41.76	32.95	32.23	22.43	50.21	73.78	72.00	45.11
L2P	Order 1	24.41	-	-	19.60	-	49.31	52.50	-	36.46
	Order 2	32.23	-	-	20.87	-	48.85	53.16	-	38.78
	Order 3	22.35	-	-	17.65	-	48.39	53.16	-	35.39
	Order 4	32.23	32.19	24.71	19.66	21.66	51.83	53.05	21.60	32.12
	Order 5	25.05	32.60	25.85	20.23	24.88	49.89	53.86	19.40	31.47
	Order 6	32.15	34.24	24.50	18.59	21.81	49.08	51.85	20.10	31.54
	Order 7	32.20	33.52	24.62	19.20	23.20	43.21	48.32	20.00	30.53
	Order 8	29.40	34.14	24.78	18.86	22.58	49.08	52.50	19.60	31.37
EWC	Order 1	40.14	-	-	30.92	-	49.08	73.12	-	48.32
	Order 2	32.30	-	-	18.79	-	52.06	52.12	-	38.82
	Order 3	33.66	-	-	29.75	-	49.08	70.95	-	45.86
	Order 4	28.55	31.56	27.57	21.59	24.12	29.08	52.91	42.70	32.26
	Order 5	37.15	41.97	36.61	30.49	25.50	30.23	62.03	73.30	42.16
	Order 6	34.59	41.81	30.55	28.38	24.58	51.72	70.35	89.80	46.47
	Order 7	31.34	32.70	24.80	21.94	21.20	31.77	46.91	31.20	30.23
	Order 8	36.77	42.37	37.40	30.59	25.04	51.95	72.80	83.50	47.55
OLoRA	Order 1	39.35	-	-	32.43	-	54.01	72.20	-	49.50
	Order 2	30.15	-	-	19.43	-	49.43	54.08	-	38.27
	Order 3	32.85	-	-	30.12	-	54.24	72.36	-	47.39
	Order 4	41.00	41.35	42.55	33.27	26.73	55.85	53.88	73.80	46.05
	Order 5	41.05	41.50	41.56	33.30	27.19	59.40	53.50	73.60	46.39
	Order 6	40.78	41.76	41.56	33.13	24.88	65.48	54.16	83.50	48.16
	Order 7	38.01	41.61	38.62	33.50	25.96	51.26	52.31	54.00	41.91
	Order 8	40.78	41.50	42.16	33.30	27.96	57.11	53.72	74.40	46.37
Light (Ours)	Order 1	43.58	-	-	30.85	-	71.67	73.78	-	54.97
	Order 2	29.60	-	-	18.46	-	63.19	55.71	-	41.74
	Order 3	35.74	-	-	30.65	-	52.64	72.36	-	47.85
	Order 4	33.16	42.17	31.58	30.32	24.58	49.20	73.56	93.50	47.26
	Order 5	42.72	41.81	40.47	29.51	27.34	50.11	72.31	92.20	49.56
	Order 6	32.11	42.99	32.41	28.84	23.66	49.54	72.74	90.20	46.56
	Order 7	31.46	40.43	27.75	29.31	22.89	49.77	70.89	83.40	44.49
	Order 8	35.41	42.12	34.89	30.79	22.73	59.40	71.82	94.60	48.97

Table 11: LLaMA Detailed performance.