SAPT: A Shared Attention Framework for Parameter-Efficient Continual Learning of Large Language Models

Anonymous ACL submission

Abstract

The continual learning (CL) ability is vital for deploying large language models (LLMs) in the dynamic world. Existing methods devise the learning module to acquire task-specific knowledge with parameter-efficient tuning (PET) block and the selection module to pick out the corresponding one for the testing input, aiming at handling the challenges of catastrophic forgetting and knowledge transfer in CL. However, these methods tend to address only one of the challenges, ignoring the potential of aligning the two modules to effectively address catastrophic forgetting and knowledge transfer simultaneously. To this end, we propose a novel Shared Attention Framework (SAPT), to align the PET learning and selection via the Shared Attentive Learning & Selection module. Extensive Experiments on two CL benchmarks demonstrate the superiority of SAPT. Moreover, SAPT consistently demonstrates its superiority when we scale it to different model sizes (from 770M to 13B), different model architectures (T5 and LLaMA-2) and unseen tasks.[1]

1 Introduction

Endowing the continual learning (CL) ability for large language models (LLMs) (Brown et al., 2020; Raffel et al., 2020; Touvron et al., 2023) to learn different tasks sequentially is crucial for their deployment in the real-world, which allows them to dynamically adapt to novel tasks and acquire additional knowledge (Luo et al., 2023; Zhai et al., 2023). However, this scenario presents two significant challenges: (1) Catastrophic Forgetting (CF), referring to the loss of previously acquired knowledge when learning new tasks (McCloskey and Cohen, 1989), and (2) Knowledge Transfer (KT), involving the efficient utilization of knowledge from past tasks to facilitate the learning of new ones (Ke and Liu, 2022).

[1]Our data and codes could be found in supplementary files.

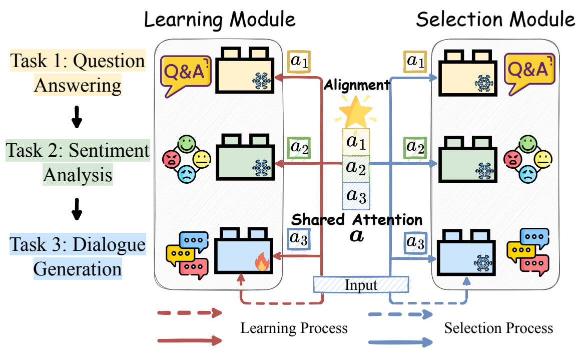


Figure 1: The conceptual framework for the learning and the selection module to achieve the continual learning of large language models based on PET blocks when the new Dialogue Generation task arrives. Dashed lines represent the working process of existing works while solid lines are for that of our SAPT in this work.

Due to the heavy burden on computation resources, recent attempts study the CL of LLMs based on parameter-efficient tuning (PET) methods (Hu et al., 2021; Ding et al., 2022). Inspired by the parameter isolation CL methods (Rusu et al., 2016; Fernando et al., 2017), existing methods can be conceptualized as two pivotal components working in the pipeline fashion. As shown in Figure 1 (dashed lines), when a new Dialogue Generation task arrives, a private PET block is allocated by the *learning module* to acquire task-specific knowledge and then saved to the PET pool for the following *selection module* to pick it out when a test sample is coming. However, the designs of each module in current works exhibit certain limitations in effectively dealing with KT and CF challenges.

On one hand, the design of *learning module* is supposed to function to facilitate KT among different tasks. Unfortunately, for existing works, the learning of PET block is either performed seperately within each single task (Wang et al., 2023b), or kept orthogonal to each other to minimize interference (Wang et al., 2023a). Such isolation cuts off the potential transfer of acquired knowledge stored in the previous PET blocks and hinders them to assist the current acquisition of new knowledge.

On the other hand, the *selection module* plays the pivotal roles in mitigating CF because only when it is capable of automatically selecting the PET block to which the current input belongs can the LLM backbone successfully accomplish the current task. However, it would make LLMs vulnerable to CF by simply implementing such selection process via the summation (Wang et al., 2023a) or concatenation (Razdaibiedina et al., 2023) of all existing PET blocks or selecting them from a fixed PET pool (Wang et al., 2022b).

More importantly, they ignore the opportunity of aligning the two modules to address challenges of CF and KT simultaneously. The intuition is that (illustrated by solid lines in Figure 1), in order to facilitate KT in the learning of the new task, the learning module should rely on task correlations to leverage the most relevant knowledge in previous PET blocks. And such attentive process, expressed as **shared attention** in our study, could be naturally repeated by the selection module to resist CF through the combination of the corresponding PET blocks belonging to each testing input. As a result, the end-to-end alignment of these two modules is established via such shared attention.

To this end, we propose a novel Shared Attention Framework for Parameter-efficient conTinual learning (**SAPT**) of large language models. In SAPT, the Shared Attentive Learning & Selection Module (SALS) is devised, where each training sample is navigated to utilize the optimal combinations of existing PET blocks for completing the current task. This is achieved through an attention weight obtained via instance-level shared attention operation. Then inputs in the testing time are capable of following the same shared attention operation to reach the attention weight and pick out the appropriate PET blocks accordingly.

However, continually updating the SALS leads to the optimal attentive combination only for the newest task, resulting in the forgetting for that of previous ones. Thus, we introduce Attentive Reflection Module (ARM) to help SALS recall what the shared attention operation of samples from previous tasks should be originally performed through generative replay. And the success of ARM offers a new perspective for the utilization of generated pseudo samples instead of just blindly mixing them with samples of new tasks for multi-task training.

We conduct extensive experiments to evaluate SAPT on SuperNI (Wang et al., 2022a) and Long Sequence (Razdaibiedina et al., 2023) benchmarks. State-of-the-art performance is achieved by SAPT compared with recent PET-based CL methods. Moreover, SAPT also exhibits superior performance when we scale it to different model sizes (from 770M to 13B), different model architectures, including T5 (Raffel et al., 2020) (encoder-decoder) and LLaMA-2 (Touvron et al., 2023) (decoder-only) and previously unseen tasks.

The main contributions of this work are summarized as follows:

- We propose a novel framework SAPT, including SALS and ARM, to align the PET learning and selection process to effectively handle the CF and KT challenges simultaneously.
- A novel perspective for the utilization of pseudo samples through generative replay is offered in ARM to assist SALS in navigating inputs to their corresponding PET blocks.
- Results of extensive experiments on the benchmark datasets demonstrate the effectiveness of SAPT to mitigate CF and facilitate KT.

2 Related Works

2.1 Parameter-Efficient Tuning

Recently, parameter-efficient tuning (PET) (Ding et al., 2022) has become an appealing research topic which aims at minimizing computational resources when adapting LLMs to specific tasks. Various approaches have emerged in this field, including adapter (Houlsby et al., 2019), prompt-based tuning (Lester et al., 2021; Li and Liang, 2021), BitFit (Zaken et al., 2022) and LoRA (Hu et al., 2021). Since LoRA has exhibited superior performance compared to many mainstream PET methods, our experiments will primarily concentrate on LoRA as a representative method. To ensure a fair comparison with previous prompt-based methods, we also implement a prompt-version of SAPT.

2.2 Continual Learning for LLMs

Conventional Continual Learning (CL) are divided into three categories. (1) *Rehearsal-based methods* introduce the fixed memory to store real samples (Lopez-Paz and Ranzato, 2017; Isele and Cosgun, 2018; Rolnick et al., 2019; de Masson D'Autume et al., 2019) or pseudo-generative

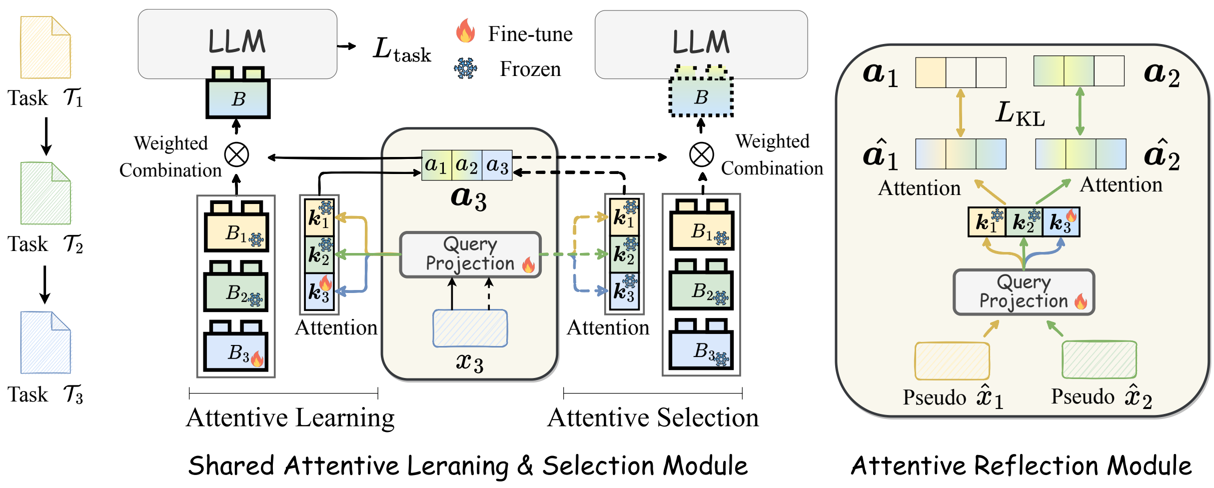


Figure 2: The overall architecture of our proposed SAPT. We assume that SAPT is currently at the time step 3 to learn the task $\mathcal{T}_3$. (1) In the SALS, as illustrated by the solid lines, the resulting attention weight $\boldsymbol{a}_3$ of task $\mathcal{T}_3$ is first obtained via the instance-level shared attention operation between the input x_3 and PET key vectors $\{\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3\}$, to perform weighted combination of all PET blocks $\{B_1, B_2, B_3\}$ for the attentive learning of the current task $\mathcal{T}_3$. And dashed lines display the process of attentive selection, following the same process of shared attention to reach the attention weight $\boldsymbol{a}_3$ and utilizing it to handle given inputs at the testing time. (2) In the ARM, for previous tasks $\mathcal{T}_1$ and $\mathcal{T}_2$, the current attention weights of them ($\hat{\boldsymbol{a}}_1$ and $\hat{\boldsymbol{a}}_2$), are pulled back to their original states ($\boldsymbol{a}_1$ and $\boldsymbol{a}_2$), with the introduction of generated pseudo samples $\hat{x}_1$ and $\hat{x}_2$.

examples (Shin et al., 2017; Sun et al., 2019; Su et al., 2019) of previous tasks. (2) *Regularization-based methods* impose constraints on the loss function to penalize changes regarding the knowledge of previous tasks (Kirkpatrick et al., 2017; Li and Hoiem, 2017; Farajtabar et al., 2020). (3) *Parameter isolation methods* dynamically expand model capacity or isolate existing model weights to mitigate interference between new and old tasks (Rusu et al., 2016; Fernando et al., 2017).

Continual Learning for LLMs with PET. Based on PET methods, current approaches for the CL of LLMs inherit the idea of parameter isolation, exhibiting a pipeline fashion to learn and select PET blocks for each task. However, most of them assume task-ids are available at testing time so that they directly use the oracle PET block of each task and just skip the selection process (Qin and Joty, 2022; Zhang et al., 2022; Qin et al., 2023). These lines of works simplify the problems of CL and could not be applied for real-world application of LLMs where the task-ids are unavailable. Thus, another branches of attempts focus on the more practical settings where the process of PET selection must be involved due to the unavailable task-ids during testing time. However, they are limited in effectively dealing with CF and KT challenges. For the PET learning, Wang et al. (2023b) allocate private prompt for each task and Wang et al. (2023a); Smith et al. (2023) constrain the learning of PET block to keep orthogonal. They restrict the knowledge transfer among different tasks. And simply implementing the PET selection via the summation (Wang et al., 2023a) or concatenation (Razdaibiedina et al., 2023) of all existing PET blocks or select them from a fixed pool (Wang et al., 2022b) make LLMs vulnerable to CF.

Our proposed SAPT stands out from them in that we attempt to align the learning and selection of PET blocks so that CF and KT can be effectively addressed simultaneously.

3 Problem Definition and Setup

Continual learning seeks to address challenges within ongoing sequences. Formally, a sequence of tasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_T\}$ arrive in a streaming fashion. Each task $\mathcal{T}_t = \left\{\left(x_t^i, y_t^i\right)\right\}_{i=1}^{n_t}$ contains a separate target dataset with the size of n_t. At any time step t, the model not only needs to adapt to the t-th task, but also keep performances on all previous tasks.

In this study, we delve into the more challenging and practical settings, addressing: (1) **Diverse task types**: Unlike previous approaches that merely focus on classification problems (Wang et al., 2023a,b), the model would encounter a sequence of tasks encompassing various types, such as dialogue generation, information extraction, etc. (2) **Absence of task identifiers**: During the testing phase, the model confronts samples without knowing which specific task they belong to.

4 Methodology

4.1 Overview of the Framework

We propose SAPT, a novel framework for the CL of LLMs, offering an effective solution to address the challenges of catastrophic forgetting (CF) and knowledge transfer (KT) simultaneously. The overall architecture of SAPT is illustrated in Figure 2, comprising two key components: (1) Shared Attentive Learning & Selection Module (SALS) and (2) Attentive Replay Module (ARM). In SALS, attentive learning (solid lines) and attentive selection (dashed lines) are aligned through the shared attention operation. Then in ARM, we assist SALS in recalling the exact attentions of inputs from previous tasks with generated pseudo samples.

4.2 Shared Attentive Learning & Selection Module

We devise the SALS module to align the learning and selection processes for PET blocks, where challenges of catastrophic forgetting and knowledge transfer could be effectively addressed.

PET Methods. We adopt two representative PET methods, Prompt Tuning (Lester et al., 2021) and LoRA (Hu et al., 2021) in SAPT. The additional trainable parameters introduced by them are referred to as PET blocks. Please refer to Appendix A for more details of the two PET methods.

Attentive Learning. In order to transfer the knowledge acquired from previous tasks, when the t-th task arrives, parameters of all previous PET blocks $\{B_1, B_2, \ldots, B_{t-1}\}$ and the current one B_t are aggregated via weighted combination for the attentive learning of task $\mathcal{T}_t$. Specifically, we allocate a key vector $\boldsymbol{k}_i$ for each PET block B_i ($i \in [1,t]$) and calculate instance-level input-key attentions.[2] Such input-key attention ensures the process of attentive learning to be PET-agnostic and compatible with both prompt tuning and LoRA in SAPT.

The process of shared attention begins when the j-th input of the current t-th task passes through the embedding layer of the LLM backbone to obtain $\boldsymbol{E}_t^j$ (we will omit the superscripts j for simplicity). Since $\boldsymbol{E}_t \in \mathbb{R}^{m\times d}$ and each key vector $\boldsymbol{k}_i \in \mathbb{R}^d$ are of different sequence length, we first perform the max-pool operation on the length dimension of $\boldsymbol{E}_t$, and obtain $\boldsymbol{e}_t \in \mathbb{R}^d$. Then $\boldsymbol{e}_t$ is fed to a sub-network to project it as a query into the spaces of the key vectors for better feature alignment. This consists of down and up projection:

$$\begin{aligned} \boldsymbol{h}_t^{\text{down}} &= \boldsymbol{W}^{\text{down}}(\boldsymbol{e}_t) \\ \boldsymbol{h}_t^{\text{up}} &= \boldsymbol{W}^{\text{up}}(\text{NonLinear}(\boldsymbol{h}_t^{\text{down}})) \\ \boldsymbol{q}_t &= \text{LayerNorm}(\boldsymbol{h}_t^{\text{up}}) \end{aligned} \tag{1}$$

where $\boldsymbol{W}^{\text{down}} \in \mathbb{R}^{d_p\times d}$ and $\boldsymbol{W}^{\text{up}} \in \mathbb{R}^{d\times d_p}$ are learnable projection parameters. Following Asai et al. (2022), we use SiLU (Elfwing et al., 2018) for the non-linear and apply Layer Norm (Ba et al., 2016) on $\boldsymbol{h}_t^{\text{up}}$ to stabilize the learning process.

Then, the attention weights $\boldsymbol{a}_t = \{a_1, a_2, \ldots, a_t\}$ are calculated by the product between $\boldsymbol{q}_t$ and each $\boldsymbol{k}_i$ with softmax:

$$a_i = \frac{e^{\boldsymbol{q}_t \boldsymbol{k}_i / T}}{\sum_{i=1}^{t} e^{\boldsymbol{q}_t \boldsymbol{k}_i / T}} \tag{2}$$

where T is a temperature factor to avoid making the attention weights over-confident and hindering the knowledge transfer. And the parameters of aggregated PET blocks can be obtained:

$$\theta_B = \sum_{i=1}^{t} a_i \, \theta_{B_i} \tag{3}$$

where θ_{B_i} is the parameters of PET block B_i.

The training loss for the attentive learning of the current task $\mathcal{T}_t$ is:

$$L_{\text{task}} = -\sum_{(x_t,y_t)\in\mathcal{T}_t} \log P\left(y_t \mid x_t; \theta_m, \theta_B, \theta_{\text{proj}}, \theta_k\right) \tag{4}$$

where $\theta_m, \theta_B, \theta_{\text{proj}}$ and θ_k are parameters of the LLM backbone, the aggregated PET block, the query projection layer and the set of all key vectors, respectively. And only those parameters belongs to the current t-th task are updated during the training, including θ_{B_t}, θ_{proj} and θ_{k_t}.

Attentive Selection. During the inference phase, when testing data from different tasks arrives, the correct PET blocks are supposed to be automatically selected to execute the corresponding tasks. Within the preceding attentive learning, each sample has already been guided to the optimal combinations of existing PET blocks through shared attention. Thus, the attentive selection process is inherently supposed to follow the same attention operation to pick out the relevant PET blocks for the testing input accordingly. To be more specific, attentive selection involves the same computation process of Equations (1) - (3).

[2] This process is called shared attention because it will be repeated by the following attentive selection.

Shared Attentive Learning & Selection. In summary, the shared attention succeeds to align the attentive learning and selection of PET blocks, leading to the shared attentive learning & selection that is of the same computation process and exhibiting promising insights to deal with the CF and KT challenges simultaneously.

4.3 Attentive Reflection Module

With the sequential training of different tasks, the query projection layer in Equation (1) is continually updated. The introduction of the Attentive Reflection Module ensures that inputs from previous tasks can still correctly perform the corresponding shared attention to identify the combination of PET blocks specific to each of them. To achieve this, we employ generative replay to constrain the projection layer with pseudo-samples. This approach ensures that no real samples are involved, thereby saving the cost associated with maintaining a fixed memory (Sun et al., 2019; Qin and Joty, 2022).

At each time step t, a PET block B_t^{ref} is trained to reconstruct input samples of task $\mathcal{T}_t$. For each sample (input-output pair), only the input part is generated conditioned on an initial token [Gen]. Thus, we have $\left\{B_1^{\text{ref}}, B_2^{\text{ref}}, \ldots, B_t^{\text{ref}}\right\}$ to obtain the generated pseudo-samples $\{G_1, G_2, \ldots, G_t\}$ (generated examples could be found in Appendix E.1).

To assist the query projection layer to reflect or recall the correct shared attention for samples from previous tasks at time step t, every instance $\hat{x}_i$ from G_i is fed to the query projection layer and performs input-key attention operation following Equation (1) - (2) to obtain the current attention weight $\hat{\boldsymbol{a}}_i$. To pull $\hat{\boldsymbol{a}}_i$ to what it should originally be, we minimize a KL divergence loss:

$$L_{\text{KL}} = \sum_{i=1}^{t-1} \sum_{j=1}^{\hat{n}_i} D_{\text{KL}}(\hat{\boldsymbol{a}}_i || \boldsymbol{a}_i) \tag{5}$$

where $\hat{n}_i$ is the number of pseudo samples from $\mathcal{T}_i$. Here, $\boldsymbol{a}_i$ is the average attention weights of the test samples from $\mathcal{T}_i$, representing the overall attention weight of it. Notably, $\boldsymbol{a}_i$ is preserved immediately after the completion of learning $\mathcal{T}_i$, and the position of $(i, t]$ in $\boldsymbol{a}_i$ is padded with 0 when it participates the calculation in Equation (5).

Finally, we jointly minimize the task loss and the KL loss in the multi-task learning fashion:

$$L = L_{\text{task}} + \lambda L_{\text{KL}} \tag{6}$$

where λ is a hyper-parameter that functions to balance the two parts.

5 Experiments

5.1 Dataset and Evaluation Metrics

5.1.1 Dataset

SuperNI Benchmark (Wang et al., 2022a): a benchmark of diverse NLP tasks and their expert-written instructions, enabling rigorous benchmarking of the more practical settings for the CL of LLMs. Specifically, in the types of dialogue generation, information extraction, question answering, summarization, and sentiment analysis, we select three tasks for each type, forming a sequence comprising a total of 15 tasks to evaluate various methods. For each task, 1,000 instances from the dataset are randomly sampled for training and 100 instances for validation and testing.

Long Sequence Benchmark (Razdaibiedina et al., 2023): a continual learning benchmark of 15 classification datasets. Following Razdaibiedina et al. (2023); Wang et al. (2023a), we select 1,000 random samples for training each task and hold out 500 samples per class for validation and testing.

We explore two different task orders for each benchmark. Please refer to Appendix B for more details about the tasks and orders.

5.1.2 Metrics

Let $a_{i,j}$ be the testing performance (Accuracy for classification task and Rouge-L (Lin, 2004) for others) on the i-th task after training on j-th task, the metrics for evaluating are:

(1) **Average Performance (AP)** (Chaudhry et al., 2018). The average performance of all tasks after training on the last task, i.e., $A_{\mathcal{T}} = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} a_{\mathcal{T},t}$;

(2) **Forgetting Rate (F.Ra)** (Chaudhry et al., 2018) measures how much knowledge has been forgotten across the first $\mathcal{T} - 1$ tasks, i.e., $F_{\mathcal{T}} = \frac{1}{\mathcal{T}-1} \sum_{t=1}^{\mathcal{T}-1} (a_{t,t} - a_{\mathcal{T},t})$;

(3) **Forward Transfer (FWT)** (Lopez-Paz and Ranzato, 2017) measures how much knowledge from previous tasks transfers to a new task, i.e., $\text{FWT}_{\mathcal{T}} = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} (a_{t,t} - a_{0,t})$, where $a_{0,t}$ refers to the performance of training task t individually;

(4) **Backward Transfer (BWT)** (Ke and Liu, 2022) measures how much the learning of subsequent tasks influences the performance of a learned task, i.e., $\text{BWT}_{\mathcal{T}} = \frac{1}{\mathcal{T}-1} \sum_{t=1}^{\mathcal{T}-1} (a_{\mathcal{T},t} - a_{t,t})$.

5.2 Baselines and Comparison Models

We evaluate SAPT against the following PET-based continual learning baseline methods: (1) **SeqLoRA**: sequentially trains the LoRA on the task

	SuperNI Benchmark				Long Sequence Benchmark			
	AP↑	F.Ra↓	FWT↑	BWT↑	AP↑	F.Ra↓	FWT↑	BWT↑
SeqLoRA	6.43	33.39	-13.58	-30.94	9.72	78.61	0.81	-73.37
Replay	35.37	16.92	-1.35	-15.79	71.28	13.05	1.28	-12.18
L2P	12.73	11.87	-19.14	-7.95	57.98	22.49	1.36	-16.63
LFPT5	34.37	15.80	-0.46	-14.47	67.01	13.89	2.48	-12.80
ProgPrompt	3.34	35.57	-3.29	-33.18	7.98	71.55	-2.63	-66.71
EPI	-	-	-	-	75.15	1.61	-0.77	-1.42
O-LoRA	25.89	26.37	-0.14	-24.59	69.24	7.00	-8.15	-4.05
SAPT-Prompt	41.11	1.32	**1.95**	-0.65	79.14	1.68	**3.29**	-1.48
SAPT-LoRA	**51.54**	**0.91**	1.88	**-0.57**	**82.02**	**1.50**	1.86	**-1.25**

Table 1: The overall results on two continual learning benchmarks with T5-Large model. Performance of continual learning (AP), forgetting rate (F.Ra), forward transfer (FWT) and backward transfer (BWT) are reported after training on the last task. All results are averaged over two different orders of each benchmark.

orders. (2) **Replay**: replays real samples from old tasks when learning new tasks to avoid forgetting. (3) **L2P** (Wang et al., 2022b): uses the input to dynamically select and update prompts from a fixed prompt pool. (4) **LFPT5** (Qin and Joty, 2022): continuously trains a soft prompt for each task with generative replay and an auxiliary loss. (5) **ProgPrompt** (Razdaibiedina et al., 2023): sequentially concatenates previous learned prompts to the current one during the training and testing time. (6) **EPI** (Wang et al., 2023b): trains prompts for each task and selects them via the distance between the input and distributions formed by labels of different classification tasks. (7) **O-LoRA** (Wang et al., 2023a): learns tasks in different LoRA subspaces that are kept orthogonal to each other and sums all LoRA weights up at testing time.

5.3 Implementation Details

SAPT is a model- and PET-agnostic CL method that is compatible with any transformer-based LLM. In our experiments, all methods are performed with instruction tuning (Wei et al., 2021; Ouyang et al., 2022) to leverage the task instruction provided in the two benchmarks. To ensure a fair comparison with recent works, we implement SAPT with both prompt tuning and LoRA based on the pre-trained encoder-decoder T5-large model (Raffel et al., 2020). We also scale SAPT to the backbone with larger model size (up to 11B and 13B) and the decoder-only LLaMA-2 model (Touvron et al., 2023). For the baselines, since they only report the AP metric in their original papers, we carefully re-implement them with their official codes to report metrics of F.Ra, FWT and BWT, providing a thorough insight of how existing methods deal with CF and KT. For more detailed settings, please refer to the Appendix C.

6 Results and Analysis

6.1 Overall Results

Table 1 demonstrates the performance comparison of SAPT and recent PET-based CL baselines on two benchmarks. All results are averaged over the two different orders of each benchmark. Detailed results of each order and each task within a specific order are provided in Appendix D.

Our SAPT could effectively deal with the challenges of CF and KT simultaneously. Compared to both prompt-based methods (SAPT-Prompt v.s LFPT5/ProgPrompt/EPI) and LoRA-based methods (SAPT-LoRA v.s Replay/O-LoRA), SAPT performs better in addressing the two critical challenges, CF (highest AP and lowest F.Ra) and KT (highest FWT and BWT) when learning different tasks sequentially. Moreover, for the replay-based methods, the better performance of SAPT over Replay and LFPT5 offers a new perspective for the utilization of pseudo samples instead of just blindly mixing them with samples of new tasks for joint training. Please refer to Appendix E.2 for more detailed results and analysis regarding the utilization of replayed samples.

The alignment of learning and selection of PET is better than previous pipeline fashion. SAPT outperforms the state-of-the-art pipeline method, EPI, which verifies the effectiveness of aligning the learning and selection with a shared attention weight. Since EPI is specifically designed for the CL of classification tasks where the selection of PET is based on the label information of each task, it can not be directly applied to the SuperNI benchmark covering various types of tasks other than

	SuperNI Benchmark				Long Sequence Benchmark			
	AP↑	F.Ra↓	FWT↑	BWT↑	AP↑	F.Ra↓	FWT↑	BWT↑
SAPT-LoRA	**51.54**	**0.91**	**1.88**	**-0.57**	**82.02**	**1.50**	1.86	**-1.25**
– ARM	11.12	42.83	0.70	-40.44	10.18	78.45	**1.93**	-73.22
+ Replay	45.41	7.70	1.26	-6.79	76.93	6.86	1.21	-6.41
– Alignment	45.90	2.98	-2.42	-2.55	77.61	2.83	-3.92	-2.48
– DA	44.36	4.16	-2.95	-3.56	67.81	8.24	-8.60	-7.59

Table 2: Results of ablation study on two benchmarks. ARM, Alignment and DA refer to the attentive reflection module, the alignment of the learning and selection in SAPT and shared attentive learning & selection, respectively.

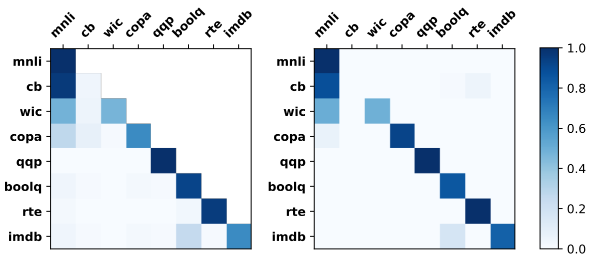


Figure 3: Visualization on shared attention of SAPT-Prompt on the Long Sequence benchmark during the training for each task (left) and testing for all tasks after the training of the last task (right).

classification. This manifests that SAPT is more practical to the real-world applications of LLMs. In addition, the best results of SAPT in terms of AP and F.Ra demonstrate the great potential that such attention-guided soft selection of PET are more resistant to CF, compared with previous methods of concatenation (ProgPrompt), summation (O-LoRA) and top-1 selection (EPI).

6.2 Visualization on Shared Attention

Figure 3 displays the heat maps for shared attention during the training and testing time. We can observe that: (1) the learning and selection processes of PET blocks are exactly aligned that the two heatmaps nearly have the same layout. (2) KT do happens in the attentive learning process to assist SAPT acquire new knowledge. These further verify the effectiveness of SAPT to deal with CF and KT. Please refer to Appendix F for more discussions and visualization results.

6.3 Ablation Study

We conduct ablation studies to verify the effectiveness of different modules proposed in SAPT-LoRA. Results are shown in Table 2.

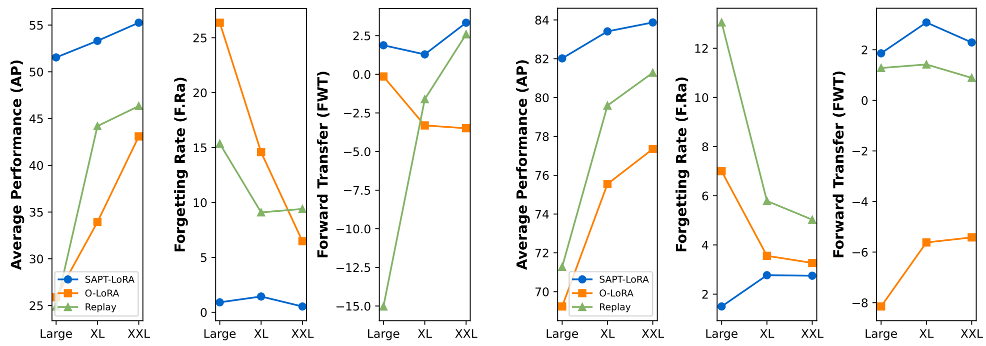
Effect of Attentive Reflection. After removing the attentive reflection module ("– ARM", implemented by discarding the L_{KL}), the significant decline highlights its crucial role in assisting different input samples to recall the correct shared attention for the corresponding PET blocks they should originally combine. When replacing ARM with naive Replay ("+ Replay"), the decline of F.Ra further verifies our claim that ARM offers a more effective solution to apply pseudo samples.

Effect of the Alignment. We transform the alignment of PET learning and selection in SAPT into an independent format. This involves initially performing attentive learning to obtain weights that represent the combination of existing PET blocks. Subsequently, a separate PET selector is trained with these weights and generated pseudo samples. The comprehensive decline in model performance validates our claim that the learning and selection processes of PET are inherently capable of aligning together to collaborate seamlessly.

Effect of Shared Attentive Learning & Selection. Furthermore, we remove the shared attentive mechanism based on the above pipeline settings, where each PET block is learned within a single task and the selector are required to pick the most confident top-1 block for inference. The model's performance has suffered significantly, especially in terms of knowledge transfer. This demonstrates that leveraging acquired knowledge comprehensively, whether in PET learning or selection, is crucial for effectively addressing CF and KT.

6.4 Power of Scale

Scale to larger backbone. We empirically analyze how increasing the backbone T5 size affects the performance of SAPT. Figure 4 displays the performance of SAPT, O-LoRA and Replay in terms of AP, F.Ra and FWT, based on the ascending backbone sizes, Large (770M), XL (3B) and XXL (11B). Overall, with the increased sizes of the backbone model, SAPT could always demonstrate superior performance over baseline models



(a) Results on the SuperNI benchmark (b) Results on the Long Sequence benchmark

Figure 4: Performance of SAPT and baseline methods based on different size of T5-model in terms of performance of continual learning, forgetting rate and forward transfer.

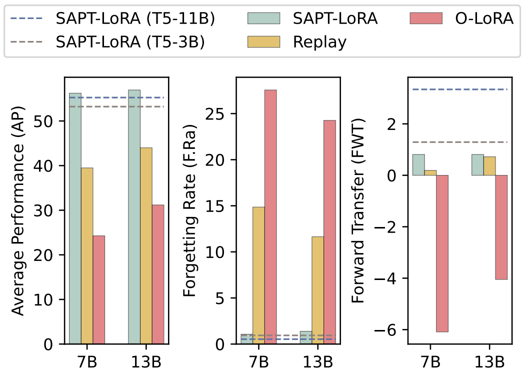


Figure 5: Comparison of SAPT and baselines based on different architectures of LLM backbones, including T5 (encoder-decoder) and LLaMA-2 (decoder-only).

	Unseen Tasks					Avg.
	Dialog	IE	QA	Sum	SA	
T5-ZS	7.49	6.70	4.28	12.14	4.54	7.03
O-LoRA	4.39	9.89	25.38	8.26	50.41	19.67
LFPT5	6.96	**35.32**	35.00	13.26	21.51	22.41
SAPT-LoRA	**11.56**	29.66	**38.04**	**13.77**	**50.62**	**28.73**

Table 3: Results on unseen tasks based on the T5-Large backbone model. We report the average Rouge-L of the 3 tasks under each category.

in resisting catastrophic forgetting and facilitating knowledge transfer. It is worth noting that even with the largest backbone model, O-LoRA (11B) still falls short in terms of Average Performance compared to the smallest version of SAPT-LoRA (770M). This further underscores the crucial importance of selecting the pertinent PET blocks for each input sample in real-world application scenarios.

Scale to different architectures. The results of SAPT and baseline methods on the SuperNI Benchmark based on different sizes of T5 and LLaMA-2 are shown in Figure 5. It can be observed that SAPT is capable of effectively mitigating CF and promoting KT across different model architectures. Moreover, the average performance improves with the enhancement of the model's basic capabilities (LLaMA-2 > T5). This further demonstrates the generality of our proposed SAPT.

Scale to unseen tasks. We further select 3 tasks from each one of the above task category to assess the model's cross-task generalization ability. This is also a crucial dimension for evaluating CL algorithms. Table 3 shows the results. T5-ZS represent the zero-shot approaches for task adaptation, respectively. SAPT yields the best performances, which can be attributed to its superiority in effectively combining acquired knowledge to address novel tasks. This suggests that we should actively promote knowledge transfer between different tasks during the process of CL.

7 Conclusion

In this paper, we propose SAPT, a novel framework for the parameter-efficient continual learning of LLMs. In SAPT, we ingeniously align the two key processes of parameter-efficient block learning and selection through the shared attention, allowing it to effectively alleviate catastrophic forgetting and promote knowledge transfer simultaneously. More importantly, SAPT works under the practical settings where no task-ids are provided for the inputs to select their corresponding parameters. Experimental results also demonstrate the applicability of SAPT across different parameter-efficient tuning methods, models of varying scales and architectures, highlighting its universality.

8 Limitations

There are several limitations to consider for future directions of continual learning of large language models. Firstly, when the learning sequence scales to hundreds of tasks, continually expanding the PET pool to allocate a PET block for each one of them would lead to large computation and storage costs. Thus, how to prune and merge similar PET blocks in the continual learning process can be an interesting direction to explore. Secondly, although SAPT exhibits the best performance of Backward Transfer (BWT), it still fails to allow subsequent tasks to impose the positive impacts on the learned ones. This could be a critical direction to further explore more advanced CL methods. Finally, even though our approach do not depend on identifying task-ids during the testing phase, it still necessitates the identification of tasks during training to establish distinct PET parameters for each task. Investigating techniques for training that is independent of task identification could prove to be a promising avenue for future research, which could favor the application of continual learning upon on the online streams of data.

References

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2021. Building and evaluating open-domain dialogue corpora with clarifying questions. In *EMNLP*.

Akari Asai, Mohammadreza Salehi, Matthew E Peters, and Hannaneh Hajishirzi. 2022. Attempt: Parameter-efficient multi-task tuning via attentional mixtures of soft prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6655–6672.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547.

Hyundong Cho and Jonathan May. 2020. Grounding conversations with improvised dialogues. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2398–2413.

Pradeep Dasigi, Nelson F Liu, Ana Marasović, Noah A Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932.

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. 2020. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A Rusu, Alexander Pritzel, and Daan Wierstra. 2017. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *arXiv preprint arXiv:1911.12237*.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The third dialog state tracking challenge. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 324–329. IEEE.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

David Isele and Akansel Cosgun. 2018. Selective experience replay for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Zixuan Ke and Bing Liu. 2022. Continual learning of natural language processing tasks: A survey. *arXiv preprint arXiv:2211.12701*.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of reddit posts with multi-level memory networks. In *Proceedings of NAACL-HLT*, pages 2519–2531.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *CoNLL 2017*, page 333.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.

Lalita Lowphansirikul, Charin Polpanumas, Attapol T Rutherford, and Sarana Nutanong. 2020. scb-mt-en-th-2020: A large english-thai parallel corpus. *arXiv preprint arXiv:2007.03541*.

Yun Luo, Zhen Yang, Xuefeng Bai, Fandong Meng, Jie Zhou, and Yue Zhang. 2023. Investigating forgetting in pre-trained representations through continual learning. *arXiv preprint arXiv:2305.05968*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CONLL)*.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *EMNLP*.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. Glucose: Generalized and contextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain Marshall, Ani Nenkova, and Byron C Wallace. 2018. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 197–207.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Denis Peskov, Benny Cheng, Ahmed Elgohary, Joe Barrow, Cristian Danescu-Niculescu-Mizil, and Jordan Boyd-Graber. 2020. It takes two to lie: One to lie, and one to listen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3811–3854.

Chengwei Qin, Chen Chen, and Shafiq Joty. 2023. Lifelong sequence generation with dynamic module expansion and adaptation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6701–6714.

Chengwei Qin and Shafiq Joty. 2022. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. In *International Conference on Learning Representations*.

Dragomir Radev, Rui Zhang, Amrit Rau, Abhinand Sivaprasad, Chiachun Hsieh, Nazneen Fatema Rajani, Xiangru Tang, Aadit Vyas, Neha Verma, Pranav Krishna, Yangxiaokang Liu, Nadia Irwanto, Jessica Pan, Faiaz Rahman, Ahmad Zaidi, Murori Mutuma, Yasin Tarabar, Ankit Gupta, Tao Yu, Yi Chern Tan, Xi Victoria Lin, Caiming Xiong, and Richard Socher. 2020. Dart: Open-domain structured data record to text generation. *arXiv preprint arXiv:2007.02871*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, and Amjad Almahairi. 2023. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations*.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. Evalution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 64–69.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3687–3697.

Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. 2023. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919.

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.

Xin Su, Shangqi Guo, Tian Tan, and Feng Chen. 2019. Generative memory for lifelong learning. *IEEE transactions on neural networks and learning systems*, 31(6):1884–1898.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2019. Lamol: Language modeling for lifelong language learning. In *International Conference on Learning Representations*.

Shahbaz Syed, Roxanne El Baff, Johannes Kiesel, Khalid Al Khatib, Benno Stein, and Martin Potthast. 2020. News editorials: Towards summarizing long argumentative texts. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5384–5396.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 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 4149–4158.

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. 2023a. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022a. Supernaturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Zhicheng Wang, Yufang Liu, Tao Ji, Xiaoling Wang, Yuanbin Wu, Congcong Jiang, Ye Chao, Zhencong Han, Ling Wang, Xu Shao, et al. 2023b. Rehearsal-free continual language learning via efficient parameter isolation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10933–10946.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022b. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. 2018. Airdialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3844–3854.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117.

Yuexiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. Investigating the catastrophic forgetting in multimodal large language models. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018b. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2022. Continual sequence generation with adaptive compositional modules. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3653–3667.

Markus Zlabinger, Marta Sabou, Sebastian Hofstätter, and Allan Hanbury. 2020. Effective crowd-annotation of participants, interventions, and outcomes in the text of clinical trial reports. In *Conference on Empirical Methods in Natural Language Processing (EMNLP-Findings 2020)*.

A Parameter-Efficient Tuning Methods

We adopt two representative PET methods, Prompt Tuning (Lester et al., 2021) and LoRA (Hu et al., 2021) in our proposed SAPT, which are referred to as PET blocks in this study.

In prompt tuning, a series of virtual tokens, called soft prompt P is prepended to the input text x, while keeping the LLM parameters frozen. In this case, during the training on the downstream tasks, gradient updates are preformed on the prompt parameters independently.

In LoRA, the pre-trained weight matrix of LLMs is updated with a low-rank decomposition. For a linear layer $h = W_0 x$, the forward pass with LoRA is modified to be:

$$h = W_0 x + BAx \tag{7}$$

where $W_0 \in \mathbb{R}^{d\times k}$, $B \in \mathbb{R}^{d\times r}$, $A \in \mathbb{R}^{r\times k}$, with the rank $r \ll \min(d, k)$. The pre-trained weight matrix W_0 remains fixed during training, while A and B contain trainable parameters.

B Dataset Details

B.1 Datasets

Table 4 & 5 show details of the datasets we used for our experiments, along with their evaluation metrics. Overall, in SuperNI, we choose 3 tasks from dialogue generation (Dialog) (Zhang et al., 2018a; Zang et al., 2020; Peskov et al., 2020), information extraction (IE) (Santus et al., 2015; Nye et al., 2018; Mostafazadeh et al., 2020), question answering (QA) (Dasigi et al., 2019; Talmor et al., 2019), summarization (Sum) (Narayan et al., 2018; Gliwa et al., 2019; Kim et al., 2019) and sentiment analysis (SA) (Socher et al., 2013; Saravia et al., 2018), respectively.

For the Long Sequence benchmark, this includes five tasks from the standard CL benchmark (AG News, Amazon reviews, Yelp reviews, DBpedia and Yahoo Answers) (Zhang et al., 2015), four from GLUE benchmark (MNLI, QQP, RTE, SST2) (Wang et al., 2018), five from SuperGLUE benchmark (WiC, CB, COPA, MultiRC, BoolQ) (Wang et al., 2019), and the IMDB movie reviews dataset (Maas et al., 2011).

And unseen tasks from the SuperNI benchmark are displayed Table 6. They also from the five categories of Dialog (Wei et al., 2018; Cho and May, 2020; Aliannejadi et al., 2021), IE (Mausam et al., 2012; Zlabinger et al., 2020; Radev et al., 2020), QA (Levy et al., 2017; Zhang et al., 2018b; Min et al., 2020), Sum (Henderson et al., 2014; Syed et al., 2020; Hasan et al., 2021) and SA (Sheng and Uthus, 2020; Lowphansirikul et al., 2020).

B.2 Task Sequence Orders

We report 4 different task orders used for our experiments in Table 7.

C Implementation Details

Our experiments are implemented with PyTorch (Paszke et al., 2019) and Transformer library (Wolf et al., 2020). The T5-Large is trained on a single NVIDIA Tesla A800 GPU and the larger backbones T5-XL, T5-XXL, LLaMA-2-7B and LLaMA-2-13B are performed on 4 NVIDIA Tesla A800 using DeepSpeed repository.

For our prompt-based methods, the length of prompts is set to 10. Following Lester et al. (2021), they are initialized from sampled vocabulary of the backbone model and trained using the Adafactor optimizer. On the SuperNI benchmark, we train SAPT-Prompt with 100 epochs, the constant learning rate of 3e-2 and the batchsize of 32 per GPU. As for the hyper-parameter λ in Equation (6), it functions to balance the share attention in the process of attentive learning for the newest task and that in the process of attentive reflection for previous tasks. The larger λ means that the attentive reflection contributes more to assist SALS in recalling the shared attention of previous tasks. However, excessive λ can impair attentive learning for the current task, thereby weakening knowledge transfer. Here, λ is set to 1, which is the relatively optimal balance of the attentive learning and reflection. The hidden dimension d_p of the query projection layer is 100. On the Long Sequence benchmark, the model is trained for 10 epochs with a hierarchical learning rate, 3e-1 for prompts and 1e-2 for the query projection layer. We always keep the total batchsize to 32. And the λ and d_p for order3 and order4 is (1.5, 200) and (1.3, 150), respectively. The attention temperature in Equation (2) is $d \times exp(1)$, where d is the LLM backbone dimension size.

For our LoRA-based methods, we use AdamW optimizer to train the model with the learning rate of 3e-4 for T5-Large, 1e-4 for those larger T5-XL and T5-XXL models, 5e-5 for LLaMA-2-7B and 1e-5 for LLaMA-2-13B. For T5 series, the batch size is set to 32 per GPU. On the SuperNI benchmark, the low rank r, λ and d_p are 4, 0.5 and 100, while they are set to 8, 0.1 and 100 for the Long Sequence benchmark. For LLaMA-2 family, and the batch size is 32 in total. The low rank r, λ and d_p are both 4, 2 and 100 for the Superni and Long Sequence benchmarks. The attention temperature in Equation (2) is $sqrt(d)$, where d is the LLM backbone dimension size.

For the generative replay to obtain pseudo samples for our Attentive Replay Module, the prompt length is 300 and is trained for 80 epochs utilizing Adafactor with learning rate of 0.5. And in LoRA, the low-rank r is 8. We train it with AdamW with the learning rate of 0.001 for 5k steps. Batch size is set to 16 for both methods.

Further, we carefully evaluate the official implementations of all baselines, in order to make the comparison as fair as possible. We strictly follow the hyper-parameter settings in their original code, where the prompt size is all set to 10 (except that for LFPT5 of 300) and the LoRA rank is set to 8. If this could not reach the expected performance, we carry out the hyper-parameter search of the learning rate and bachsize for them. Following Sun et al. (2019); Qin and Joty (2022), the volume of replay samples is 0.02 of the original training set for SAPT and all replay baseline methods (Replay and LFPT5). Please refer to Appendix E.2 for deeper analysis for the volume of pseudo samples. All the methods are evaluated for 3 random runs.

D Fine-grained Results for the Main Experiments

We report the results of each task order on the two benchmark in Table 8 and Table 9. And results of the average performance at each time step is displayed in Figure 8. Overall, the our proposed SAPT demonstrates excellent capabilities in addressing CF and KT.

E More Results and Analysis on Generated Pseudo-Samples

E.1 Examples of Pseudo Samples

Table 10 shows several pseudo samples generated by SAPT for the SuperNI an Long Sequence Benchmark. Since there are tasks instructions in these two benchmarks, the input-output format of real samples is consists of three elements: [INS] task instruction, [IN] task input and [OUT] task output. And we only generate the input part, [INS] and [IN], to perform attentive reflection in SAPT, which is a novel ways of pseudo-samples usage and greatly different from previous works where complete pseudo samples are generated and mixed with the current task data for multi-task learning. We can see that SAPT can indeed generate high-quality pseudo samples to assist samples from previous tasks in correctly identify the combination of PET blocks specific to each of them.

E.2 Different Volumes and Types of Replayed Samples

In SAPT, the Attentive Reflection Module (ARM) provides a novel perspective for utilizing generated pseudo-data. We conduct additional experiments to analyze the impact of using varying scales of pseudo-data and real data on SAPT and the baseline models Replay and LFPT5. The results are shown in Figure 6. We have the following two observations: (1) Regardless of whether real data or pseudo-data is used, SAPT demonstrates computational efficiency during replay, showing superior performance even with the minimum replay scale [0.02] compared to the maximum replay scale [1] of LFPT5 and Replay. It is worth mentioning that when the replay data volume of Replay is 1, it corresponds to the setting of multi-task learning, which is commonly considered as the upper bound of continual learning. SAPT is able to surpass this upper bound, demonstrating its ability to flexibly handle different inputs, enabling them to be processed by corresponding parameters. (2) For SAPT, there is no significant difference in performance between using real data and pseudo-data. This firstly indicates the reliability of the pseudo-data we generated and the sufficient robustness of our proposed ARM, which can utilize replay data of different qualities to accomplish reflection on shared attention.

F Visualization on Shared Attention

We demonstrate the visualization on shared attention operation of SAPT-Prompt on the SuperNI (Figure 9) and the Long Sequence (Figure 10) Benchmark, and the SAPT-LoRA on the SuperNI (Figure 11) and the Long Sequence (Figure 12) Benchmark. And the resulting attention weights is obtained through the average attention weights of the testing samples from a specific task.

Overall, whether based on Prompt or LoRA, SAPT can maintain the alignment for the learning and selection process of PET blocks through shared attention on both benchmarks. Even as the task sequences become longer, it does not affect

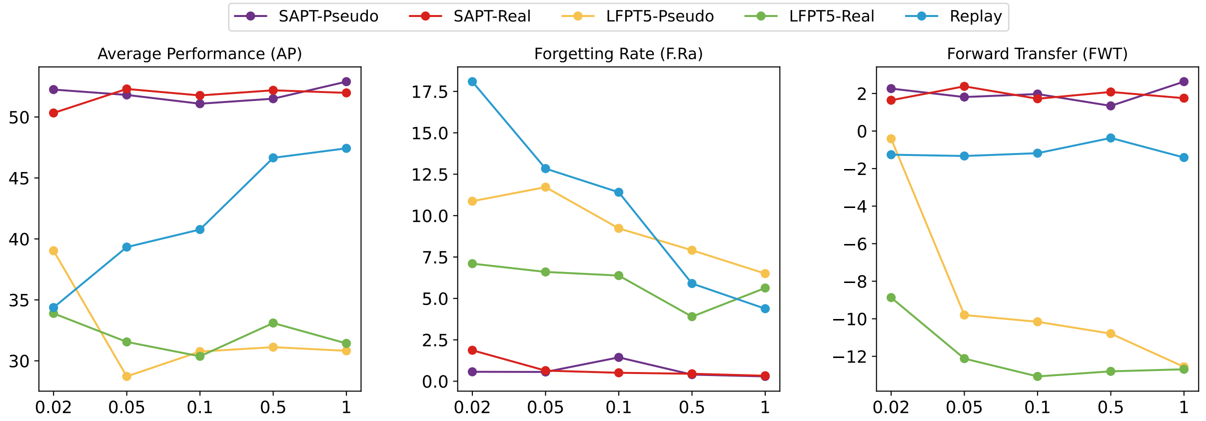


Figure 6: Comparison of SAPT-LoRA and baselines based on different types (real and pseudo) and volumes of replayed data, in terms of Average Performance (AP), Forgetting Rate (F.Ra) and Forward Transfer (FWT).

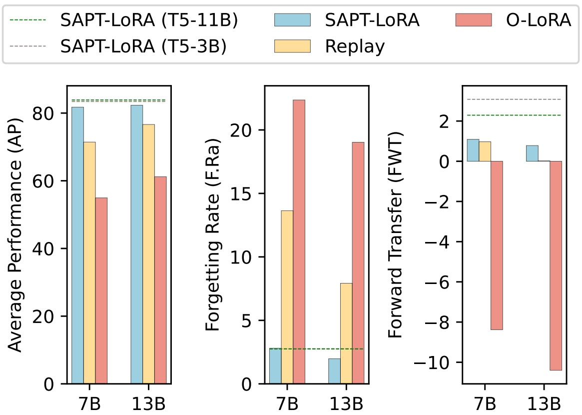


Figure 7: Comparison of SAPT and baselines based on different architectures of LLM backbones on the Long Sequence benchmark, including T5 (encoder-decoder) and LLaMA-2 (decoder-only).

the ability to identify suitable combinations of PET modules. This directly demonstrates its effectiveness in addressing CF and KT.

Furthermore, both methods demonstrate varying degrees of knowledge transfer on the two benchmarks. Overall, the PET blocks in the current task contribute more significantly, as indicated by the darkest color of the diagonal elements. However, there are also interesting observations where the PET blocks for other tasks have weights higher than the current task, surpassing the higher similarity between these tasks (yelp & amazon, mnli & cb). Additionally, the knowledge transfer of Prompt appears slightly more pronounced than LoRA, but overall, LoRA outperforms Prompt in terms of the overall performance. This may be attributed to LoRA's superior representation and learning of task-specific knowledge in the low-rank space, aligning with the conclusions in previous works (Hu et al., 2021; Ding et al., 2022).

G Scale to LLaMA-2 Model

The results of SAPT and baseline methods on the Long Sequence Benchmark based on different sizes of T5 and LLaMA-2 are shown in Figure 7. It can be observed that our proposed SAPT still exhibits superiority to effectively mitigating CF and promoting KT over baseline methods.

Dataset name	Task	Metric
1. task639_multi_woz_user_utterance_generation	dialogue generation	Rouge-L
2. task1590_diplomacy_text_generation	dialogue generation	Rouge-L
3. task1729_personachat_generate_next	dialogue generation	Rouge-L
4. task181_outcome_extraction	information extraction	Rouge-L
5. task748_glucose_reverse_cause_event_detection	information extraction	Rouge-L
6. task1510_evalution_relation_extraction	information extraction	Rouge-L
7. task002_quoref_answer_generation	question answering	Rouge-L
8. task073_commonsenseqa_answer_generation	question answering	Rouge-L
9. task591_sciq_answer_generation	question answering	Rouge-L
10. task511_reddit_tifu_long_text_summarization	summarization	Rouge-L
11. task1290_xsum_summarization	summarization	Rouge-L
12. task1572_samsum_summary	summarization	Rouge-L
13. task363_sst2_polarity_classification	sentiment analysis	accuracy
14. task875_emotion_classification	sentiment analysis	accuracy
15. task1687_sentiment140_classification	sentiment analysis	accuracy

Table 4: The details of 15 datasets in the SuperNI Benchmark (Wang et al., 2022a).

Dataset name	Category	Task	Domain	Metric
1. Yelp	CL Benchmark	sentiment analysis	Yelp reviews	accuracy
2. Amazon	CL Benchmark	sentiment analysis	Amazon reviews	accuracy
3. DBpedia	CL Benchmark	topic classification	Wikipedia	accuracy
4. Yahoo	CL Benchmark	topic classification	Yahoo Q&A	accuracy
5. AG News	CL Benchmark	topic classification	news	accuracy
6. MNLI	GLUE	natural language inference	various	accuracy
7. QQP	GLUE	paragraph detection	Quora	accuracy
8. RTE	GLUE	natural language inference	news, Wikipedia	accuracy
9. SST-2	GLUE	sentiment analysis	movie reviews	accuracy
10. WiC	SuperGLUE	word sense disambiguation	lexical databases	accuracy
11. CB	SuperGLUE	natural language inference	various	accuracy
12. COPA	SuperGLUE	question and answering	blogs, encyclopedia	accuracy
13. BoolQA	SuperGLUE	boolean question and answering	Wikipedia	accuracy
14. MultiRC	SuperGLUE	question and answering	various	accuracy
15. IMDB	SuperGLUE	sentiment analysis	movie reviews	accuracy

Table 5: The details of 15 classification datasets in the Long Sequence Benchmark (Razdaibiedina et al., 2023). First five tasks correspond to the standard CL benchmark (Zhang et al., 2015).

Dataset name	Task	Metric
1. task360_spolin_yesand_response_generation	dialogue generation	Rouge-L
2. task574_air_dialogue_sentence_generation	dialogue generation	Rouge-L
3. task1714_convai3_sentence_generation	dialogue generation	Rouge-L
4. task180_intervention_extraction	information extraction	Rouge-L
5. task678_ollie_actual_relationship_answer_generation	information extraction	Rouge-L
6. task1410_dart_relationship_extraction	information extraction	Rouge-L
7. task339_record_answer_generation	question answering	Rouge-L
8. task670_ambigqa_question_generation	question answering	Rouge-L
9. task1327_qa_zre_answer_generation_from_question	question answering	Rouge-L
10. task522_news_editorial_summary	summarization	Rouge-L
11. task1356_xlsum_title_generation	summarization	Rouge-L
12. task1499_dstc3_summarization	summarization	Rouge-L
13. task421_persent_sentence_sentiment_classification	sentiment analysis	accuracy
14. task833_poem_sentiment_classification	sentiment analysis	accuracy
15. task929_products_reviews_classification	sentiment analysis	accuracy

Table 6: The details of unseen tasks from the SuperNI benchmark.

Order	Model	Task Sequence
1	T5, LLaMA-2	task1572 → task363 → task1290 → task181 → task002 → task1510 → task639 → task1729 → task073 → task1590 → task748 → task511 → task591 → task1687 → task875
2	T5, LLaMA-2	task748 → task073 → task1590 → task639 → task1572 → task1687 → task591 → task363 → task1510 → task1729 → task181 → task511 → task002 → task1290 → task875
3	T5, LLaMA-2	mnli → cb → wic → copa → qqp → boolqa → rte → imdb → yelp → amazon → sst-2 → dbpedia → ag → multirc → yahoo
4	T5, LLaMA-2	yelp → amazon → mnli → cb → copa → qqp → rte → imdb → sst-2 → dbpedia → ag → yahoo → multirc → boolqa → wic

Table 7: Four different orders of task sequences used for our experiments. Orders 1-2 correspond to the SuperNI benchmark. Orders 3-4 are long-sequence orders following Razdaibiedina et al. (2023).

	Order 1				**Order 2**			
	AP↑	**F.Ra↓**	**FWT↑**	**BWT↑**	**AP↑**	**F.Ra↓**	**FWT↑**	**BWT↑**
SeqLoRA	5.05	30.94	-17.01	-28.88	7.80	35.84	-10.15	-32.99
Replay	34.37	18.09	-1.26	-16.89	36.37	15.74	-1.44	-14.69
L2P	15.18	6.23	-20.97	-3.65	10.27	17.51	-17.30	-12.24
LFPT5	39.03	10.87	-0.41	-9.85	29.70	20.72	-0.51	-19.08
ProgPrompt	2.83	35.65	-3.70	-33.27	3.85	35.48	-2.87	-33.09
EPI	-	-	-	-	-	-	-	-
O-LoRA	20.95	30.91	-0.43	-28.83	30.82	21.83	0.15	-20.35
SAPT-Prompt	41.88	1.41	**2.83**	-0.75	40.34	**1.23**	1.07	**-0.54**
SAPT-LoRA	**52.25**	**0.57**	2.26	**-0.23**	**50.82**	1.24	**1.50**	-0.90

Table 8: The overall results on each task order of the SuperNI benchmark with T5-Large model. Performance of continual learning (AP), forgetting rate (F.Ra), forward transfer (FWT) and backward transfer (BWT) are reported after training on the last task.

	Order 3				Order 4			
	AP↑	**F.Ra↓**	**FWT↑**	**BWT↑**	**AP↑**	**F.Ra↓**	**FWT↑**	**BWT↑**
SeqLoRA	6.71	82.07	1.19	-76.60	12.73	75.15	0.43	-70.14
Replay	68.20	16.21	1.20	-15.13	74.25	9.89	1.36	-9.23
L2P	58.61	21.55	1.01	-15.43	57.34	23.42	1.70	-17.82
LFPT5	66.62	14.57	2.89	-13.60	67.40	13.20	2.06	-11.99
ProgPrompt	6.14	74.64	-1.65	-69.53	9.83	68.45	-3.61	-63.89
EPI	75.19	0.77	-1.54	**-0.60**	75.10	2.44	0.00	-2.23
O-LoRA	69.22	8.30	-7.79	-4.42	69.26	5.70	-8.51	-5.09
SAPT-Prompt	80.20	0.91	**3.63**	-0.76	78.08	2.45	**2.95**	-2.20
SAPT-LoRA	**83.44**	**0.75**	1.99	-0.66	**80.60**	**2.25**	1.72	**-1.94**

Table 9: The overall results on each task order of the Long Sequence benchmark with T5-Large model. Performance of continual learning (AP), forgetting rate (F.Ra), forward transfer (FWT) and backward transfer (BWT) are reported after training on the last task.

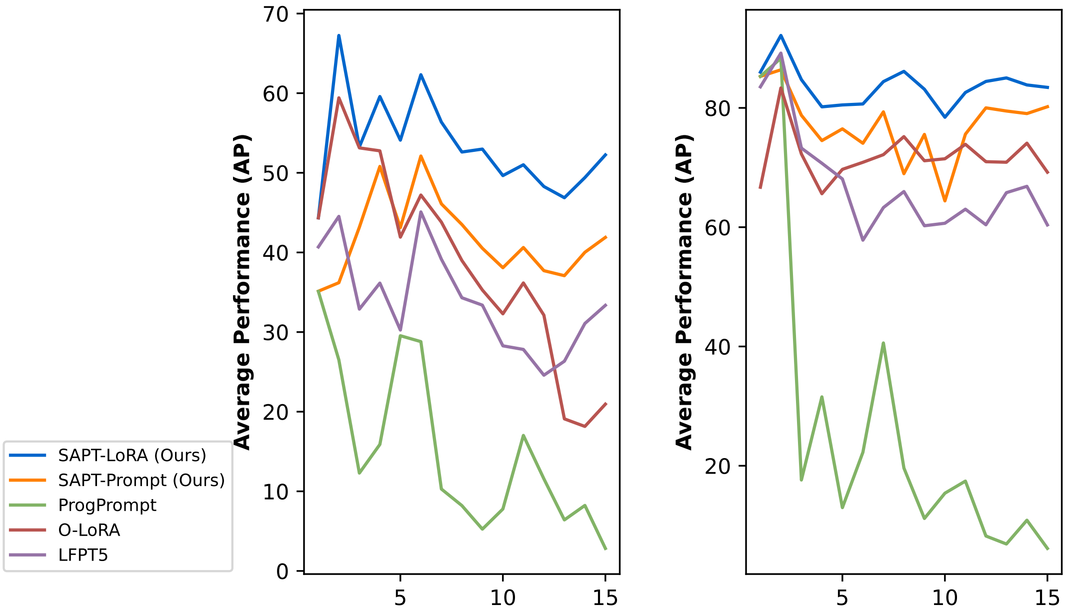


Figure 8: The average performance of SAPT and baseline models at each time step on the SuperNI (left) and the Long Sequence (right) benchmark.

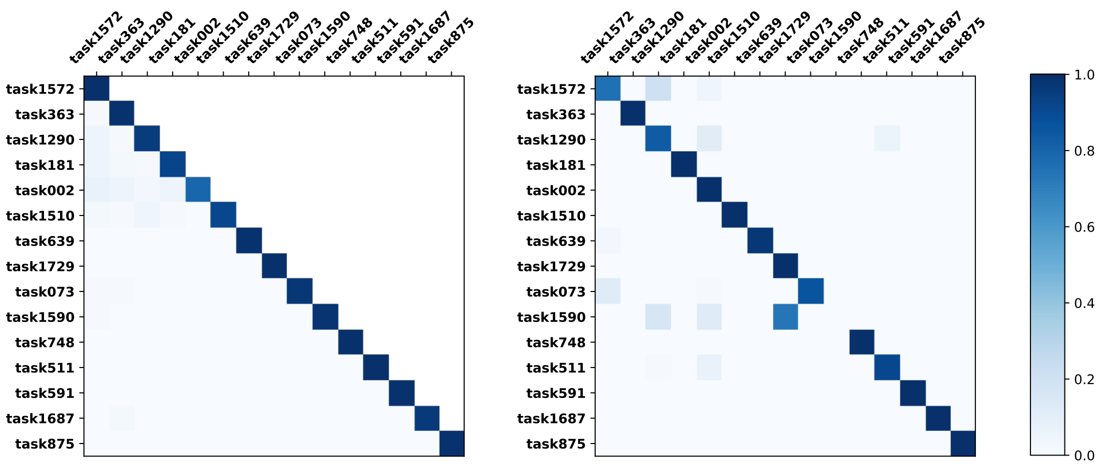


Figure 9: Visualization on shared attention of SAPT-Prompt on the SuperNI benchmark during the training (left) and testing time (right).

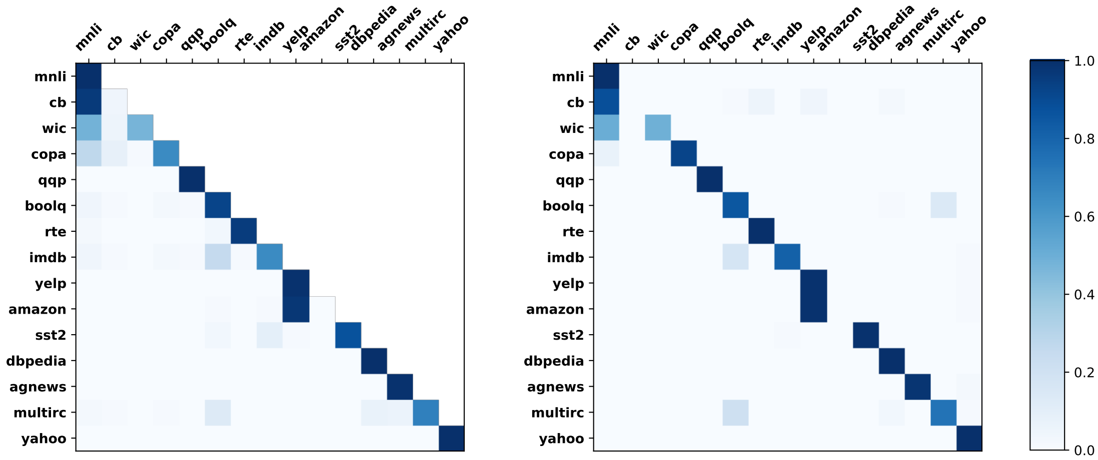


Figure 10: Visualization on shared attention of SAPT-Prompt on the Long Sequence benchmark during the training (left) and testing time (right).

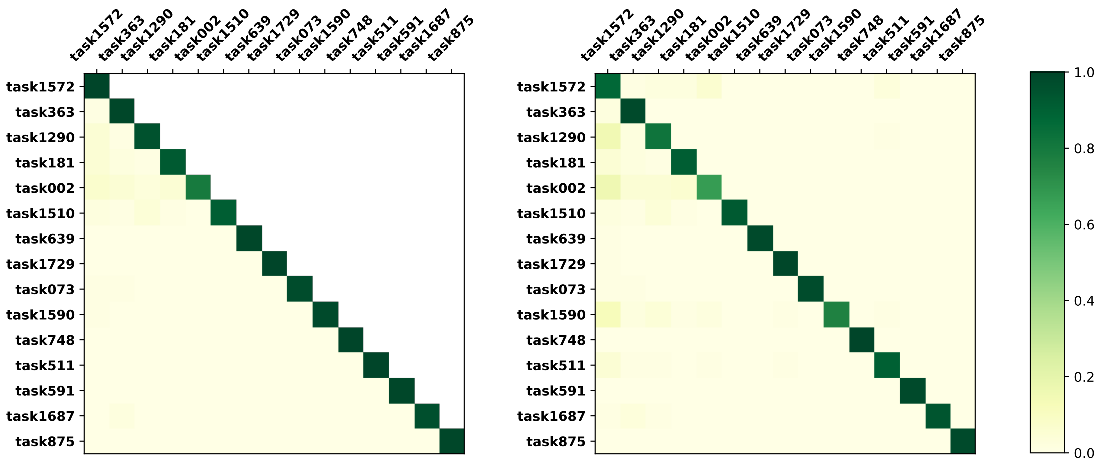


Figure 11: Visualization on shared attention of SAPT-LoRA on the SuperNI benchmark during the training (left) and testing time (right).

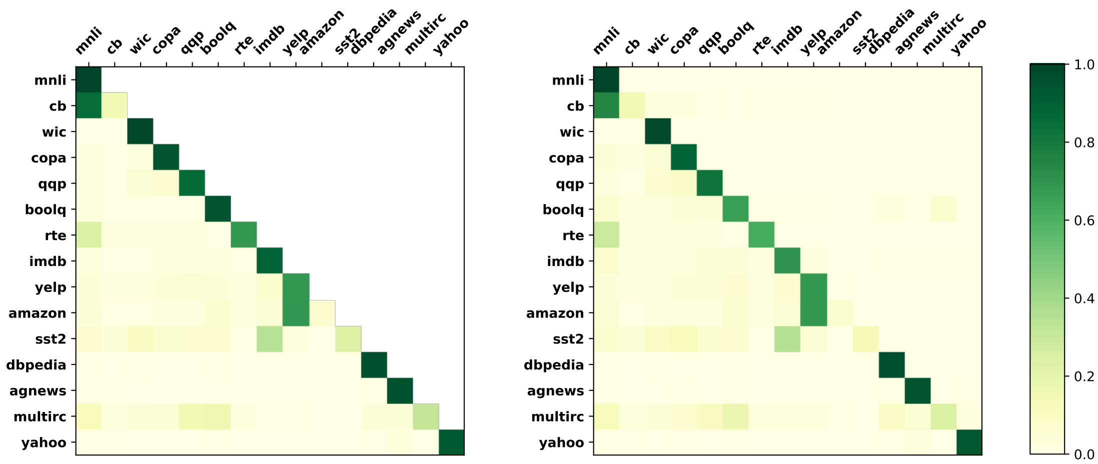


Figure 12: Visualization on shared attention of SAPT-LoRA on the Long Sequence benchmark during the training (left) and testing time (right).

Benchmark	Task Name	Type	Data Sample
SuperNI	task002_quoref_answer_generation	Real	[INS] In this task, you're expected to write answers to questions involving multiple references to the same entity. The answer to the question should be unambiguous and a phrase in the paragraph. Most questions can have only one correct answer. [IN] Passage: Phaedra is a poor Greek sponge diver on the island of Hydra. She works from the boat of her boyfriend, Rhif, an illegal immigrant from Albania. She accidentally finds an ancient Greek statue of a boy riding a dolphin on the bottom of the Aegean Sea. Her efforts to sell it to the highest bidder lead her to two competing individuals: Dr. James Calder, an honest archaeologist who will surrender it to Greek authorities, and Victor Parmalee, an aesthete and an unscrupulous dealer with a history of trying to acquire works of art stolen by the Nazis from their owners.\nCalder and Parmalee each try to win Phaedra's cooperation. She works in concert with Parmalee, while developing feelings for Calder. When she seems to waver, Rhif decides to make the deal with Parmalee work. The film reaches a happy conclusion, with virtue rewarded, the statue celebrated by the people of Hydra, and Phaedra and Calder in each other's arms. Parmalee, a man with no apparent national loyalties or heritage, sets course for Monte Carlo. \nQuestion: Who is dating an illegal immigrant? [OUT] Phaedra.
		Pesudo	[INS] In this task, you're expected to write answers for questions involving multiple references to the same entity. The answer to the question should be unambiguous and a phrase in the paragraph. Most questions can have only one correct answer. [IN] Passage: In 1899 British President Johnston (now known as John Marshall) is charged with running riots against officers of the Royal Navy. After two days of debriefing, Admiral Sir William Rogers Jr. becomes interested in Robert R̊obJ̊ohnson, another officer on staff, who finds out that Johnson was not present at the rehearsals for the film; however, his father also claims that he has been fired from the navy because Mr Williams had been shot before him. He then went off to see an officer looking after the military. He later admitted himself guilty—for failing which he became involved in the crime? Although no arrest was made during his service they served less than 40 days. He does remain behind but never gets away or die first. He takes control over
	task875_emotion_classification	Real	[INS] In this task, you are given a sentence containing a particular emotion. You must classify the sentence into one of the six emotions: 'joy', 'love', 'anger', 'fear', or 'surprise'. [IN] i could clearly feel my adomen muscles contract everytime i cough like some adomen exercise haha and im aching from it now sigh [OUT] sadness
		Pesudo	[INS] In this task, you are given a sentence containing monetary hardship. You must classify the sentence into one of the six emotions: 'joy', 'love', 'anger', 'fear', or'surprise'. [IN] i feel like it is worthwhile to support local artists and so does clay so im amazed by how supportive and passionate artistic writers can be
	task1572_samsum_summary	Real	[INS] In this task, you are given a conversation, and your task is to generate a summary from the information present in the given conversation. Generate a summary in such a way that the context should be present in the conversation. It should cover the complete context of the conversation. [IN] Mary: Can you pick up the kids?, John: Sure. You've already asked me this morning., Mary: Really? I don't remember. , John: Ok, doesn't matter. [OUT] John will pick up the kids at Mary's request.
		Pesudo	[INS] In this task, you are given a conversation, and your task is to generate a summary from the information present in the given conversation. Generate a summary in such a way that the context should be present in the conversation. It should cover the complete context of the conversation. [IN] Leda: Ok, so first Homo Faber., Gaby: I've already booked the tickets., Ivan: Where exactly?, Leda: Isola San Giorgio., Gaby: They have limited spaces., Ivan: They are always full, but we can ask around it., Gaby: So for us? We won't make it all day., Marc: OK, what about Giardini or Arsenale? I think we won't make it to see both. We can ask around., Gaby: I'll go back!, Ivan: Me too. We could also go different places... There are a lot of satellitanni: What about the transportation? Same ticket
Long Sequence	amazon	Real	[INS] Given a review text from an e-commerce platform, please analyze the sentiment polarity expressed in the text. Choose one from the following sentiment categories: very negative, negative, neutral, positive and very positive.\n[IN] Title: Great\nText: Jim Cramer is the best - his stock tips and philosophy of investing is truly outstanding.Thanks Jim for all of your insight!\n[OUT] positive
		Pesudo	[INS] Given a review text from an e-commerce platform, please analyze the sentiment polarity expressed in the text. Choose one from the following sentiment categories: very negative, negative, neutral, positive and very positive! [IN] Title: Just not as cool as I thought it would be Text: This was just too cool to give up on this movie! Unfortunately, there are way more people with bad taste than Meg McConnell here or even Amy Schumer who make some of her own movies instead of being a good mommy.
	imdb	Real	[INS] Classify the sentiment polarity expressed in the following movie reviews.\n [IN] The Japanese Run Lola Run,his is one offbeat movie which will put a smile on just about anyone's face. Fans of Run Lola Run, Tampopo, Go!, and Slacker will probably like this one. It does tend to follow a formula that is increasingly popular these days of separate, seemingly unrelated vignettes, all contributing the the overall story in unexpected ways. catch it if you see it, otherwise wait for the rental. [OUT] Good
		Pesudo	[INS] Classify the sentiment polarity expressed in this movie reviews. [IN] Well, I've just got back from watching this film for my birthday and it's been an incredible experience to see how well done that film is! The production values are unbelievable; not only does it look really good, but they have amazing acting: John Wayne makes one of the best actors on his career, and when he plays Jack Ryan (Scott Stierlin) there can be very few performances you would want him to lose weight because he is over 50 pounds. It should all appeal to everybody. There must also explain why such great cinema was important at the time so many years ago. This picture is another highlight which shows how bad James Bond could be like with anything else. We know what we were facing before leaving our country after 24 hours or 48 days...and now we have had some real talent in this film!!
	yahoo	Real	[INS] I will give you a pair of question and answer, please categorize the topic discussed within.\n [IN] Question: who is the actress from india at da vinci premiere?\nAnswer: Aishwarya Rai, the former Miss World.\n [OUT] Entertainment & Music
		Pesudo	[INS] I will give you a pair of question and answer, please categorize the topic discussed within. [IN] Question: what is the word "butterfly"? Answer: It means butterfly-like flower that grow in a basket or vase with lilies on it

Table 10: Examples of generated pseudo samples of the SuperNI and the Long Sequence benchmarks. [INS], [IN] and [OUT] represent the task instruction, task input and task output, respectively.