

# Mitigate Negative Transfer with Similarity Heuristic Lifelong Prompt Tuning

Chenyuan Wu<sup>1</sup>, Gangwei Jiang<sup>1</sup>, Defu Lian<sup>1\*</sup>

<sup>1</sup>University of Science and Technology of China

{wuchenyuan, gwjiang}@mail.ustc.edu.cn, liandefu@ustc.edu.cn

## Abstract

Lifelong prompt tuning has significantly advanced parameter-efficient lifelong learning with its efficiency and minimal storage demands on various tasks. Our empirical studies, however, highlights certain transferability constraints in the current methodologies: a universal algorithm that guarantees consistent positive transfer across all tasks is currently unattainable, especially when dealing dissimilar tasks that may engender negative transfer. Identifying the misalignment between algorithm selection and task specificity as the primary cause of negative transfer, we present the Similarity Heuristic Lifelong Prompt Tuning (SHLPT) framework. This innovative strategy partitions tasks into two distinct subsets by harnessing a learnable similarity metric, thereby facilitating fruitful transfer from tasks regardless of their similarity or dissimilarity. Additionally, SHLPT incorporates a parameter pool to combat catastrophic forgetting effectively. Our experiments shows that SHLPT outperforms state-of-the-art techniques in lifelong learning benchmarks and demonstrates robustness against negative transfer in diverse task sequences.<sup>1</sup>

## 1 Introduction

Drawing on the remarkable capacity of humans to amass new knowledge throughout their lifetime, lifelong learning (LL) systems aim to emulate this progressive learning trajectory by sequentially mastering various tasks, each contributing to the system’s cumulative knowledge base. However, it is not trivial for deep learning models to achieve this ideal due to inherent challenges. These include the need to (1) avoid catastrophic forgetting - where the acquisition of new information can lead to the erosion of previously learned knowledge, and to

\* Corresponding author.

<sup>1</sup>Source code is available at <https://github.com/wcyuo23/SHLPT>.

Task Source → Target	Prompt tuning (w/o transfer)	Continual Initialization (w/ transfer)	Progressive Prompts (w/ transfer)
Yahoo → AG News		<b>86.83 ± 2.24</b>	85.33 ± 1.61
DBpedia → AG News	86.25 ± 1.75	83.92 ± 2.98	85.00 ± 1.73
Amazon → AG News		85.50 ± 1.75	86.17 ± 0.95
AG News → Yahoo		66.43 ± 1.53	65.17 ± 2.11
DBpedia → Yahoo	67.03 ± 0.46	<b>67.73 ± 1.10</b>	67.13 ± 1.65
Amazon → Yahoo		66.43 ± 1.53	65.77 ± 3.33
Yahoo → DBpedia		97.57 ± 0.91	97.94 ± 0.38
AG News → DBpedia	97.86 ± 0.50	<b>98.33 ± 0.42</b>	97.81 ± 0.89
Amazon → DBpedia		97.40 ± 0.47	97.76 ± 0.51
DBpedia → Amazon		48.86 ± 1.10	48.67 ± 3.70
Yahoo → Amazon	47.53 ± 3.95	43.73 ± 2.41	49.00 ± 3.89
AG News → Amazon		<b>50.73 ± 4.32</b>	50.60 ± 1.20

Table 1: Transfer learning results on AGNews, Yahoo, DBpedia, Amazon. We use accuracy as the metric. Continual initialization refers to initializing the prompt for the target task with the fine-tuned prompt obtained from the source task. Progressive Prompts (Razdaibiedina et al., 2022) refers to concatenating the prompts fine-tuned on the source task onto the randomly initialized prompt of the target task.

(2) promote efficient knowledge transfer - where the model can leverage past learning experiences to aid in the understanding and performance of future tasks. Addressing these challenges is crucial for the development of LL systems that can adapt and grow in a manner akin to human learning.

Recent advancements in lifelong learning of language model (LM) have integrated the concept of prompt tuning to enhance its capabilities. These approaches maintain the pre-trained model’s parameters, while training a small set of additional prompts to adapt the model to various downstream tasks. The efficiency and lightweight nature of prompt tuning align well with the demands of LL, sidestepping the need for the onerous storage of entire model versions for every new task. This technique facilitates the model to accumulate knowledge over time, adapt flexibly to new tasks, and recall how to perform on older tasks with the aid of task-specific prompts. Moreover, substantial ef-

fort is dedicated to the transfer of knowledge from past tasks. This includes methods such as prompt concatenation (Razdaibiedina et al., 2022), parameter sharing (Wang et al., 2022b,c), and weighted summation (Smith et al., 2023; Jiang et al., 2023a), which are pivotal in ensuring that knowledge is effectively retained and utilized throughout the lifelong learning process.

Current transfer learning methods commonly presuppose that earlier tasks can positively impact succeeding tasks. Razdaibiedina et al. (2022) leverages prompts from previous tasks in the process of learning new ones, and similarly, Zhu et al. (2022) utilizes prompts from former tasks as a foundation to harness prior knowledge. Nevertheless, we suggest that when there is a considerable dissimilarity between tasks, these approaches don't consistently assure positive transfer; on occasions, they might even provoke negative transfer. This phenomenon is vividly illustrated by the empirical analysis in Table 1, which shows the transfer efficiency among diverse tasks within different soft prompt learning. On the other hand, how to effectively utilize negative transfer during the lifelong learning remains an open question for research. In this paper, we will further explore strategies for customized transfer learning tailored to the characteristics of different tasks, aimed at achieving more efficient knowledge accumulation while mitigating the potential impacts of negative transfer.

To address the issue of inconsistent knowledge transfer, we introduce a new approach to lifelong prompt tuning, which we refer as to SHLPT (Similarity Heuristic Lifelong Prompt Tuning). First, we construct a prompt pool for learned tasks, thereby reducing the risk of forgetting. Then, we segment our knowledge transfer module into three components: (1) assessing the similarity between the current task and previous tasks, (2) categorizing the previous tasks into similar and dissimilar subsets, and (3) applying different transfer algorithms to each subset accordingly. For the initial step, we calculate an attention-weighted combination of past prompt embeddings and incorporate this into the current task's prompt. During optimization, the model assigns higher attention scores to tasks that are more beneficial. Then, we utilize this attention score as task similarity metric and split task set accordingly. For tasks deemed similar, we integrate their parameters to provide the current task with an optimized starting point. Conversely, for tasks that are dissimilar, we introduce a variety

of innovative regularization techniques aimed at guiding the pre-trained model towards accessing a broader range of knowledge. This nuanced approach allows the model to better adapt to each new task while preserving and effectively utilizing the knowledge from all previous learning experiences without negative transfer.

In this paper, we make several notable contributions: (1) We address the novel challenge of mitigating negative transfer and facilitating knowledge transfer from dissimilar tasks in lifelong learning, which is particularly important in sequences of low-similarity tasks. (2) We present SHLPT, an innovative lifelong prompt tuning technique that reduces forgetting and enables knowledge transfer across tasks with varying degrees of similarity. Our extensive experiments demonstrate that SHLPT surpasses existing methods on benchmark datasets. (3) We introduce a challenging benchmark characterized by low task similarity, which typically results in increased negative transfer. Our approach exceeds the performance of previous state-of-the-art methods in this context.

## 2 Related Work

**Lifelong Parameter Efficient Tuning.** Parameter efficient tuning tunes a subset of parameters of pre-trained language models, and can largely reduce the computation costs and memory usage (Houlsby et al., 2019; Li and Liang, 2021; Hu et al., 2021; Lester et al., 2021; Ben Zaken et al., 2022). For lifelong prompt tuning, LFPT5 (Qin and Joty, 2021) uses distillation loss and a generative replay to learn a continuous prompt. Progressive Prompts (Razdaibiedina et al., 2022) learns a new prompt for every task, and progressively concatenate it with previous prompts. LPT (Liang et al., 2023) employs a trainable binary mask on the overall prompt to selectively choose parameters for different tasks. L2P (Wang et al., 2022c) initializes a prompt pool and selects a certain number of prompts from it for each task. Through this approach, parameters among different tasks can be shared and isolated simultaneously. Base on this parameter pool architecture, CODA-prompt (Smith et al., 2023) employs an attention-based prompt selection strategy, Diana (Dai et al., 2023) and HiDe-Prompt (Wang et al., 2023a) decompose prompts into a hierarchical structure. In addition to lifelong learning on prompt tuning, O-LoRA (Wang et al., 2023b) introduces Orthogonal regularization to LoRA, facilitate

learning in mutually orthogonal subspaces for different tasks; CLASSIC (Ke et al., 2021b) adds task masks to the Adapter layer and utilizes contrastive loss to transfer knowledge between similar tasks.

**Similarity Heuristic Methods.** Similarity heuristic lifelong learning methods use task similarity to identify which tasks can transfer knowledge and minimize interference from dissimilar tasks. CAT (Ke et al., 2020) compares the performance between transfer model and reference model to determine whether use the transfer model. B-CL (Ke et al., 2021c) and CTR (Ke et al., 2021a) utilize capsule network and routing algorithm to cluster similar tasks along with their shareable features. CLASSIC (Ke et al., 2021b) creates views from the hidden space information of previous similar tasks, and uses the contrastive loss to help current tasks to learn shared knowledge.

### 3 Preliminary

#### 3.1 Lifelong Learning Setup

In lifelong learning, the language model will be sequentially finetuned across a series of  $n$  tasks, labeled  $T_1, \dots, T_n$ . These tasks could originate from various domains and types. The training objective is to minimize the expected loss of all learned tasks with no access to data from previous tasks. Let  $D_1, D_2, \dots, D_n$  be the set of datasets corresponding to tasks  $T_1, T_2, \dots, T_n$ , respectively. Each dataset  $D_i$  encompasses a collection of data  $(X^{i,j}, y^{i,j})$ , where  $X^{i,j} = [x_1, x_2, \dots, x_l]$  is the input text with length  $l$ , and  $y^{i,j}$  is the corresponding output. The model is trained sequentially on these datasets with loss function  $L_i$  specific to each task.

In this paper, we categorize LL scenarios into two types based on task similarity. In the first type (de Masson D’Autume et al., 2019; Razdaibiedina et al., 2022), the task sequence  $T_1, \dots, T_n$  exhibits a relatively high degree of similarity, meaning that the likelihood of any task  $T_i$  being significantly dissimilar from previous tasks  $T_i (i < t)$  is low. The second type represents the opposite situation. The latter is more likely to emerge during the early stages of a real-world system when it has not yet accumulated a substantial knowledge base. We suppose that dissimilar tasks sequences are more prone to negative transfer, and for the first time, we create a lifelong learning benchmark composed of tasks that are dissimilar and likely to induce negative transfer for research purposes.

#### 3.2 Prompt Tuning

Prompt tuning, as introduced by Lester et al. (2021), presents a resource-efficient methodology for adapting language models without necessitating extensive fine-tuning of the original model. This technique utilizes a small set of trainable parameters known as a prompt, denoted  $P = [p_1, p_2, \dots, p_{l_p}]$ , which serves as a prefixed sequence to the input token embeddings  $X_e = [e(x_1), e(x_2), \dots, e(x_l)]$ . Here,  $l_p$  denotes the length of the prompt,  $e$  symbolizes the embedding function, and each vector  $p_i \in \mathbb{R}^d$ . For simplification, we use  $X$  to replace  $X_e$  below.

In the process of prompt tuning, the model employs the combined sequence  $[P; X] \in \mathbb{R}^{(l_p+l) \times d}$  as the input for the LM. The objective function for adapting to a specific downstream task is given by  $\mathcal{L}_{Down}([P; X]) = -\log p(y | [P; X])$ , which seeks to maximize the likelihood of the correct output  $y$  given the input embedding  $X$  and the prompt vectors  $P$ . Notably, this fine-tuning procedure exclusively modifies the prompt parameters  $P$ , leaving the remaining weights of the LM fixed.

#### 3.3 Empirical Study of Negative Transfer in Prompt Tuning

**Definition of Negative Transfer.** Transfer learning leverages data or knowledge from source tasks to enhance target task’s learning performance. However, the effectiveness of transfer learning is not always guaranteed (Zhang et al., 2022); the performance on the target task may even worsen after transfer learning, a situation that is referred to as negative transfer (Pan and Yang, 2009). Let  $\mathcal{S}$  be one or more source tasks,  $\mathcal{T}$  a target task,  $\epsilon_{\mathcal{T}}$  target task’s test error,  $A(\mathcal{S}, \mathcal{T})$  a transfer learning algorithm between  $\mathcal{S}$  and  $\mathcal{T}$ ,  $A(\emptyset, \mathcal{T})$  the same algorithm without source tasks’ information. The test error reduction after transfer learning can be formulated as

$$r_{\mathcal{T}} = \epsilon_{\mathcal{T}}(A(\emptyset, \mathcal{T})) - \epsilon_{\mathcal{T}}(A(\mathcal{S}, \mathcal{T})) \quad (1)$$

A positive test error reduction  $r_{\mathcal{T}}$  indicates a successful transfer learning result, while a negative value indicates negative transfer.

**Negative Transfer in Prompt Tuning.** Vu et al. (2022) shows many tasks can benefit each other under soft prompt transfer learning setting. In order to examine the negative transfer phenomenon in this setting, we conduct an empirical study on the transferability of soft prompt across a collection of

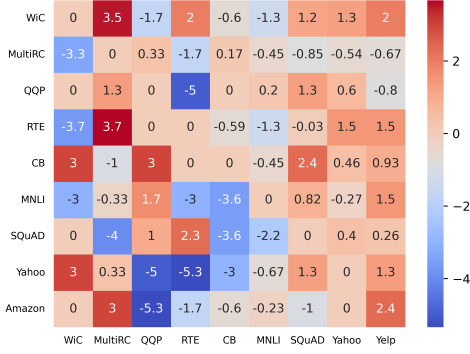


Figure 1: Test error reduction on the target tasks (column) after transferring from different source tasks (row). The negative transfer (indicated by cool colors) exists when use single transfer algorithm.

language tasks (details on the tasks are available in Appendix A, while experimental specifics can be found in Appendix C). Here, a prompt learned on one source task is transferred and used as the initial prompt for a target task. As shown in Figure 1, the negative transfer is a common occurrence in the setting of soft prompt transfer, particularly when the source and target tasks are highly dissimilar – for instance, Yahoo→RTE resulted in a 5.3% drop in accuracy (from 78.67 to 73.33). When the source and target task pairs are similar, transfers tend to yield positive results. For instance, Amazon→Yelp, both tasks involve sentiment analysis of reviews, leading to an 2.4% increase in accuracy. Interestingly, positive transfer can sometimes occur even between dissimilar tasks (e.g., CB→QQP). To better predict and avoid negative transfer, we design a similarity estimator that can discern transfer potential more effectively.

We further evaluated the few-shot (16 shot per class) performance of two task similarity agnostic transfer learning methods on four tasks from Standard CL Benchmark (de Masson D’Autume et al., 2019): AGNews, Yahoo, DBpedia, Amazon. Results in Table 1 indicate that these two methods also cannot guarantee positive transfer among all tasks (e.g., DBpedia→AG News, Amazon→Yahoo).

**Negative Transfer Benchmark.** Based on above results, we have developed a benchmark for lifelong learning research that consists of tasks prone to negative transfer (Appendix B). This benchmark requires lifelong learning algorithms to effectively avoid negative transfer or further benefit from tasks involving negative transfer.

**Theoretical Analysis.** We further analyze the causes of negative transfer from theoretical bound

for domain adaptation (Ben-David et al., 2010).

$$\epsilon_{\mathcal{T}}(A(\mathcal{S}, \mathcal{T})) \leq \epsilon_{\mathcal{S}}(A(\mathcal{S}, \mathcal{T})) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_s, D_t) + \lambda \quad (2)$$

where  $D$  is domain distribution,  $d_{\mathcal{H}\Delta\mathcal{H}}(D_s, D_t)$  is the divergence between source domain  $\mathcal{S}$  and target domain  $\mathcal{T}$ ,  $\lambda$  is a problem-specific constant. Based on the theoretical bound, negative transfer in LL can be attributed to the following reasons: (1) large error in the source tasks. This is caused by the few-shot setting or the learning difficulty of source tasks. (2) large distribution divergence. This is because the tasks are too dissimilar. (3) Unsuitable transfer learning algorithms. Transfer learning algorithms cannot adapt to all scenarios, an unsuitable algorithm may leads to negative transfer.

## 4 Method

To this end, we propose SHLPT (Similarity Heuristic Lifelong Prompt Tuning), which intends to achieve the following goals for a robust lifelong learning system: (1) avoid forgetting previous knowledge, (2) transfer knowledge from both similar and dissimilar tasks, (3) reduce the probability of negative transfer during the transfer stage.

SHLPT is composed of the following three structures: (1) A prompt pool preserving previous task’s prompts  $\{P^1, P^2, \dots, P^t\}$  to avoid forgetting of past learning in a lifelong learning context. During the testing phase, identification of the task is necessary. (2) A task similarity estimator which can estimate the similarity between current tasks and previous tasks. We operate under the premise that "the more similar the tasks, the more effective the knowledge transfer." Based on estimated similarity, the collection of previous prompts is further categorized into two subsets: prompts from similar tasks  $P_s^1, P_s^2, \dots, P_s^m$ , and those from dissimilar tasks  $P_d^1, P_d^2, \dots, P_d^n$ . (3) Two transfer learning algorithms address transfer problems in different scenarios. One to disseminate shared knowledge extracted from similar tasks, and the other to differentiate core features from dissimilar tasks. Each structure is detailed below.

### 4.1 Attention-based Similarity Estimator

To tailor the transfer learning algorithm for appropriate tasks, we first conduct a similarity assessment and partitioning of the previous task. When training task  $t$ , the previous task’s prompt embeddings are recorded as  $\{P^1, P^2, \dots, P^{t-1}\}$ . Rather than calculate similarity between different tasks,



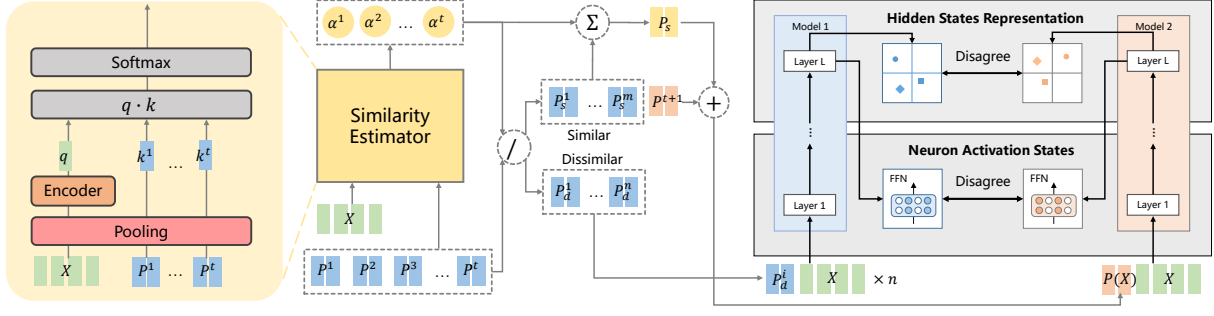


Figure 2: Illustration of our method SHLPT. The previous task prompts are partitioned based on an instance-wise similarity. Then, different transfer learning algorithm is applied on similar and dissimilar task scenarios. Similar tasks’ prompts are composed and added to current task prompt. The current task’s model behavior and representation are pushed away from those of dissimilar tasks. Only current task’s prompt  $P^{t+1}$  and encoder in similarity estimator are trainable.

we calculate similarity between task  $t$ ’s every instance and previous tasks, which is more robust to data variance (Wu et al., 2022). Given that we lack access to data from past tasks, our similarity assessment relies on the prompts of previous tasks. Let  $\alpha_i(X)$  to denote the instance-wise attention value for previous task  $i$  given  $X$ . It is calculated by extracting features from three elements: previous task prompts  $\{P^1, P^2, \dots, P^{t-1}\}$ , current task prompt  $P^t$  and instance embedding  $X$ . The max pooling operation is first applied to  $X \in \mathbb{R}^{l \times d}$  and  $P^i \in \mathbb{R}^{l_p \times d}$ , and transforms them into compact forms  $\hat{X}, \hat{P}^i \in \mathbb{R}^d$ , respectively.  $\hat{X}$  is then fed to a two-layer network to map it to the prompt space:

$$\tilde{X} = \text{LayerNorm}(\text{SiLU}(\hat{X}W_{down})W_{up}), \quad (3)$$

where  $W_{down} \in \mathbb{R}^{d \times r}$  and  $W_{up} \in \mathbb{R}^{r \times d}$  are trainable projection weights, SiLU (Elfwing et al., 2018) is employed as an activation function and LayerNorm (Ba et al., 2016) is used to address the issue of gradient explosion, following Asai et al. (2022). Finally, the similarity  $\alpha_i$  of previous task  $i$  can be obtained through tempering softmax of inner product between  $\tilde{X}$  and  $\hat{P}^i$

$$\alpha_i(X) = \frac{e^{\tilde{X}\hat{P}^i} / \tau_{sim}}{\sum_{j=1}^{t-1} e^{\tilde{X}\hat{P}^j} / \tau_{sim}} \quad (4)$$

where  $\tau_{sim}$  is a modifiable temperature parameter controlling the separation of similarity. Lian et al. (2020a) points out that the parameter  $\tau$  in tempering softmax is an important factor controlling the output distribution. If  $\tau$  is too large, the output is close to the uniform distribution. Conversely, if  $\tau$  is too small, there will be a gradient vanishing problem. The temperature should be neither too large nor too small (Lian et al., 2020b).

Having obtained the task similarity between current instance and previous tasks, previous tasks can be easily partitioned based on it. We compare similarity value  $\alpha_i$  with threshold  $\delta$ . If  $\alpha_i > \delta$ , the prompt corresponding to task  $i$  is classified into similar task prompt set  $\{P^{s_1}, P^{s_2}, \dots, P^{s_m}\}$ , otherwise it will be moved into dissimilar set  $\{P^{d_1}, P^{d_2}, \dots, P^{d_n}\}$ . Here  $s_i$  and  $d_j$  denotes the original indicate of  $i$ -th similar task and  $j$ -th dissimilar task, respectively.

$$\begin{cases} P \rightarrow \{P^{s_1}, P^{s_2}, \dots, P^{s_m}\}, & \text{if } \alpha > \delta \\ P \rightarrow \{P^{d_1}, P^{d_2}, \dots, P^{d_n}\}, & \text{if } \alpha \leq \delta \end{cases} \quad (5)$$

We use a predefined threshold to clearly partition the previous task into two sets, enabling the model to avoid negative transfer by applying the suitable transfer algorithm for different task scenario.

## 4.2 Similar Task Transferring

Towards previous tasks similar to current tasks, we consider transferring by prompt initialization perspective. Previous work has shown prompt tuning is sensitive to parameter initialization and usually suffer from slow convergence (Lester et al., 2021; Vu et al., 2022; Wang et al., 2022a; Shi and Lipani, 2023). Thus, we use a mixture of similar tasks’ prompt embeddings to initialize prompt, and sum it with a newly allocated prompt  $P^t$  for the current task. This method not only enhances model performance on current task, but also reduces the overall training time. The final prompt is derived as:

$$P(X) = \sum_{j=1}^{t-1} \tilde{\alpha}_j(X)P^j + P^t \quad (6)$$

Here, only the current prompt  $P^t$  and weights in similar estimator network are trainable. The

prompt  $P^j$  of previous tasks are fixed, thus preventing forgetting or backward regression on those tasks. The dissimilar tasks’ prompts are explicitly excluded to reduce their interference. Mixture value  $\tilde{\alpha}_i$  is obtained from following steps. First, per task similarity is input into a threshold function to let dissimilar tasks’ similarity value to 0.

$$\hat{\alpha}_i = \begin{cases} \alpha_i, & \text{if } \alpha_i > \delta \\ 0, & \text{if } \alpha_i \leq \delta \end{cases} \quad (7)$$

Then, a normalization operation is applied on it to let the overall sum equal to 1. At the end of current task training stage, all training samples’ similarity  $\alpha$  are averaged and used to calculate final prompt embedding  $P$  of current task. Then  $P$  is added to the prompt pool.

### 4.3 Dissimilar Task Transferring

Exclude dissimilar tasks from specific transfer learning algorithm can also mitigate negative transfer. Furthermore, we explore the use of an alternative transfer learning algorithm capable of leveraging dissimilar tasks to facilitate positive transfer effects. This is a novel problem that do not mentioned in previous lifelong learning researches.

Rather than transfer knowledge from prompt parameters (Asai et al., 2022; Wang et al., 2022a), our approach leverages the knowledge embedded in a pre-trained model with selected prompts. Since higher layers of a pre-trained model often exhibit more task-specific behavior (Liu et al., 2019), we introduce two novel loss functions that based on language model’s inner representation. These functions are designed to differentiate the behaviors of the current task from those of dissimilar tasks.

#### 4.3.1 Hidden States Contrastive Loss(HSC)

Recognizing that dissimilar tasks may offer limited shareable knowledge with the current task, our approach emphasizes the divergence in the output representations of these tasks. we prepend the transferred current task’s prompt  $P(X)$ , as well as the prompts from the dissimilar tasks  $\{P^{d_1}, P^{d_2}, \dots, P^{d_n}\}$ , and input the combined results into a pre-trained language model. We then calculate the last hidden states of the decoder,  $h$  for the current task, and  $h_1, \dots, h_n$  for the  $n$  dissimilar tasks, all based on the same instance. The pairs  $(h, h_1), \dots, (h, h_n)$  are treated as negative examples, while  $(h, h)$  is treated as a positive example. The

hidden states contrastive(HSC) loss is defined as

$$\mathcal{L}_{HSC} = -\log \frac{\exp(\cos(h, h)/\tau_{hsc})}{\exp(\cos(h, h)/\tau_{hsc}) + \sum_{k=1}^n \exp(\cos(h, h_k)/\tau_{hsc})} \quad (8)$$

where  $\cos$  refers to cosine similarity,  $\tau_{hsc}$  refers to temperature parameter. The HSC loss is designed to diverge the hidden states representation between current task and dissimilar tasks for a same input.

#### 4.3.2 Activation States Contrastive Loss(ASC)

Previous work has shown activation states of neurons in transformers’ feed forward network are associated to specific behaviour (Geva et al., 2021; Dai et al., 2022), and can be used to calculate task similarity (Su et al., 2022). The feed forward network FFN can be formulated as

$$\text{FFN}(h) = f(hW_i)W_o \quad (9)$$

where  $f$  is an activation function,  $h$  is the hidden states and  $W_i, W_o$  are parameter matrices.

We denote activation value  $f(hW_i)$  as  $s$ . The activation states, computed as  $\text{sign}(s)$ , take the form of binary vectors where each element indicates the status of a particular neuron.

Since the activation value  $s$  contains more information, we use it for our implementation. We then add a mask  $m$  to filter out neurons activated by instance X.

$$m = 1 - \text{sign}(s_0), \hat{s} = s \odot m, \quad (10)$$

where  $s_0$  is the activation value when no prompt is prepended,  $\odot$  refers to element-wise multiply. As previous probing experiments has shown higher layers’ feature are more task-specific (Liu et al., 2019), we construct our activation contrastive loss based on last FFN layers’ activation states from current task:  $\hat{s}$  and dissimilar tasks:  $\{\hat{s}_1, \dots, \hat{s}_n\}$

$$\mathcal{L}_{ASC} = -\log \frac{\exp(\cos(\hat{s}, \hat{s})/\tau_{asc})}{\exp(\cos(\hat{s}, \hat{s})/\tau_{asc}) + \sum_{k=1}^n \exp(\cos(\hat{s}, \hat{s}_k)/\tau_{asc})} \quad (11)$$

where  $\tau_{asc}$  refers to temperature parameter. The activation states contrastive loss aims to reduce the overlapping rate of activation states between current task and dissimilar tasks.

Finally, the overall loss is computed by

$$\mathcal{L} = \mathcal{L}_{Down} + \lambda_1 \mathcal{L}_{HSC} + \lambda_2 \mathcal{L}_{ASC} \quad (12)$$

where  $\mathcal{L}_{Down}$  is the standard prompt tuning loss on the downstream task. In addition, we use the hidden states and activation states at first position of the decoder output to compute contrastive losses.

## 5 Experimental Setup

### 5.1 Datasets and Metrics

We utilize three benchmarks to evaluate the model performance:

**Standard CL Benchmark** is a widely used benchmark for lifelong language learning models’ evaluation. It consists of four text classification datasets on different tasks or domains (Zhang et al., 2015): AGNews (topic classification), Yahoo (QA categorization), DBpedia (Wikipedia article classification), Amazon (sentiment analysis).

**Large Number of Tasks** consists of 15 classification tasks and is used to evaluate lifelong learning methods’ performance on long sequences of tasks (Razdaibiedina et al., 2022). It includes four datasets from standard CL benchmark, four datasets from GLUE benchmark (Wang et al., 2018), five datasets from SuperGLUE benchmark (Wang et al., 2019a), Yelp reviews (Zhang et al., 2015) and IMDB reviews (Maas et al., 2011).

**Negative Transfer Benchmark** is a benchmark that we introduced to evaluate the robustness of a lifelong learning system under sequences of dissimilar/negative transfer tasks. We construct the benchmark in the following steps: First, we use initialization from the source task as the transfer algorithm and test which source/target task pair exhibits negative transfer in a collection of datasets: MNLI, QQP, RTE from GLUE benchmark (Wang et al., 2018), WiC, CB, COPA, BoolQ, MultiRC from SuperGLUE benchmark (Wang et al., 2019a), SQuAD 2.0 (Rajpurkar et al., 2018), Yahoo, Yelp and Amazon (Zhang et al., 2015). The experiment result is shown in Figure 1 and Appendix C. Then, we construct three dissimilar task sequences, with the requirement that preceding tasks induce negative transfer on subsequent tasks.

The detailed information regarding the task sequences are provided in the Appendix B. We use normalized F1 score (McCann et al., 2018) for SQuAD, accuracy for other datasets. The task details and metrics are provided in the Appendix A.

### 5.2 Baselines and Training Details

We compare SHLPT with the following baselines, including recent SOTA methods. **Finetune**: continually finetunes the whole model parameters on sequences of tasks. **Online EWC** (Schwarz et al., 2018): utilizes a regularization loss to constrain updates on crucial parameters associated with previous tasks. **ER** (Chaudhry et al., 2019): replays

samples from previous tasks when training future tasks. **Per-task Prompts**: trains each task with a separate prompt and keeps the remaining parameters fixed. This represents prompt tuning without transfer. **L2P** (Wang et al., 2022c): maintains a prompt pool and selects prompts from it using an instance-wise query. **CODA-Prompt** (Smith et al., 2023): implements instance-wise prompts through a weighted summation of prompts from the pool. **ProgPrompt** (Razdaibiedina et al., 2022): train a new prompt for each task and progressively concatenate it with prompts from old tasks.

To ensure a fair comparison with ProgPrompt and SHLPT, task identity is provided during test stage for L2P and CODA-prompt. While the task identity is not mandatory for these two methods, providing it can enhance their performance. The above methods are all implemented on original T5-large model. We report the mean results over three runs with different random seeds. The temperature  $\tau_{sim}$  for similarity estimator is set to  $2 \times 10^4$ . The temperature  $\tau$  for each contrastive loss is set to 1 while the weights  $\lambda_1$  and  $\lambda_2$  of the losses are set to 0.1 and 0.5 respectively. The similarity threshold  $\delta$  is set to 0.06 for Standard CL Benchmark and Large Number of Tasks, and 1.5 for Negative Transfer Benchmark. We include further analysis of the sensitivity of SHLPT to the similarity threshold, as well as additional training details, in Appendix D.

## 6 Results and Analysis

### 6.1 Results on Existing Benchmark

We first evaluate our method and other baselines on two existing benchmark: Standard CL Benchmark and Large Number of Tasks. Considering the task order may impact the results, we conduct experiments under three different task orders. Table 2 shows that our method SHLPT outperforms all baselines in these two existing benchmarks. SHLPT improves the recent SOTA with an increase of accuracy by 2.6% on Standard CL Benchmark and 0.95% on Large Number of Tasks. Since prompt-based methods can all avoid forgetting, we attribute the improvement of SHLPT to its better inter-task transfer effect.

In addition to the average accuracy metrics, we also provide the backward transfer scores and forward transfer scores in Appendix H. These scores are employed to evaluate the effectiveness of mitigating forgetting and negative transfer.

Method	Standard CL Benchmark				Large Number of Tasks			
	Order1	Order2	Order3	Avg	Order4	Order5	Order6	Avg
Finetune	30.92	32.27	35.94	33.04	13.40	11.58	14.38	13.12
Online EWC	62.82	57.31	66.37	62.17	49.70	49.82	48.61	49.38
ER	64.63	69.36	67.28	67.09	60.34	55.98	54.37	56.90
Per-task Prompts	78.50	78.50	78.50	78.50	75.51	75.51	75.51	75.51
L2P	76.26	75.63	74.97	75.62	73.86	73.67	74.07	73.87
CODA-Prompt	77.01	<b>80.16</b>	75.86	77.67	76.21	<b>77.02</b>	76.40	76.54
ProgPrompt	73.73	77.06	78.13	76.31	73.71	72.52	71.51	72.58
SHLPT(ours)	<b>80.21</b>	79.69	<b>80.93</b>	<b>80.28</b>	<b>77.62</b>	76.97	<b>77.87</b>	<b>77.49</b>

Table 2: Results on Standard CL Benchmark and Large Number of Tasks. We present the model’s average accuracy after learning the last task. The standard deviations are provided in Appendix G.

Method	Negative Transfer Benchmark			
	Seq1	Seq2	Seq3	Avg
Finetune	44.91	46.00	22.95	37.95
Online EWC	70.68	59.30	56.79	62.26
ER	58.18	66.26	62.41	62.28
Per-task Prompts	<b>83.45</b>	81.65	69.03	78.04
L2P	82.60	80.59	68.04	77.08
CODA-Prompt	82.66	80.08	69.71	77.48
ProgPrompt	78.79	80.00	67.58	75.46
SHLPT(ours)	83.37	<b>82.47</b>	<b>70.16</b>	<b>78.67</b>

Table 3: Results on Negative Transfer Benchmark. We report the average score after learning the last task. The standard deviations are provided in Appendix G.

## 6.2 Results on Negative Transfer Benchmark

We also assess SHLPT using a challenge Negative Transfer Benchmark that we proposed. Our motivation is to test robustness to sequences of dissimilar tasks. When similarity is low, negative transfer is more likely to occur. This is a more realistic lifelong learning scenario, requiring the model to mitigate negative transfer and transfer knowledge from dissimilar tasks. We conduct experiments on three different sequences that exhibit negative transfer and the overall results are shown in Table 3. SHLPT achieves an improvement of 1.2% in average score over CODA-Prompt. These findings indicate that SHLPT exhibits greater robustness when confronted with sequences of tasks involving negative transfer. The backward transfer scores and forward transfer scores are reported in Appendix H.

## 6.3 Ablation Studies

We present the ablation results in Table 4. "-ASE", "-ASC", "-HSC" and "-STT" denote models without Attention-based Similarity Estimator, without ASC loss, without HSC loss, without Similar Task Transferring, respectively. For "-ASE", we replace the similarity estimator with one that outputs the same similarity across all old tasks (for example,

Model	Seq1	Seq2	Seq3	Seq4	Avg
-ASE	81.84	80.54	70.30	80.15	78.21
-ASC	83.22	82.22	<b>70.92</b>	79.69	79.01
-HSC	<b>83.51</b>	82.01	70.13	80.01	78.92
-STT	83.37	<b>82.47</b>	70.16	79.49	78.87
-ASC,-HSC	81.06	82.44	70.58	80.20	78.57
<b>SHLPT</b>	83.37	<b>82.47</b>	70.16	<b>80.21</b>	<b>79.05</b>

Table 4: Ablation experiment results. Seq1-3 refer to sequences from Negative Transfer Benchmark, Seq4 refers to the sequence from Standard CL Benchmark with Order1. The standard deviations are provided in Appendix G.

{0.25, 0.25, 0.25, 0.25}). "-STT" and SHLPT have same result in Seq1-3, because there is no similar task in Negative Transfer Benchmark. We observe substantial drops when eliminating "STT" in Seq4, indicating that it contributes to certain transfer benefits. Table 4 shows that every component is effective and the full SHLPT gives best results.

## 6.4 Knowledge Transfer from Dissimilar Tasks

We further investigate whether the proposed two contrastive losses can facilitate positive knowledge transfers from dissimilar tasks. The experiments are conducted on eight task pairs where negative transfer is observed when transferring through initialization. Table 5 demonstrates that in the majority of cases, these two losses have positive transfers on dissimilar tasks. Notably, the combination of both losses yields the best transfer effects (+2.1%). This confirms SHLPT’s effectiveness in transferring knowledge across diverse task sequences.

## 6.5 Task Order Analysis

As Table 2 shows, task order may affect the performance of SHLPT. For example, SHLPT’s performance is relatively low on Order5. We further



Task Source → Target	Prompt tuning (w/o transfer)	Continual Initialization (w/ transfer)	Add HSC Loss (w/ transfer)	Add ASC Loss (w/ transfer)	Add ASC & HSC Loss (w/ transfer)
QQP → RTE		73.67	83.00	79.00	82.33
Yahoo → RTE	78.67	73.33	82.67	87.00	84.67
MNLI → CB		83.93	91.67	91.07	91.67
SQuAD → CB	87.50	83.93	92.26	89.88	89.29
COPA → QQP		81.00	81.67	85.67	86.67
Yahoo → QQP	86.00	81.00	85.33	85.67	85.33
COPA → MNLI		87.11	90.22	89.11	88.22
SQuAD → MNLI	88.67	86.44	90.00	89.78	90.00
Average Accuracy	85.21	81.30	87.10	87.15	87.27

Table 5: Different transfer learning algorithms’ transfer learning results on negative transfer task pairs. Positive transfer are shown in red and negative transfer are shown in blue.

investigate the possible reasons for this order’s impact on performance, as outlined below.

In Order5, SHLPT’s performance on RTE is inferior compared to others, as shown in Table 6. The difference in RTE’s previous tasks between three orders is that Order5 includes MultiRC, which is dissimilar to RTE in task type, and negative transfer (MultiRC → RTE) is observed in Table 10. This may indicate that SHLPT has poor transfer performance when transferring from MultiRC to RTE. In conclusion, the task order can impact SHLPT’s performance, as SHLPT has different transfer learning results based on different previous task sets.

Task	RTE	QQP	Yahoo
Order4	84.17	87.17	74.43
Order5	79.00	83.50	73.73
Order6	85.00	85.33	76.70

Table 6: Results of three tasks on which SHLPT performs poorly in Order5 compared to other orders.

The additional analysis about the training curves of task similarities and the impact of ASC loss on activation states is provided in Appendix E and Appendix F.

## 7 Conclusion

In this paper, we introduce a novel method SHLPT for lifelong language learning. SHLPT aims to alleviate negative transfer by learning task similarity in one step and employing suitable transfer method for different old task scenarios (similar or dissimilar). Experimental results show SHLPT outperforms baseline methods via better knowledge transfer on two existing benchmarks and our custom Negative Transfer Benchmark.

## Limitations

Though our approach has achieved significant results, there are still some limitations. Firstly, SHLPT requires task identity at training and inference stage. The identification and mitigation of negative transfer in task agnostic lifelong learning setting remain undiscovered. Secondly, we do not evaluate SHLPT on multilingual tasks (Wang et al., 2020) or other application tasks (Lian et al., 2014; Wang et al., 2019b; Jiang et al., 2023b), and negative transfer may also occur in these scenarios. Last, the scalability of SHLPT across different language models, especially LLM, has not been thoroughly researched. We leave these for our future work.

## References

- Akari Asai, Mohammadreza Salehi, Matthew 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.
- Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine learning*, 79:151–175.
- 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, Dublin, Ireland. Association for Computational Linguistics.

- Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc’Aurelio Ranzato. 2019. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*.
- Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. **Knowledge neurons in pretrained transformers**. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.
- Yi Dai, Hao Lang, Yinhe Zheng, Bowen Yu, Fei Huang, and Yongbin Li. 2023. **Domain incremental lifelong learning in an open world**. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5844–5865, Toronto, Canada. Association for Computational Linguistics.
- 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.
- Stefan Elfving, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. **Transformer feed-forward layers are key-value memories**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- 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*.
- Gangwei Jiang, Caigao Jiang, Siqiao Xue, James Zhang, Jun Zhou, Defu Lian, and Ying Wei. 2023a. **Towards anytime fine-tuning: Continually pre-trained language models with hypernetwork prompts**. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12081–12095, Singapore. Association for Computational Linguistics.
- Junguang Jiang, Baixu Chen, Junwei Pan, Ximei Wang, Dapeng Liu, Jie Jiang, and Mingsheng Long. 2023b. **Forkmerge: Mitigating negative transfer in auxiliary-task learning**. In *Advances in Neural Information Processing Systems*, volume 36, pages 30367–30389. Curran Associates, Inc.
- Zixuan Ke, Bing Liu, and Xingchang Huang. 2020. Continual learning of a mixed sequence of similar and dissimilar tasks. *Advances in Neural Information Processing Systems*, 33:18493–18504.
- Zixuan Ke, Bing Liu, Nianzu Ma, Hu Xu, and Lei Shu. 2021a. Achieving forgetting prevention and knowledge transfer in continual learning. *Advances in Neural Information Processing Systems*, 34:22443–22456.
- Zixuan Ke, Bing Liu, Hu Xu, and Lei Shu. 2021b. **CLASSIC: Continual and contrastive learning of aspect sentiment classification tasks**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6871–6883, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Zixuan Ke, Hu Xu, and Bing Liu. 2021c. **Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks**. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4746–4755, Online. Association for Computational Linguistics.
- 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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- 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, Online. Association for Computational Linguistics.
- Defu Lian, Qi Liu, and Enhong Chen. 2020a. Personalized ranking with importance sampling. In *Proceedings of The Web Conference 2020*, pages 1093–1103.
- Defu Lian, Haoyu Wang, Zheng Liu, Jianxun Lian, Enhong Chen, and Xing Xie. 2020b. Lightrec: A memory and search-efficient recommender system. In *Proceedings of The Web Conference 2020*, pages 695–705.
- Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 831–840.
- Zujie Liang, Feng Wei, Yin Jie, Yuxi Qian, Zhenghong Hao, and Bing Han. 2023. **Prompts can play lottery tickets well: Achieving lifelong information extraction via lottery prompt tuning**. In *Proceedings of the*

- 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 277–292, Toronto, Canada. Association for Computational Linguistics.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. [Linguistic knowledge and transferability of contextual representations](#). 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Andrew L. 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, Portland, Oregon, USA. Association for Computational Linguistics.
- Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.
- Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.
- 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.
- Chengwei Qin and Shafiq Joty. 2021. Lfpt5: A unified framework for lifelong few-shot language learning based on prompt tuning of t5. In *International Conference on Learning Representations*.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. [Know what you don’t know: Unanswerable questions for SQuAD](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.
- Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madihan Khabza, Mike Lewis, and Amjad Almahairi. 2022. Progressive prompts: Continual learning for language models. In *The Eleventh International Conference on Learning Representations*.
- Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, pages 4528–4537. PMLR.
- Zhengxiang Shi and Aldo Lipani. 2023. Dept: Decomposed prompt tuning for parameter-efficient fine-tuning. *arXiv preprint arXiv:2309.05173*.
- James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zolt 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.
- Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. 2022. [On transferability of prompt tuning for natural language processing](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3949–3969, Seattle, United States. Association for Computational Linguistics.
- Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou’, and Daniel Cer. 2022. [SPoT: Better frozen model adaptation through soft prompt transfer](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. 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 Bowman. 2018. [GLUE: A multi-task benchmark and analysis platform for natural language understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. 2023a. [Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 69054–69076. Curran Associates, Inc.
- Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing

- Huang. 2023b. [Orthogonal subspace learning for language model continual learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671, Singapore. Association for Computational Linguistics.
- Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogério Feris, Huan Sun, and Yoon Kim. 2022a. Multitask prompt tuning enables parameter-efficient transfer learning. In *The Eleventh International Conference on Learning Representations*.
- Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. 2022b. Dual-prompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer.
- Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022c. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149.
- Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. 2019b. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11293–11302.
- Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. [On negative interference in multilingual models: Findings and a meta-learning treatment](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450, Online. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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, Online. Association for Computational Linguistics.
- Zhuofeng Wu, Sinong Wang, Jiatao Gu, Rui Hou, Yuxiao Dong, V.G. Vinod Vydiswaran, and Hao Ma. 2022. [IDPG: An instance-dependent prompt generation method](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5507–5521, Seattle, United States. Association for Computational Linguistics.
- Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. 2022. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.
- Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. 2022. [Continual prompt tuning for dialog state tracking](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1124–1137, Dublin, Ireland. Association for Computational Linguistics.



## A Dataset Details

We present detailed information about datasets we used in Table 7. Following previous studies (Asai et al., 2022; Razdaibiedina et al., 2022), we use datasets from <http://goo.gl/JyCnZq> for Standard CL Benchmark, while using HuggingFace datasets library (<https://github.com/huggingface/datasets>) for the remaining datasets.

## B Task Sequence Details

We use six different orders of sequences in existing benchmark experiments, and the sequence information is presented in Table 8. For Negative Transfer Benchmark, we utilize three sequences composed of different datasets. The benchmark requires that preceding tasks induce negative transfer on subsequent tasks. Therefore, we do not alter the order as it may involve positive transfer. The sequence information is shown in Table 9.

## C Soft Prompt Transfer Results

The detailed results of our empirical study experiment are shown in Table 10. We perform transfer by initializing the target prompt from the source prompt. For classification tasks, we sample 100 samples per class to form the training set and validation set, and 100 samples per class to form the test set. For SQuAD dataset, we sample 400 samples to form the training set and validation set, and 400 samples to form the test set.

## D Implementation Details

We implement all methods with PyTorch (Paszke et al., 2019) and huggingface transformers (Wolf et al., 2020) library. All the experiments are run on eight NVIDIA 3090 GPUs. We set the max token length to 256 for all datasets. Following Razdaibiedina et al. (2022), we use the available validation set as the test set and create validation set from the training set. If not specifically mentioned, for classification tasks, we sample 75 samples per class to form the training set and validation set, and 100 samples per class to form the test set, following Qin and Joty (2021). For SQuAD dataset, we sample 300 samples to form the training set and validation set, and 400 samples to form the test set. We use AdamW optimizer (Loshchilov and Hutter, 2017) with a weight decay of 0.01 and a batch size of 8. All results are averaged over three runs with random seeds {42, 142, 242}.

For baselines that tuning all parameters (Fine-tune, online EWC, ER), we use the learning rate of  $1 \times 10^{-4}$ . For prompt-based methods, we use the learning rate of 0.3.

For SHLPT, we perform grid search on  $\tau_{sim}$  within  $\{2 \times 10^4, 2 \times 10^5, 2 \times 10^6, 2 \times 10^7\}$ ,  $\tau_{hsc}$  and  $\tau_{asc}$  within  $\{0.03, 0.5, 0.8, 1\}$ , and  $\lambda_1$  and  $\lambda_2$  within  $\{0.03, 0.1, 0.5, 0.8, 1.5\}$ . We set prompt token length to 150 and train the model with 80 epochs in Large Number of Tasks benchmark, while for others, we train 50 epochs. Early stopping mechanism is employed on all experiments. For similarity threshold  $\delta$ : as the Standard CL Benchmark and Large Number of Tasks contain some similar task pairs, we perform a grid search within the range of  $\{0.02, 0.04, \dots, 0.20\}$ . We opt for a relatively low value range as we aim to ensure that no similar tasks are mistakenly categorized as dissimilar. Then we select 0.06 for these benchmarks. For Negative Transfer Benchmark, the previous task are all dissimilar from current task, so we search in a relatively high value range:  $\{0.50, \dots, 1.50\}$ . And we select 1.5 for this benchmark.

Table 11 reports the searching results of threshold on Standard CL Benchmark in Order1. The results exhibits relatively minor changes as the threshold approaches 0. When the threshold increases, we notice that there is a certain performance decrease observed in the tasks towards the end of the task sequence. This is because the threshold value affects how and when SHLPT partitions previous tasks, thus further impacting the performance of the transfer algorithm. The tasks within the Standard CL Benchmark exhibit relatively high similarity, hence a lower threshold can prevent similar tasks from being partitioned as dissimilar.

## E Similarity Training Curve

We conduct an experiment on datasets from Standard CL Benchmark and show how similarity is learned in the similarity estimator. Figure 3 illustrates the variation of similarity between the Amazon task and four source tasks during SHLPT training. The similarity is quickly learned within a few steps, and the most similar task, Yelp, also aligns with intuition (both are sentiment analysis tasks for reviews).

## F Effects of ASC Loss on activation states

To investigate how much ASC loss affects activation states, we visualize the cosine similarity of

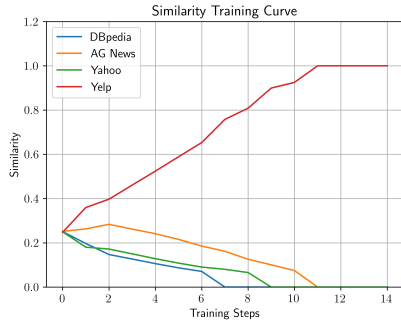


Figure 3: The variation of similarity output by the estimator as training steps increase. We only display a few steps in the early epochs because the similarity does not change afterwards.

activation states between prompts trained on different tasks (Figure 4). The similarity is calculated as follows: we average the similarity values of activation states from different prompts for all samples on each dataset (row). For the figure on the right, we added ASC Loss during the training process to diverge activation states from other tasks. As shown in Figure 4, activation states vary across different tasks, and adding ASC Loss reduces the similarity of these tasks’ activation states, thereby promoting more diverse neuron activation and a more varied utilization of pre-training knowledge.

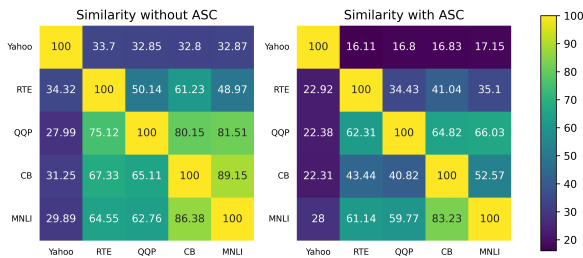


Figure 4: The cosine similarity of activation states at last layer obtained from prompts trained on different tasks.

## G Standard Deviations

Table 12, 13, 14 report the standard deviations of the results from Table 2 (on Standard CL Benchmark and Large Number of Tasks), 3 (on Negative Transfer Benchmark), 4 (ablation studies). Based on the standard deviation results, we find that the performance of SHLPT on the Large Number of Tasks benchmark is significantly better than other baselines. While on the Standard CL Benchmark and Negative Transfer Benchmark, SHLPT also shows a considerable improvement. We believe that the variations in improvement across differ-

ent benchmarks may stem from this reason: the Large Number of Tasks benchmark contains more datasets, thus resulting in a more significant cumulative performance gain for each task through transfer learning.

## H Backward Transfer Scores and Forward Transfer Scores

We further compare SHLPT and the baseline methods in terms of backward transfer scores and forward transfer scores, as defined in Wang et al. (2024). Table 15 reports the average results across three orders/sequences on three benchmarks.

The prompt-based methods, such as L2P, show no forgetting phenomenon, resulting in a backward transfer score of 0. Conversely, other baselines have negative backward transfer scores, as subsequent tasks induce forgetting, leading to impaired performance in previous tasks.

We use the forward transfer scores to measure negative transfer in lifelong learning, similar to the negative transfer gap in Wang et al. (2019b). A negative forward transfer score indicates negative transfer, while a positive score indicates positive transfer. As shown in Table 15, our method SHLPT achieves the best forward transfer score and exhibits the most effective mitigation of negative transfer in three benchmarks.

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	NLI	various	accuracy
7. QQP	GLUE	paraphrase	detection Quora	accuracy
8. RTE	GLUE	NLI	news, Wikipedia	accuracy
9. SST2	GLUE	sentiment analysis	movie reviews	accuracy
10. WiC	SuperGLUE	word sense disambiguation	lexical databases	accuracy
11. CB	SuperGLUE	NLI	various	accuracy
12. COPA	SuperGLUE	QA	blogs, encyclopedia	accuracy
13. BoolQ	SuperGLUE	boolean QA	Wikipedia	accuracy
14. MultiRC	SuperGLUE	QA	various	accuracy & F1
15. IMDB	Other	sentiment analysis	movie reviews	accuracy
16. SQuAD v2	Other	extractive QA	Wikipedia	nF1 & EM

Table 7: Details of 16 tasks used in our experiments. For datasets utilizing two metrics, the primary metric is the one listed first.

Order	Task Sequence
1	DBpedia→ Amazon→ Yahoo→ AG News
2	DBpedia→ Amazon→ AG News→ Yahoo
3	Yahoo→ Amazon→ AG News→ DBpedia
4	MNLI→ CB→ WiC→ COPA→ QQP→ BoolQ→ RTE→ IMDB→ Yelp→ Amazon→ SST2→ DBpedia→ AG News→ MultiRC→ Yahoo
5	MultiRC→ BoolQ→ WiC→ MNLI→ CB→ COPA→ QQP→ RTE→ IMDB→ SST2→ DBpedia→ AG News→ Yelp→ Amazon→ Yahoo
6	Yelp→ Amazon→ MNLI→ CB→ COPA→ QQP→ RTE→ IMDB→ SST2→ DBpedia→ AG News→ Yahoo→ MultiRC→ BoolQ→ WiC

Table 8: Different orders of task sequences used in existing benchmark experiments. Orders 1-3 are employed for Standard CL Benchmark. Orders 4-6 are employed for Large Number of Tasks.

Sequence ID	Task Sequence
1	Yahoo→ RTE→ QQP→ CB→ MNLI
2	QQP→ RTE→ SQuAD v2→ MNLI→ CB
3	MultiRC→ RTE→ SQuAD v2→ WiC→ MNLI

Table 9: Three different task sequences used in Negative Transfer Benchmark.

	WiC	MultiRC	QQP	RTE	CB	MNLI	SQuAD	Yahoo	Yelp	Amazon
Baseline	62.67	52.00	86.00	78.67	87.50	88.67	65.58	74.67	58.67	54.27
WiC		55.50	84.33	80.67	86.90	87.33	66.74	75.93	60.67	56.27
MultiRC	59.33		86.33	77.00	87.67	88.22	64.73	74.13	58.00	45.47
QQP	62.67	53.33		73.67	87.50	88.87	66.84	75.27	57.87	53.73
RTE	59.00	55.67	86.00		86.91	87.33	65.55	76.13	60.13	54.93
CB	65.67	51.00	89.00	78.67		88.22	67.94	75.13	59.60	55.47
MNLI	59.67	51.67	87.67	75.67	83.93		66.40	74.40	60.13	54.80
SQuAD v2	62.67	48.00	87.00	81.00	83.93	86.44		75.07	58.93	55.47
Yahoo	65.67	52.33	81.00	73.33	84.52	88.00	66.85		60.00	56.80
Yelp	61.33	54.67	81.00	80.67	91.67	88.45	68.23	76.07		55.60
Amazon	62.67	55.00	80.67	77.00	86.90	88.44	64.57	74.67	61.07	

Table 10: Each cell in the columns represents the performance of the target task transferred from a specific source task (row). Baseline refers to the task accuracy without transfer. Positive transfers are shown in red while negative transfers are shown in blue.

Threshold $\delta$	0.02	0.04	0.06	0.08	0.10	0.15	0.20
Accuracy (%)	80.00 $\pm$ 0.20	80.03 $\pm$ 0.24	<b>80.21 <math>\pm</math> 0.37</b>	80.02 $\pm$ 0.18	80.02 $\pm$ 0.18	79.57 $\pm$ 0.25	79.63 $\pm$ 0.14

Table 11: Searching results for SHLPT’s optimal similarity threshold on the Standard CL Benchmark in Order1. We report the average accuracy after learning the last task.

Method	Standard CL Benchmark				Large Number of Tasks			
	Order1	Order2	Order3	Avg	Order4	Order5	Order6	Avg
Finetune	$\pm$ 1.38	$\pm$ 0.24	$\pm$ 3.61	$\pm$ 0.67	$\pm$ 3.98	$\pm$ 0.41	$\pm$ 0.41	$\pm$ 1.45
Online EWC	$\pm$ 4.37	$\pm$ 4.04	$\pm$ 4.16	$\pm$ 2.47	$\pm$ 9.44	$\pm$ 6.36	$\pm$ 5.74	$\pm$ 1.69
ER	$\pm$ 2.47	$\pm$ 1.04	$\pm$ 3.79	$\pm$ 2.27	$\pm$ 4.22	$\pm$ 6.06	$\pm$ 1.06	$\pm$ 1.82
Per-task Prompts	$\pm$ 2.20	$\pm$ 2.20	$\pm$ 2.20	$\pm$ 2.20	$\pm$ 0.92	$\pm$ 0.92	$\pm$ 0.92	$\pm$ 0.92
L2P	$\pm$ 0.98	$\pm$ 0.54	$\pm$ 0.44	$\pm$ 0.20	$\pm$ 1.00	$\pm$ 0.79	$\pm$ 1.00	$\pm$ 0.34
CODA-Prompt	$\pm$ 2.17	$\pm$ 1.40	$\pm$ 5.18	$\pm$ 1.72	$\pm$ 0.84	$\pm$ 0.01	$\pm$ 0.35	$\pm$ 0.33
ProgPrompt	$\pm$ 3.39	$\pm$ 2.19	$\pm$ 0.49	$\pm$ 1.86	$\pm$ 1.19	$\pm$ 1.50	$\pm$ 0.68	$\pm$ 0.41
SHLPT(ours)	$\pm$ 0.37	$\pm$ 0.40	$\pm$ 0.47	$\pm$ 0.07	$\pm$ 0.39	$\pm$ 1.32	$\pm$ 0.45	$\pm$ 0.42

Table 12: Standard deviations of the related metrics of SHLPT and baseline methods on Standard CL Benchmark and Large Number of Tasks.

Method	Negative Transfer Benchmark			
	Seq1	Seq2	Seq3	Avg
Finetune	$\pm$ 4.69	$\pm$ 1.13	$\pm$ 9.44	$\pm$ 1.67
Online EWC	$\pm$ 3.67	$\pm$ 7.16	$\pm$ 6.06	$\pm$ 5.38
ER	$\pm$ 8.42	$\pm$ 5.37	$\pm$ 1.23	$\pm$ 1.92
Per-task Prompts	$\pm$ 0.77	$\pm$ 1.36	$\pm$ 1.73	$\pm$ 1.25
L2P	$\pm$ 1.40	$\pm$ 1.79	$\pm$ 1.31	$\pm$ 1.46
CODA-Prompt	$\pm$ 0.15	$\pm$ 0.78	$\pm$ 2.26	$\pm$ 0.67
ProgPrompt	$\pm$ 1.25	$\pm$ 1.13	$\pm$ 2.44	$\pm$ 1.39
SHLPT(ours)	$\pm$ 1.21	$\pm$ 0.78	$\pm$ 0.39	$\pm$ 0.28

Table 13: Standard deviations of the related metrics of SHLPT and baseline methods on Negative Transfer Benchmark.

Model	Seq1	Seq2	Seq3	Seq4	Avg
-ASE	$\pm$ 1.73	$\pm$ 0.29	$\pm$ 1.07	$\pm$ 1.68	$\pm$ 0.75
-ASC	$\pm$ 0.62	$\pm$ 0.36	$\pm$ 1.57	$\pm$ 0.75	$\pm$ 0.20
-HSC	$\pm$ 0.31	$\pm$ 1.39	$\pm$ 0.39	$\pm$ 0.87	$\pm$ 0.33
-STT	$\pm$ 1.21	$\pm$ 0.78	$\pm$ 0.39	$\pm$ 0.43	$\pm$ 0.11
-ASC,-HSC	$\pm$ 1.36	$\pm$ 0.49	$\pm$ 0.80	$\pm$ 0.71	$\pm$ 0.52
<b>SHLPT</b>	$\pm$ 1.21	$\pm$ 0.78	$\pm$ 0.39	$\pm$ 0.37	$\pm$ 0.29

Table 14: Standard deviations of the related metrics of SHLPT and the ablations.

Method	Standard CL Benchmark		Large Number of Tasks		Negative Transfer Benchmark	
	BWT	FWT	BWT	FWT	BWT	FWT
Finetune	-62.00	-0.69	-64.41	-3.73	-44.02	-3.43
Online EWC	-21.56	-2.17	-24.27	-4.98	-13.30	-3.94
ER	-15.82	-1.21	-16.62	-4.55	-9.04	-8.06
L2P	0.00	-4.34	0.00	-2.30	0.00	-0.85
CODA-Prompt	0.00	-1.80	0.00	0.56	0.00	-0.49
ProgPrompt	0.00	-3.57	0.00	-3.88	0.00	-3.17
SHLPT(ours)	0.00	<b>1.36</b>	0.00	<b>1.45</b>	0.00	<b>0.99</b>

Table 15: The average backward transfer scores (BWT) and forward transfer scores (FWT) on Standard CL Benchmark, Large Number of Tasks and Negative Transfer Benchmark.