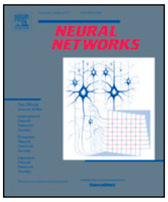
Neural Networks 190 (2025) 107676

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

Information-theoretic complementary prompts for improved continual text classification

Duzhen Zhang [a,1], Yong Ren [c,1], Chenxing Li [c], Dong Yu [d], Tielin Zhang [b,*]

[a] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates
[b] Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, China
[c] Tencent AI Lab, Beijing, China
[d] Tencent AI Lab, Bellevue, USA

ARTICLE INFO

Keywords:
Continual learning
Text classification
Complementary learning systems
Prompt tuning
Information-theoretic framework

ABSTRACT

Continual Text Classification (CTC) aims to continuously classify new text data over time while minimizing catastrophic forgetting of previously acquired knowledge. However, existing methods often focus on task-specific knowledge, overlooking the importance of shared, task-agnostic knowledge. Inspired by the complementary learning systems theory, which posits that humans learn continually through the interaction of two systems — the hippocampus, responsible for forming distinct representations of specific experiences, and the neocortex, which extracts more general and transferable representations from past experiences — we introduce Information-Theoretic Complementary Prompts (InfoComp), a novel approach for CTC. InfoComp explicitly learns two distinct prompt spaces: P(rivate)-Prompt and S(hared)-Prompt. These respectively encode task-specific and task-invariant knowledge, enabling models to sequentially learn classification tasks without relying on data replay. To promote more informative prompt learning, InfoComp uses an information-theoretic framework that maximizes mutual information between different parameters (or encoded representations). Within this framework, we design two novel loss functions: (1) to strengthen the accumulation of task-specific knowledge in P-Prompt, effectively mitigating catastrophic forgetting, and (2) to enhance the retention of task-invariant knowledge in S-Prompt, improving forward knowledge transfer. Extensive experiments on diverse CTC benchmarks show that our approach outperforms previous state-of-the-art methods.

1. Introduction

Mastering a wide range of tasks, accumulating experience, and preventing forgetfulness are fundamental characteristics of human-level intelligence. Despite the substantial advancements in Pretrained Language Models (PLMs) like BERT (Devlin, Chang, Lee, & Toutanova, 2019), their performance still deteriorates when confronted with a sequence of downstream text classification tasks—a scenario known as Continual Text Classification (CTC) (de Masson D'Autume, Ruder, Kong, & Yogatama, 2019; Huang, Zhang, Chen, Wang, & Yang, 2021). CTC presents two key challenges: (1) Catastrophic Forgetting (CF), where models tend to lose previously acquired knowledge when learning new tasks (Goodfellow, Mirza, Xiao, Courville, & Bengio, 2013; McCloskey & Cohen, 1989; Robins, 1995), and (2) Forward Knowledge Transfer (FKT), which involves utilizing knowledge from earlier tasks to enhance the learning efficiency of subsequent tasks.

As the parameter scale of modern PLMs continues to expand, fine-tuning the entire model becomes increasingly impractical, which has led to growing interest in Parameter-Efficient Fine-Tuning (PEFT) techniques. Among these, Prompt Tuning (PT) (Liu et al., 2022) has emerged as a prominent solution. PT works by learning a soft prompt and appending it to the input of PLMs, while keeping the model itself frozen (Liu et al., 2022). Previous research demonstrates that adapting PLMs to individual downstream tasks through prompt learning can achieve performance on par with full model fine-tuning, while using less than 0.01% of the parameters (Lester, Al-Rfou, & Constant, 2021; Liu, Yuan, et al., 2023). Essentially, PT shifts the focus from modifying the model's parameters to crafting prompts that guide the model's learning for specific tasks. These prompts encapsulate task-specific knowledge, making PT more effective in utilizing frozen PLMs compared to traditional fine-tuning approaches. In the context of CTC, harnessing PT to acquire and retain knowledge holds great potential (Wang, Zhang, Lee, et al., 2022).

Recently, Progressive Prompt (ProgPrompt) has been introduced as an extension of PT for handling sequential downstream tasks, achieving

[*] Corresponding author.
E-mail address: zhangtielin@ion.ac.cn (T. Zhang).
[1] Duzhen Zhang and Yong Ren are co-first authors of this paper. This work was done when Yong Ren was interning at Tencent, AI Lab, Beijing, China.

https://doi.org/10.1016/j.neunet.2025.107676
Received 31 October 2024; Received in revised form 24 March 2025; Accepted 22 May 2025
Available online 9 June 2025

State-Of-The-Art (SOTA) performance on various CTC benchmarks (Razdaibiedina et al., 2023). ProgPrompt learns distinct prompts for each incoming task and reuses them during inference, effectively mitigating CF. Moreover, it facilitates FKT by incorporating previously learned prompts (which remain frozen) as inputs when learning a new task. However, ProgPrompt faces several significant limitations. First, it focuses exclusively on task-specific knowledge without leveraging the knowledge shared across tasks. Second, by directly concatenating all previously learned prompts for FKT, it inevitably introduces redundant information. Lastly, ProgPrompt maintains a continuously growing prompt list. As the number of tasks (N) increases, the list grows at a rate of $\mathcal{O}(N)$, while transformer-based PLMs have a computational complexity of $\mathcal{O}(N^2)$ (Vaswani et al., 2017). This results in escalating costs for both training and inference as the number of tasks increases.

To overcome these limitations, we propose Information-Theoretic Complementary Prompts (InfoComp), a method designed to effectively guide PLMs in learning sequential downstream text classification tasks. Drawing inspiration from complementary learning systems (Kumaran, Hassabis, & McClelland, 2016; McClelland, McNaughton, & O'Reilly, 1995), we model our approach after the way humans engage in continual learning through the collaboration between the hippocampus and neocortex. The hippocampus excels at acquiring pattern-separated representations from specific experiences, while the neocortex captures more general and transferable representations from a sequence of past experiences (Arani, Sarfraz, & Zonooz, 2021; Pham, Liu, & Hoi, 2021; Wang, Zhang, Ebrahimi, et al., 2022). Based on this analogy, InfoComp explicitly learns two distinct prompt spaces: P(rivate)-Prompt and S(hared)-Prompt. The P-Prompt encodes task-specific knowledge, while the S-Prompt captures task-invariant knowledge shared across all tasks. Instead of concatenating all previously learned P-Prompts, InfoComp utilizes the task-invariant knowledge accumulated in the S-Prompt for FKT, thereby avoiding redundant information. When learning a new task, only the corresponding P-Prompt and the shared S-Prompt are required, ensuring that the prompt length remains constant as the number of tasks (N) increases.

To further improve the generation of more informative prompts, InfoComp frames soft PT as a process of maximizing Mutual Information (MI) between prompts and other model parameters, as well as between different encoded representations. Within this framework, we introduce two novel loss functions based on MI: (1) Enhancing task-specific knowledge in P-Prompt: We maximize the MI between the P-Prompt and the task-specific classifier. Since the classifier typically encapsulates critical information from the downstream task (Wu et al., 2023), optimizing this MI allows the P-Prompt to capture richer task-relevant information, thereby more effectively mitigating CF. (2) Preserving task-invariant knowledge in S-Prompt: We maximize the MI between the encoded representations of the same input under the current and previous S-Prompt conditions. By doing so, we ensure better preservation of task-invariant knowledge, facilitating more effective FKT and promoting continuity of shared knowledge across sequential tasks.

Our contributions can be summarized as follows:

- We introduce InfoComp for CTC, which integrates P-Prompt and S-Prompt to effectively capture task-specific and task-invariant knowledge, respectively. This method is easy to implement and eliminates the need for data replay, making it particularly well-suited for real-world continual learning applications.
- We advance prompt learning from an information-theoretic perspective by proposing two novel loss functions: one to enhance the acquisition of task-specific knowledge in P-Prompt, and the other to preserve task-invariant knowledge in S-Prompt.
- Extensive experiments on diverse CTC benchmarks, including both standard setups and more challenging scenarios with longer task sequences, demonstrate that InfoComp significantly outperforms previous SOTA methods.

2. Related work

2.1. Continual learning

Continual Learning (CL) aims to develop algorithms capable of progressively accumulating knowledge from a sequence of tasks (Wang, Zhang, Su, & Zhu, 2023). Traditional CL methods are generally divided into three categories: replay-based, regularization-based, and architecture-based approaches. Replay-based methods utilize a rehearsal buffer to retrain on a subset of previous examples (Chaudhry, Ranzato, Rohrbach, & Elhoseiny, 2018; Rebuffi, Kolesnikov, Sperl, & Lampert, 2017). However, their effectiveness diminishes as the buffer size decreases (Cha, Lee, & Shin, 2021), and they are unsuitable for scenarios where data privacy is a concern (Shokri & Shmatikov, 2015). Regularization-based methods address forgetting by imposing constraints on network weights (Farajtabar, Azizan, Mott, & Li, 2020; Huang et al., 2021; Kirkpatrick et al., 2017), intermediate features (Hou, Pan, Loy, Wang, & Lin, 2019), or output probabilities (Li & Hoiem, 2017). While these approaches mitigate forgetting, they often incur additional storage costs for gradients or models from previous tasks, and they tend to struggle with long task sequences. Architecture-based methods, on the other hand, dynamically allocate isolated modules to store knowledge from different tasks, thereby minimizing interference (Rusu et al., 2016; Yoon, Yang, Lee, & Hwang, 2018). However, this comes at the cost of significantly increasing the number of learnable parameters, making deployment more complex. Our approach is fundamentally architecture-based but stands out by requiring only a small set of parameters (*i.e.*, prompts) to be learned. This design allows for high efficiency in handling long task sequences without the need to store data, gradients, or models from previous tasks.

2.2. Prompt tuning

PT (Gu, Han, Liu, & Huang, 2022; Karimi Mahabadi, Henderson, & Ruder, 2021; Li & Liang, 2021; Wang, Panda, et al., 2023) is a lightweight approach for adapting PLMs to specific downstream tasks. This method involves optimizing a sequence of virtual tokens, known as soft prompts, which are appended to the input of PLMs, while keeping the PLMs themselves frozen. Recently, PT has been extended to the CL domain Liang et al. (2023), Qin and Joty (2022), Razdaibiedina et al. (2023), Wang, Liu, et al. (2023), Wang, Zhang, Ebrahimi, et al. (2022), Wang, Zhang, Lee, et al. (2022), Zhu, Li, Mi, Zhu, and Huang (2022). Among these methods, ProgPrompt (Razdaibiedina et al., 2023) stands out, achieving SOTA performance across various CTC benchmarks. It mitigates CF by learning distinct prompts for each task and facilitates FKT by progressively appending previously learned prompts as inputs when acquiring new tasks.

However, ProgPrompt has limitations. It overlooks the shared knowledge among tasks, and directly concatenating all learned prompts for FKT introduces redundant information. Furthermore, the growing prompt list increases computational overhead within the Transformer architecture. To address these issues, we propose InfoComp, which incorporates a S-Prompt to encode task-invariant knowledge shared across all tasks Bai, Liu, and Li (2024), Liu, Xiong, Yuan, and Wang (2023), ensuring that only essential knowledge is utilized for FKT, thereby avoiding redundant information. In addition, the prompt list remains constant in length during CL, and our information theory framework promotes the learning of more informative prompts.

2.3. Information theory approaches in natural language processing

Information theory, pioneered by Claude Shannon in the mid-20th century, provides a mathematical framework for quantifying information (Cover, 1999; Shannon, 1948, 1951). It has played a crucial role in advancing various natural language processing tasks, including language modeling, machine translation, and information retrieval (Brown, Della Pietra, Desouza, Lai, & Mercer, 1992; Manning

& Schutze, 1999; Steinborn, Dufter, Jabbar, & Schuetze, 2022; Tishby, Pereira, & Bialek, 2000). By measuring entropy and MI within language data, researchers have developed models that effectively capture linguistic structures and semantic relationships.

In text memorization, information-theoretic metrics help evaluate a model's ability to retain and reproduce textual information (Ju et al., 2021; West, Holtzman, Buys, & Choi, 2019). This involves quantifying the amount of preserved information during processing to prevent overfitting or underfitting, ensuring a balanced trade-off between memorization and generalization.

In multimodal translation, where textual and visual data are integrated, MI serves as a key criterion for aligning different modalities (Ji, Zhang, Zou, Hu, & Shen, 2022). By maximizing MI, models can better capture cross-modal dependencies, improving translation accuracy and robustness across multimodal inputs.

For model pretraining, InfoBERT (Wang et al., 2021) enhances BERT's robustness by applying the information bottleneck principle, which filters out task-irrelevant information to improve generalization and resilience against adversarial perturbations. Similarly, INFOXLM (Chi et al., 2021) employs MI-based optimization to refine cross-lingual language modeling, promoting semantic alignment across languages and enhancing zero-shot transfer capabilities. Furthermore, Wei, Tan, Li, Wang, and Huang (2024) introduces an evaluation metric based on the "effective rank" of model representations, drawing on insights from information theory and geometry to analyze and quantify how large language models eliminate redundant information throughout the training process. This approach offers a comprehensive assessment of model performance by capturing changes in representational efficiency before and after training.

For fine-tuning, Mahabadi, Belinkov, and Henderson (2021) introduces an information bottleneck approach to low-resource fine-tuning, retaining task-relevant information while minimizing redundancy to enhance efficiency and generalization. Moreover, Sorensen et al. (2022) and Wu et al. (2023) propose information-theoretic prompt engineering methods, leveraging MI to optimize prompt selection and improve language model conditioning.

Building on these methods, we leverage MI to quantify the information shared between two random variables, allowing us to compare prompts with other parameters as well as different encoded representations. This methodology facilitates the generation of more informative prompts, leading to improved task-specific and task-invariant knowledge representation for CTC.

3. Preliminary

3.1. Problem formulation

Following ProgPrompt (Razdaibiedina et al., 2023), this work explores a CL scenario in which a PLM is required to handle a sequence of n text classification tasks $(T_1, \dots, T_n)$. Each task T_k $(k = 1, \dots, n)$ is composed of a set of *i.i.d.* training examples $\{X_i^k, Y_i^k\}_{i=1}^{m_k}$, where X_i^k denotes the ith sequence of input text tokens, and Y_i^k is the corresponding label drawn from a predefined set $\mathcal{Y}_k$. The PLM, parameterized by Θ, is assumed to have access to task identity during both the training and inference phases.[2] Unlike many previous approaches, we do not assume access to data from previous tasks (*i.e.*, Rehearsal-free). Therefore, the PLM is limited to using data from task T_k only while training on that task. As a result, the learning objective across all tasks is defined as:

$$\max_{\Theta} \sum_{k=1}^{n} \sum_{i=1}^{m_k} \log p_{\Theta}(Y_i^k | X_i^k) \tag{1}$$

The simplest method for CL is finetuning, where the PLM sequentially minimizes the loss for each task T_k, $k \in \{1..n\}$, by updating all its parameters Θ (including the parameters of PLM and classifier):

$$\mathcal{L}_k(\Theta) = -\sum_{i=1}^{m_k} \log p(Y_i^k | X_i^k, \Theta) \tag{2}$$

While continual finetuning facilitates FKT to future tasks, it also leads to CF, where performance on previously learned tasks deteriorates after learning new ones, ultimately resulting in increased generalization loss (de Masson D'Autume et al., 2019; Kirkpatrick et al., 2017; McCloskey & Cohen, 1989).

3.2. ProgPrompt

To remedy this defect, ProgPrompt adopts a strategy of learning distinct prompts for each task to mitigate CF and facilitates FKT by progressively appending previous prompts (frozen) as inputs when learning a new task (Razdaibiedina et al., 2023). The training objective for the kth task T_k in Eq. (2) is therefore modified as:

$$\mathcal{L}_k(\theta_{P_k}, \theta_k^{\text{head}}) = -\sum_{i=1}^{m_k} \log p(Y_i^k | [P_k, \dots, P_1, X_i^k], \theta, \theta_k^{\text{head}}, \theta_{P_1}, \dots, \theta_{P_k}), \tag{3}$$

during training, the parameters θ of the PLM remain fixed, while only the current prompt-specific parameters θ_{P_k} and the current classifier parameters θ_k^{head} are updated for task T_k; they are frozen once the task T_k is completed.

It becomes apparent from Eq. (3) that ProgPrompt faces several limitations. First, it emphasizes task-specific knowledge while failing to exploit shared knowledge across different tasks. Second, its approach of concatenating all previously learned prompts to achieve FKT results in the inclusion of redundant information, which can reduce efficiency. Additionally, ProgPrompt continuously expands the list of prompts as new tasks are introduced. With the number of tasks (N) growing, the size of the prompt list increases linearly, at a rate of $\mathcal{O}(N)$. Given that transformer-based PLMs have a computational complexity of $\mathcal{O}(N^2)$ (Vaswani et al., 2017), this leads to rising costs in both training and inference as the number of tasks accumulates. Consequently, the approach becomes less scalable over time.

4. Method

To address the aforementioned limitations, we introduce the InfoComp method, which combines a pair of complementary prompts with an information-theoretic framework. This approach is designed to efficiently guide PLMs in learning a sequence of downstream text classification tasks, enabling better performance in CL scenarios.

4.1. Complementary prompts

The complementary learning systems theory (Kumaran et al., 2016; McClelland et al., 1995) in cognitive science suggests that humans rely on the interaction between the hippocampus and the neocortex to achieve CL. The hippocampus specializes in forming pattern-separated representations from specific, individual experiences, while the neocortex is responsible for extracting more general, transferable knowledge from a sequence of past experiences (Arani et al., 2021; Pham et al., 2021; Wang, Zhang, Ebrahimi, et al., 2022). This collaboration enables humans to effectively learn and adapt to new information while retaining previously acquired knowledge.

Building on this theory, InfoComp introduces a pair of complementary prompts: P-Prompt and S-Prompt, as illustrated in Fig. 1. For each task, a unique P-Prompt is learned to encode task-specific knowledge, helping to prevent interference between tasks and mitigate CF. Meanwhile, a shared S-Prompt is utilized across all tasks to capture

[2] We focus on the task-incremental setup, where task identity is generally assumed to be known during inference (Razdaibiedina et al., 2023).

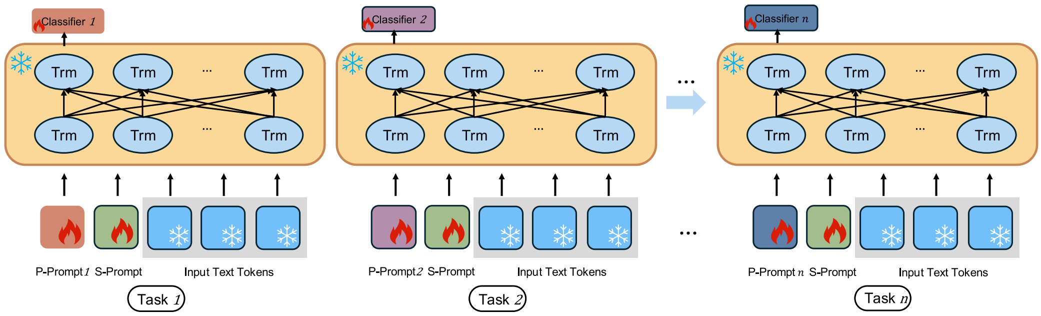


Fig. 1. InfoComp learns two distinct prompt types: P-Prompt and S-Prompt. The P-Prompt encodes task-specific knowledge to reduce CF, while the S-Prompt captures task-invariant knowledge to enhance FKT. "Trm" represents the Transformer encoder block within the PLM.

task-invariant knowledge, facilitating FKT. The training objective for the kth task T_k is formulated as follows:

$$\mathcal{L}_k(\theta_{P_k},\theta_S,\theta_k^{\text{head}}) = -\sum_{i=1}^{m_k}\log p(Y_i^k|[P_k,S,X_i^k],\theta,\theta_k^{\text{head}},\theta_{P_k},\theta_S), \tag{4}$$

where θ_{P_k} represents the parameters associated with the task-specific P-Prompt P_k, while θ_S denotes the parameters linked to the S-Prompt S.

In contrast to ProgPrompt as defined in Eq. (3), InfoComp in Eq. (4) introduces an additional S-Prompt to capture task-invariant knowledge shared across all tasks. Rather than concatenating all previously learned P-Prompts for FKT, InfoComp leverages the generalized knowledge stored in the S-Prompt, effectively reducing redundant information. When training on a new task, only the corresponding P-Prompt and the shared S-Prompt are used, ensuring that the total prompt length remains fixed, even as the number of tasks (N) increases. This design keeps the model efficient and scalable, regardless of the number of tasks involved.

4.2. Information-theoretic framework

To further enhance the learning of more informative prompts, InfoComp introduces an information-theoretic framework, framing PT as a process of maximizing MI between prompts and other model parameters, as well as across different encoded representations. This framework leads to the design of two novel MI-based training objectives: one aimed at enhancing task-specific knowledge in P-Prompt, and another focused on preserving task-invariant knowledge in S-Prompt. These objectives are intended to balance the integration of task-relevant insights with the retention of generalized knowledge.

Enhancing task-specific knowledge in P-prompt. Recent research has indicated that directly optimizing prompts alone is insufficient for encoding sufficient task-specific information (Wu et al., 2023). In contrast, task-specific classifiers are typically enriched with information from downstream tasks, and their parameters tend to adapt to task-specific knowledge more efficiently since they are closely aligned with the classification objectives. To explicitly enhance the task-specific information within the P-Prompt, we maximize the mutual information $I(\theta_{P_k};\theta_k^{\text{head}}|X_i^k)$ between the P-Prompt parameters θ_{P_k} and the classifier parameters θ_k^{head}, thereby more effectively mitigating CF.

Given the simplicity of vector dot product calculations and their gradient-friendly nature, we directly maximize the inner product between the P-Prompt parameters θ_{P_k} and the classifier parameters θ_k^{head}. This guides the model toward learning representations that are more closely related. By doing so, we effectively enhance the MI between these parameters. The process can be formalized as follows:

$$\mathcal{L}_k^{\text{p-info}}(\theta_{P_k},\theta_k^{\text{head}}) = -\theta_k^{\text{head}^T} W_1 \theta_{P_k}^T, \tag{5}$$

where W_1 is a trainable transformation matrix.

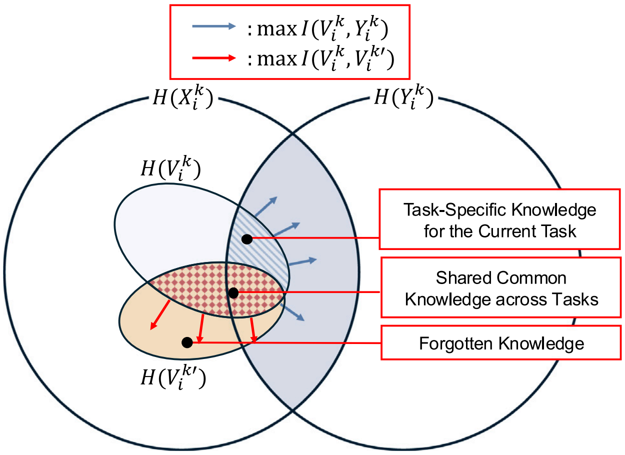

: $\max I(V_i^k, Y_i^k)$
: $\max I(V_i^k, V_i^{k'})$
$H(X_i^k)$
$H(Y_i^k)$
$H(V_i^k)$
$H(V_i^{k'})$
Task-Specific Knowledge for the Current Task
Shared Common Knowledge across Tasks
Forgotten Knowledge

Fig. 2. Alleviating the issue of forgetting shared knowledge across tasks by maximizing the MI $I(V_i^k, V_i^{k'})$.

Preserving task-invariant knowledge in S-prompt. Since the S-Prompt participates in the optimization of every task, it is inevitably affected by task-specific updates. To mitigate this influence and ensure the S-Prompt captures and retains as much task-agnostic, general knowledge as possible, we maximize the MI between the encoded representations of the same input using both the current and previous S-Prompt states. This strategy ensures better preservation of task-invariant knowledge, enhances FKT, and supports the continuity of shared knowledge across sequential tasks.

Let the PLM encoder be denoted as $F(\cdot)$. Given an input pair (X_i^k, Y_i^k) from the current task T_k, we define $V_i^k = F(X_i^k, S)$ as the representation generated using the current S-Prompt S, and $V_i^{k'} = F(X_i^k, S')$ as the representation produced by the frozen S-Prompt S', which was the S-Prompt at the end of the previous task optimization. Consider a Markov chain $V_i^{k'} \leftarrow X_i^k \rightarrow V_i^k \rightarrow Y_i^k$. From an information-theoretic perspective, optimizing for each task involves maximizing the MI $I(V_i^k, Y_i^k)$. As illustrated in Fig. 2, this process increases the amount of task-specific knowledge (represented by the blue arrow), while reducing the shared common knowledge across tasks. To address the issue of forgetting shared knowledge within the S-Prompt, we need to introduce an explicit constraint by maximizing the MI $I(V_i^k, V_i^{k'})$, following the direction of the red arrow.

Inspired by SimSiam (Chen & He, 2021), as shown in Fig. 3(a), it is a positive-only contrastive learning method that captures invariant features from positive sample pairs. Its training objective is formalized as follows:

$$\mathcal{L}^{\text{SimSiam}} = -(W_p z_1)^T z_2 = -p_1^T z_2, \tag{6}$$

where W_p is a trainable matrix designed to capture a shared feature embedding space between different views of positive samples (For

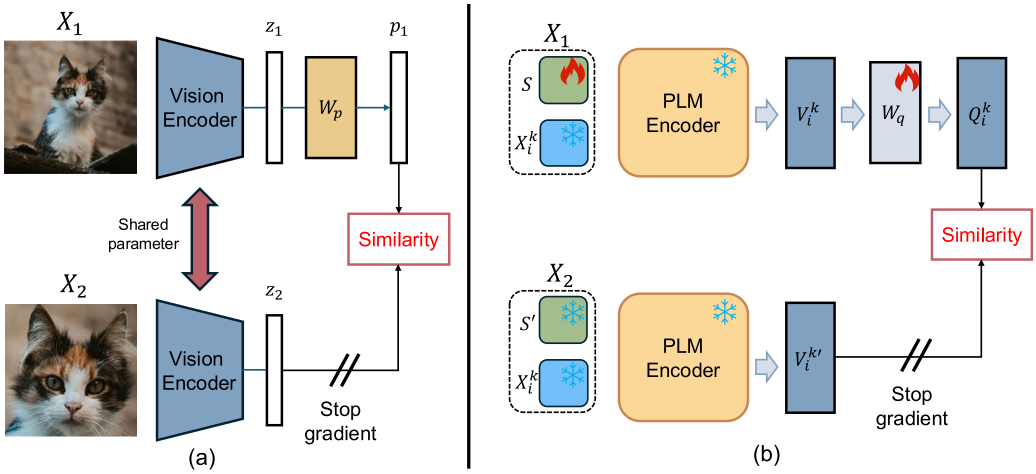


Fig. 3. (a) SimSiam for the positive sample pair. (b) SimSiam for the prompt-augmented positive sample pair.

example, as shown in Fig. 3(a), it depicts different views of the same cat.).

By analogy, we can view $X_1 = (X_i^k, S)$ and $X_2 = (X_i^k, S')$ as a pair of positive sample views of the same input X_i^k under different S-Prompt conditions from different tasks, as shown in Fig. 3(b). To increase the MI $I(V_i^k, V_i^{k'})$, we can adopt the SimSiam training objective, bringing this pair of positive samples closer in the embedding space to enhance the interdependence between V_i^k and $V_i^{k'}$. This can be formalized as follows:

$$\mathcal{L}_k^{\text{s-info}}(\theta_S) = -(W_q V_i^k)^T V_i^{k'} = -Q_i^{k^T} V_i^{k'}, \tag{7}$$

where W_q is a trainable matrix designed to capture the shared subspace between tasks. The parameters of the current S-Prompt, denoted as θ_S, are optimized to generate an embedding space that aligns with the S-Prompt from the previous task. This guides the current S-Prompt to expand the shared subspace with prior tasks, thereby enhancing the retention of previously learned general knowledge.

4.3. Overall training objective

Finally, the overall training objective function of our proposed InfoComp method is formulated as follows:

$$\mathcal{L}_k^{\text{overall}} = \mathcal{L}_k + \lambda_1 \mathcal{L}_k^{\text{p-info}} + \lambda_2 \mathcal{L}_k^{\text{s-info}}, \tag{8}$$

where λ_1 and λ_2 are hyperparameters that balance the importance of the loss terms.

5. Experimental settings

5.1. Datasets

Standard CTC benchmark. Building on the methodology outlined in ProgPrompt (Razdaibiedina et al., 2023), we initially assess our proposed approach using the widely recognized CTC benchmark, which comprises five text classification datasets introduced by Zhang, Zhao, and LeCun (2015): AG News, Amazon reviews, DBpedia, Yelp reviews, and Yahoo Answers. In line with earlier CTC studies such as IDBR (Huang et al., 2021) and MBPA++ (de Masson D'Autume et al., 2019), we conduct evaluations across four distinct task orderings of these five datasets. The training and testing sets are identical to those used in IDBR (Huang et al., 2021) and MBPA++ (de Masson D'Autume et al., 2019), comprising 115,000 training and 7,600 testing instances per task. Following the approach in Huang et al. (2021), for each task we randomly withhold 500 samples per class from the training set for validation, applying early stopping based on validation accuracy.

Table 1
The description of the 15 datasets utilized in our CTC experiments. "Acc." stands for accuracy. The first five tasks are part of the standard CTC benchmark, while the remaining tasks are included in our long-sequence experiments.

Name	Alias	Source	#.Class	Metric
1. AG News	ag	Standard	4	Acc.
2. Amazon reviews	amazon	Standard	5	Acc.
3. DBpedia	dbpedia	Standard	14	Acc.
4. Yelp reviews	yelp	Standard	5	Acc.
5. Yahoo Answers	yahoo	Standard	10	Acc.
6. MNLI	mnli	GLUE	3	Acc.
7. QQP	qqp	GLUE	2	F1
8. RTE	rte	GLUE	2	Acc.
9. SST2	sst2	GLUE	2	Acc.
10. WiC	wic	SuperGLUE	2	Acc.
11. CB	cb	SuperGLUE	3	Acc.
12. COPA	copa	SuperGLUE	2	Acc.
13. MultiRC	multirc	SuperGLUE	2	Acc.
14. BoolQ	boolq	SuperGLUE	2	Acc.
15. IMDB movie	imdb	Other	2	Acc.

Long-sequence CTC benchmark. A more realistic CTC scenario involves longer task sequences with a larger number of tasks. To address this, we also assess InfoComp's performance on a more challenging benchmark comprising 15 text classification tasks (Razdaibiedina et al., 2023). This benchmark includes the five datasets from the standard CTC benchmark, along with four tasks from the GLUE benchmark (MNLI, QQP, RTE, SST2) (Wang, 2018), five tasks from the SuperGLUE benchmark (Wang et al., 2019) (WiC, CB, COPA, MultiRC, BoolQ), and the IMDB movie reviews dataset (Maas et al., 2011). In line with ProgPrompt (Razdaibiedina et al., 2023), we evaluate performance across varying dataset sizes by creating three versions of each dataset, containing 20, 200, and 1000 training examples per class, and report test results for each case. Consistent with the approach in Huang et al. (2021), for each task we randomly set aside 500 samples per class from the training data for validation, using early stopping based on validation accuracy.

Table 1 provides a detailed overview of the 15 datasets employed in our CTC experiments. These datasets are drawn from the standard CTC benchmark (Zhang et al., 2015), the GLUE (Wang, 2018) and SuperGLUE (Wang et al., 2019) benchmarks, with the addition of the IMDB movie reviews dataset. The task orders used in our CTC experiments are presented in Table 2.

Table 2
Seven distinct task sequence orders were used for the CTC experiments. Orders 1 through 4 correspond to the standard CTC benchmark, while Orders 5 through 7 represent long-sequence settings involving 15 tasks.

Order	Task Sequence
1	ag → yelp → amazon → yahoo → db
2	yelp → yahoo → amazon → db → ag
3	db → yahoo → ag → amazon → yelp
4	yelp → ag → db → amazon → yahoo
5	mnli → cb → wic → copa → qqp → boolq → rte → imdb → yelp → amazon → sst2 → dbpedia → ag → multirc → yahoo
6	multirc → boolq → wic → mnli → cb → copa → qqp → rte → imdb → sst2 → dbpedia → ag → yelp → amazon → yahoo
7	yelp → amazon → mnli → cb → copa → qqp → rte → imdb → sst2 → dbpedia → ag → yahoo → multirc → boolq → wic

5.2. Baselines

We compare InfoComp against eight baseline methods[3]:

- **Finetune** (de Masson D'Autume et al., 2019; Huang et al., 2021; Wang, Mehta, Poczós, & Carbonell, 2020): all model parameters are trained sequentially on a series of tasks, without applying any regularization constraints or replaying samples from earlier tasks.
- **Experience Replay**: the entire model is fine-tuned using a memory buffer, with samples from previous tasks replayed during the learning of new tasks to mitigate forgetting.
- **A-GEM** (Chaudhry et al., 2018): store examples from previous tasks and limit the gradients used for updating the model on new tasks by referencing the retrieved examples.
- **MBPA++** (de Masson D'Autume et al., 2019): enhance BERT with an episodic memory that stores all encountered examples. Replay occurs during training, while local adaptation is applied during testing.
- **IDBR** (Huang et al., 2021): this BERT-specific method continuously trains the entire model by incorporating experience replay along with a regularization loss, which separates sentence representations into task-specific and task-generic components.
- **Per-task Prompts** (Lester et al., 2021): a separate soft prompt is trained for each task, while the original model's parameters remain unchanged. This method effectively avoids CF by ensuring that task-specific parameters are preserved as new tasks are introduced. However, it lacks FKT, as shared knowledge across tasks is not leveraged.
- **Prompt Tuning** (Lester et al., 2021; Qin & Joty, 2022): only a shared soft prompt is trained sequentially across all tasks, with the original model parameters kept fixed throughout.
- **InfoCL** (Song et al., 2023): a novel replay-based CTC approach that leverages fast-slow and current-past contrastive learning to enhance the representation of diverse knowledge.
- **Q-Tuning** (Guo et al., 2024): incorporates a prompt queue alongside an adaptive knowledge aggregation low-rank matrix, which is optimized to assess the significance of stored prompts and improve FKT.
- **SLM** (Bohao, Tian, Liu, Yang, & Jia, 2024): offers a model-agnostic framework for the scalable acquisition of knowledge and skills. It integrates vector space retrieval into the language model and features two key components: joint adaptive re-parameterization and dynamic retrieval of task-specific knowledge.
- **ProgPrompt** (Razdaibiedina et al., 2023): it learns unique prompts for each new task and reuses them during inference, effectively reducing CF. Additionally, it enables FKT by using previously learned, frozen prompts as inputs when training on a new task.

Table 3
Summary of results on the standard CTC benchmark. The highest result is highlighted in **red**, while the second-highest result is shown in **blue**. A † marker indicates a statistically significant result with a p-value of < 0.05 in comparison to ProgPrompt (Razdaibiedina et al., 2023) and SLM (Bohao et al., 2024). The column "ER" specifies whether experience replay is required by each baseline. Baselines labeled with $^{\diamond}$ maintain a frozen PLM, while the remaining methods train the entire model. Those marked with $*$ are derived from our re-implementation using their open-source code, whereas the other baseline results are directly cited from ProgPrompt (Razdaibiedina et al., 2023) or their respective original papers.

Baseline	ER	Order				
		1	**2**	**3**	**4**	Avg.
Finetune		14.8	27.8	26.7	4.5	18.4
Experience Replay	✓	67.2	64.7	64.7	44.6	57.8
A-GEM	✓	70.6	65.9	67.5	63.6	66.9
MBPA++	✓	70.8	70.9	70.2	70.7	70.6
IDBR	✓	75.9	76.2	76.4	76.7	76.3
InfoCL*	✓	76.5	76.8	70.0	76.3	74.9
ProgPrompt$^{\diamond}$		78.0	77.7	77.9	77.9	77.9
ProgPrompt$^{\diamond,*}$		78.0	77.8	77.8	77.8	77.9
Q-Tuning$^{\diamond}$		78.5	78.3	78.3	78.4	78.4
SLM$^{\diamond}$		**79.2**	**78.8**	**79.0**	**79.2**	**79.1**
InfoComp (Ours)$^{\diamond}$		**79.6**†	**80.1**†	**79.9**†	**80.2**†	**80.0**†
Improve		$\Uparrow$**0.4**	$\Uparrow$**1.3**	$\Uparrow$**0.9**	$\Uparrow$**1.0**	$\Uparrow$**0.9**

5.3. Evaluation metric

Following ProgPrompt (Razdaibiedina et al., 2023), we use average accuracy on the test set as our evaluation metric, representing the overall average accuracy across all tasks after training on the final task. Additionally, for certain task sequences, we include detailed task-wise accuracy plots. These plots capture the average accuracy of all tasks completed up to that point, measured after each task is trained. To assess the statistical significance of our improvements, we conduct a paired t-test at a significance level of 0.05 (Koehn, 2004).

5.4. Implementation details

For consistency with prior works, including IDBR (Huang et al., 2021), MBPA++ (de Masson D'Autume et al., 2019), and ProgPrompt (Razdaibiedina et al., 2023), we adopt the pre-trained bert-base-uncased model as the PLM backbone. Our implementation relies on PyTorch (Paszke et al., 2019) along with the HuggingFace Transformers library (Wolf, 2019).

For the standard CTC benchmark, we use the official datasets from (Zhang et al., 2015),[4] consistent with (de Masson D'Autume et al., 2019; Zhang et al., 2015). Additionally, we obtain data for GLUE tasks (Wang, 2018), SuperGLUE tasks (Wang et al., 2019), and the IMDB movie reviews dataset (Maas et al., 2011) using the HuggingFace Datasets library,[5] leveraging these datasets for long-sequence CTC experiments. In line with previous studies (de Masson D'Autume et al., 2019; Rao et al., 2019), we repurpose each dataset's validation set as the test set (where test data is unavailable) and reserve 500 samples from the training set to create the validation set.

We employ the Adam optimizer (Kingma, 2014) for all experiments, setting a batch size of 8 and a learning rate of $1e$-4. Prompts are trained for between 40 and 300 epochs, based on the dataset size. Specifically, for the standard CTC benchmark, we set the epoch count to 40. For the long-sequence CTC benchmark, the epoch count is set to 300, 150, and

[3] Consistent with (Huang et al., 2021; Razdaibiedina et al., 2023), in all replay methods, 1% of samples per class are stored (with a minimum of 1 sample for smaller datasets).

[4] http://goo.gl/JyCnZq

[5] https://github.com/huggingface/datasets

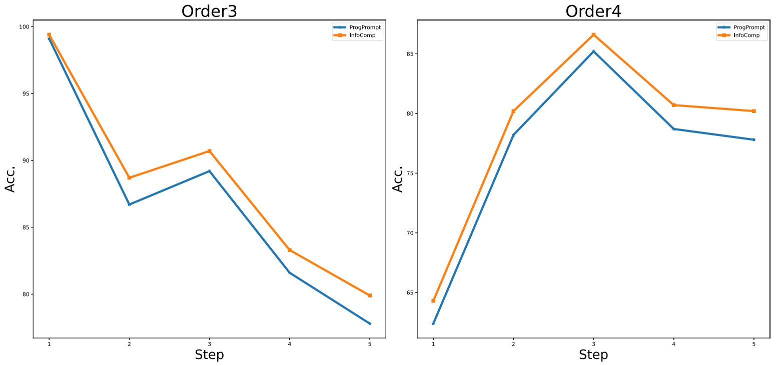


Fig. 4. Comparison of task-wise accuracy for the task sequences Order3 and Order4. Our InfoComp consistently outperforms ProgPrompt (Razdaibiedina et al., 2023), across all task-wise evaluations.

40 when each class contains 20, 200, and 1000 samples, respectively. The final prompts are selected based on the best validation set scores obtained from the prompt checkpoints. Following (Lester et al., 2021), prompts are initialized with randomly sampled tokens. In all experiments, we configure the P-Prompt length, G-Prompt length, and the loss coefficients λ_1 and λ_2 to be 35, 5, 0.05, and 0.1, respectively[6]. All of our results are the average of 3 runs. For all other baselines, we use hyperparameters outlined in their original papers.

6. Experimental results

6.1. Standard CTC benchmark

6.1.1. Main results

Table 3 presents a comparison of InfoComp's performance against four different task sequences and existing CTC methods, including the previous SOTA approaches, ProgPrompt (Razdaibiedina et al., 2023) and SLM (Bohao et al., 2024). Overall, InfoComp enhances CTC performance, achieving scores of 79.6, 80.1, 79.9, and 80.2 across the four sequences, with an average score of 80.0. This represents an improvement of 1.6, 2.3, 2.0, and 2.3 over ProgPrompt, and 0.4, 1.3, 0.9, and 1.0 over SLM, respectively. Furthermore, unlike traditional CTC methods such as IDBR (Huang et al., 2021), MBPA++ (de Masson D'Autume et al., 2019), and InfoCL (Song et al., 2023), InfoComp does not require the storage of data from previous tasks for future replay. These findings highlight the effectiveness of InfoComp.

6.1.2. Task-wise comparisons

Fig. 4 illustrates the task-wise comparison results that comprehensively assess the effectiveness of our InfoComp method in CTC scenarios. The findings reveal that InfoComp consistently surpasses ProgPrompt (Razdaibiedina et al., 2023), across all task-wise comparisons. This enhanced performance is particularly pronounced in the experimental setups for task sequences Order3 and Order4. These results emphasize the robustness of InfoComp in sustaining high performance and adaptability while tackling new tasks throughout the learning process. The marked improvement over ProgPrompt highlights InfoComp's potential to set a new benchmark for CTC tasks, particularly in complex, real-world data environments.

6.2. Long-sequence CTC benchmark

Table 4 presents a comparison of various common CTC approaches, including the SOTA method ProgPrompt (Razdaibiedina et al., 2023), within the context of sequence continual learning involving 15 tasks. We provide averaged results across three task orders (5, 6, and 7) and include the complete non-averaged results for each order. To examine the impact of limited data settings, we conduct training using different dataset sizes of 20, 200, and 1000 samples per class. Our method, InfoComp, consistently outperforms all competing approaches across these varying data constraints, achieving average scores of 57.9, 69.6, and 71.2 for the few-shot scenarios of 20, 200, and 1000 samples per class, respectively. This results in improvements of 3.7, 2.7, and 1.9 over the previous SOTA method, ProgPrompt.

6.3. Ablation study

This subsection assesses the effectiveness of the individual components of our InfoComp method in the task sequences Order6 and Order7 through ablation studies, as illustrated in Table 5. When we remove P-Prompt and S-Prompt from InfoComp, it indicates that only a shared soft prompt is trained sequentially across all tasks, or that a separate soft prompt is trained for each individual task. The first approach neglects task-specific knowledge, rendering it susceptible to CF for prior tasks, while the second approach overlooks shared knowledge among tasks, which can impede FKT to some extent. Both strategies result in a significant drop in performance compared to the complete InfoComp method, highlighting that effectively addressing CTC tasks requires simultaneously capturing both task-specific and task-invariant knowledge.

When we remove $\mathcal{L}_k^{\text{p-info}}$ and $\mathcal{L}_k^{\text{s-info}}$ from the InfoComp method, it indicates that the information-theoretic framework is not utilized to support the learning of prompts during training, which in turn affects the generation of more informative prompts. The individual removal of $\mathcal{L}_k^{\text{p-info}}$ compromises the enhancement of task-specific knowledge acquisition in P-Prompt, while the exclusion of $\mathcal{L}_k^{\text{s-info}}$ undermines the preservation of task-invariant knowledge in S-Prompt. As a result, these variations lead to a significant decline in CTC performance, underscoring the importance of each component in our InfoComp method for effectively addressing CTC challenges.

7. Conclusion

In this paper, we introduced InfoComp, a novel approach to CTC that effectively balances the learning of task-specific and task-invariant knowledge. InfoComp utilizes two distinct prompt spaces, P-Prompt and S-Prompt, enabling models to sequentially acquire new classification tasks while minimizing CF. Furthermore, by leveraging an information-theoretic framework, our designed loss functions significantly enhance the accumulation of task-specific knowledge in P-Prompt and improve the retention of task-invariant knowledge in S-Prompt. This reduces CF and fosters better FKT. Extensive experiments across various CTC benchmarks, including standard setups and more challenging scenarios with longer task sequences, demonstrated that InfoComp consistently outperforms existing SOTA methods, underscoring

[6] The prompt length refers to the sequence length of tokens in a prompt.

Table 4
Overview of results on the long-sequence CTC benchmark. The top result is highlighted in **red**, while the second-best is marked in **blue**. † denotes statistical significance with a p-value less than 0.05 compared to ProgPrompt (Razdaibiedina et al., 2023). Results are reported across different data limits: 20, 200, and 1000 samples per class. Baselines marked with $^\diamond$ indicate only soft prompts are trained while the PLM remains frozen, whereas other methods involve training the entire model. Results marked with $*$ come from our re-implementation, while other baseline results are directly sourced from ProgPrompt (Razdaibiedina et al., 2023).

Baseline ↓	Order 5			Order 6			Order 7			Avg.		
Num. samples →	20	200	1000	20	200	1000	20	200	1000	20	200	1000
Finetune	29.9	43.4	40.9	30.5	42.0	42.5	33.6	41.9	41.8	31.3	42.4	41.7
Prompt Tuning$^\diamond$	–	–	–	–	–	–	–	–	–	47.6	57.2	59.5
Experience Replay	34.9	46.3	51.0	39.3	48.1	51.5	34.9	46.5	46.3	36.4	47.0	49.6
Per-task Prompts$^\diamond$	50.6	62.4	67.2	50.6	62.4	67.2	50.6	62.4	67.2	50.6	62.4	67.2
IDBR	39.7	48.4	52.3	37.9	46.6	54.1	32.9	48.8	50.1	36.8	47.9	52.2
ProgPrompt$^\diamond$	**55.3**	**67.9**	**68.9**	53.3	65.8	**70.0**	51.9	**66.9**	**69.0**	53.5	**66.9**	**69.3**
ProgPrompt$^{\diamond,*}$	51.4	66.8	68.4	**54.2**	**66.9**	67.5	**57.1**	66.4	68.7	**54.2**	66.7	68.2
InfoComp (Ours)$^\diamond$	**56.4**†	**69.6**†	**70.4**†	**57.7**†	**69.2**†	**72.4**†	**59.7**†	**69.9**†	**70.8**†	**57.9**†	**69.6**†	**71.2**†
Improve	⇑**1.1**	⇑**1.7**	⇑**1.5**	⇑**3.5**	⇑**2.3**	⇑**2.4**	⇑**2.6**	⇑**3.0**	⇑**1.8**	⇑**3.7**	⇑**2.7**	⇑**1.9**

Table 5
The ablation study of our InfoComp under the task sequences Order6 and Order7. In comparison to our InfoComp method, all ablation variants show a significant decline in CTC performance, confirming the necessity of each component for collaboratively addressing CTC. **Bold** denotes the best results.

Variants	Order6			Order7		
	20	200	1000	20	200	1000
InfoComp w/o P-Prompt	48.2	57.9	60.3	49.1	58.2	61.1
InfoComp w/o S-Prompt	51.2	63.0	68.7	52.4	64.1	68.1
InfoComp w/o $\mathcal{L}_k^{\text{p-info}}$&$\mathcal{L}_k^{\text{s-info}}$	54.4	65.7	70.2	57.3	66.1	68.8
InfoComp w/o $\mathcal{L}_k^{\text{p-info}}$ in Eq. (5)	56.5	67.6	71.3	59.0	68.6	70.0
InfoComp w/o $\mathcal{L}_k^{\text{s-info}}$ in Eq. (7)	55.6	67.0	70.7	58.2	67.4	69.5
InfoComp (Ours)	**57.7**	**69.2**	**72.4**	**59.7**	**69.9**	**70.8**

its effectiveness in addressing the challenges of continual learning. Future work could investigate the application of our model-agnostic InfoComp method with additional PLMs beyond BERT, allowing for further validation of its efficacy across a broader range of CTC benchmarks.

CRediT authorship contribution statement

Duzhen Zhang: Writing – original draft, Software, Methodology. **Yong Ren:** Validation, Software. **Chenxing Li:** Resources, Methodology, Investigation. **Dong Yu:** Supervision, Resources. **Tielin Zhang:** Writing – review & editing, Supervision.

Funding

This work was supported by the Beijing Nova Program (Grant No. 20230484369), Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX), the Youth Innovation Promotion Association CAS, and the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDB1010302).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Arani, E., Sarfraz, F., & Zonooz, B. (2021). Learning fast, learning slow: A general continual learning method based on complementary learning system. In *International conference on learning representations*.

Bai, Y., Liu, Y., & Li, Y. (2024). Learning frequency-aware cross-modal interaction for multimodal fake news detection. *IEEE Transactions on Computational Social Systems*.

Bohao, P., Tian, Z., Liu, S., Yang, M.-C., & Jia, J. (2024). Scalable language model with generalized continual learning. In *ICLR 2024*.

Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C., & Mercer, R. L. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, *18*(4), 467–480.

Cha, H., Lee, J., & Shin, J. (2021). Co^{2}L: Contrastive continual learning. In *2021 IEEE/CVF international conference on computer vision* (pp. 9496–9505).

Chaudhry, A., Ranzato, M., Rohrbach, M., & Elhoseiny, M. (2018). Efficient lifelong learning with A-GEM. In *International conference on learning representations*.

Chen, X., & He, K. (2021). Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15750–15758).

Chi, Z., Dong, L., Wei, F., Yang, N., Singhal, S., Wang, W., et al. (2021). Infoxlm: An information-theoretic framework for cross-lingual language model pre-training. In *Proceedings of the 2021 conference of the North American chapter of the Association for Computational Linguistics: human language technologies* (pp. 3576–3588).

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

de Masson D'Autume, C., Ruder, S., Kong, L., & Yogatama, D. (2019). Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, *32*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. 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)* (pp. 4171–4186).

Farajtabar, M., Azizan, N., Mott, A., & Li, A. (2020). Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics* (pp. 3762–3773). PMLR.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., & Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. arXiv preprint arXiv:1312.6211.

Gu, Y., Han, X., Liu, Z., & Huang, M. (2022). PPT: Pre-trained prompt tuning for few-shot learning. In *Proceedings of the 60th annual meeting of the Association for Computational Linguistics (volume 1: long papers)* (pp. 8410–8423).

Guo, Y., Xu, S., Fu, J., Liu, J., Dong, C., & Wang, B. (2024). Q-Tuning: Queue-based prompt tuning for lifelong few-shot language learning. In *Findings of the Association for Computational Linguistics: NAACL 2024* (pp. 2595–2622).

Hou, S., Pan, X., Loy, C. C., Wang, Z., & Lin, D. (2019). Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 831–839).

Huang, Y., Zhang, Y., Chen, J., Wang, X., & Yang, D. (2021). Continual learning for text classification with information disentanglement based regularization. In *Proceedings of the 2021 conference of the North American chapter of the Association for Computational Linguistics: human language technologies* (pp. 2736–2746).

Ji, B., Zhang, T., Zou, Y., Hu, B., & Shen, S. (2022). Increasing visual awareness in multimodal neural machine translation from an information theoretic perspective. In *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 6755–6764).

Ju, J., Liu, M., Koh, H. Y., Jin, Y., Du, L., & Pan, S. (2021). Leveraging information bottleneck for scientific document summarization. In *Empirical methods in natural language processing 2021* (pp. 4091–4098). Association for Computational Linguistics (ACL).

Karimi Mahabadi, R., Henderson, J., & Ruder, S. (2021). Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, *34*, 1022–1035.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, *114*(13), 3521–3526.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 388–395).

Kumaran, D., Hassabis, D., & McClelland, J. L. (2016). What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends in Cognitive Sciences*, *20*(7), 512–534.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 3045–3059).

Li, Z., & Hoiem, D. (2017). Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(12), 2935–2947.

Li, X. L., & Liang, P. (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)* (pp. 4582–4597).

Liang, Z., Wei, F., Jie, Y., Qian, Y., Hao, Z., & Han, B. (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)* (pp. 277–292).

Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., et al. (2022). P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th annual meeting of the Association for Computational Linguistics (volume 2: short papers)* (pp. 61–68).

Liu, Y., Xiong, Z., Yuan, Y., & Wang, Q. (2023). Transcending pixels: boosting saliency detection via scene understanding from aerial imagery. *IEEE Transactions on Geoscience and Remote Sensing*, *61*, 1–16.

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, *55*(9), 1–35.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: human language technologies* (pp. 142–150).

Mahabadi, R. K., Belinkov, Y., & Henderson, J. (2021). Variational information bottleneck for effective low-resource fine-tuning. arXiv preprint arXiv:2106.05469.

Manning, C., & Schutze, H. (1999). *Foundations of statistical natural language processing*. MIT Press.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*(3), 419.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, *24*, 109–165.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*.

Pham, Q., Liu, C., & Hoi, S. (2021). Dualnet: Continual learning, fast and slow. *Advances in Neural Information Processing Systems*, *34*, 16131–16144.

Qin, C., & Joty, S. (2022). LFPT5: A unified framework for lifelong few-shot language learning based on prompt tuning of T5. In *International conference on learning representations*.

Rao, D., Visin, F., Rusu, A., Pascanu, R., Teh, Y. W., & Hadsell, R. (2019). Continual unsupervised representation learning. *Advances in Neural Information Processing Systems*, *32*.

Razdaibiedina, A., Mao, Y., Hou, R., Khabsa, M., Lewis, M., & Almahairi, A. (2023). Progressive prompts: Continual learning for language models. In *The eleventh international conference on learning representations*.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2001–2010).

Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, *7*(2), 123–146.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., et al. (2016). Progressive neural networks. arXiv preprint arXiv:1606.04671.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, *27*(3), 379–423.

Shannon, C. E. (1951). Prediction and entropy of printed English. *Bell System Technical Journal*, *30*(1), 50–64.

Shokri, R., & Shmatikov, V. (2015). Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security* (pp. 1310–1321).

Song, Y., Wang, P., Xiong, W., Zhu, D., Liu, T., Sui, Z., et al. (2023). InfoCL: Alleviating catastrophic forgetting in continual text classification from an information theoretic perspective. In *Findings of the Association for Computational Linguistics: EMNLP 2023* (pp. 14557–14570).

Sorensen, T., Robinson, J., Rytting, C. M., Shaw, A. G., Rogers, K. J., Delorey, A. P., et al. (2022). An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th annual meeting of the Association for Computational Linguistics (volume 1: long papers)* (pp. 819–862).

Steinborn, V., Dufter, P., Jabbar, H., & Schuetze, H. (2022). An information-theoretic approach and dataset for probing gender stereotypes in multilingual masked language models. In *Findings of the Association for Computational Linguistics: NAACL 2022* (pp. 921–932).

Tishby, N., Pereira, F. C., & Bialek, W. (2000). The information bottleneck method. arXiv preprint physics/0004057.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wang, A. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461.

Wang, Z., Liu, Y., Ji, T., Wang, X., Wu, Y., Jiang, C., et al. (2023). 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)* (pp. 10933–10946).

Wang, Z., Mehta, S. V., Poczós, B., & Carbonell, J. G. (2020). Efficient meta lifelong-learning with limited memory. In *Proceedings of the 2020 conference on empirical methods in natural language processing* (pp. 535–548).

Wang, Z., Panda, R., Karlinsky, L., Feris, R., Sun, H., & Kim, Y. (2023). Multitask prompt tuning enables parameter-efficient transfer learning. In *The eleventh international conference on learning representations*.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., et al. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, *32*.

Wang, B., Wang, S., Cheng, Y., Gan, Z., Jia, R., Li, B., et al. (2021). InfoBERT: Improving robustness of language models from an information theoretic perspective. In *9th international conference on learning representations*.

Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.-Y., et al. (2022). Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision* (pp. 631–648). Springer.

Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., et al. (2022). Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 139–149).

Wang, L., Zhang, X., Su, H., & Zhu, J. (2023). A comprehensive survey of continual learning: Theory, method and application. http://dx.doi.org/10.48550/ARXIV.2302.00487, CoRR abs/2302.00487, arXiv:2302.00487.

Wei, L., Tan, Z., Li, C., Wang, J., & Huang, W. (2024). Diff-eRank: A novel rank-based metric for evaluating large language models. arXiv preprint arXiv:2401.17139.

West, P., Holtzman, A., Buys, J., & Choi, Y. (2019). Bottlesum: Unsupervised and self-supervised sentence summarization using the information bottleneck principle. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 3750–3759).

Wolf, T. (2019). Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

Wu, J., Yu, T., Wang, R., Song, Z., Zhang, R., Zhao, H., et al. (2023). InfoPrompt: Information-theoretic soft prompt tuning for natural language understanding. arXiv preprint arXiv:2306.04933.

Yoon, J., Yang, E., Lee, J., & Hwang, S. J. (2018). Lifelong learning with dynamically expandable networks. In *International conference on learning representations*.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, *28*.

Zhu, Q., Li, B., Mi, F., Zhu, X., & Huang, M. (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)* (pp. 1124–1137).