Fed-CPrompt: Contrastive Prompt for Rehearsal-Free Federated Continual Learning

Gaurav Bagwe [1] **Xiaoyong Yuan** [2] **Miao Pan** [3] **Lan Zhang** [1]

Abstract

Federated continual learning (FCL) learns incremental tasks over time from confidential datasets distributed across clients. This paper focuses on rehearsal-free FCL, which has severe forgetting issues when learning new tasks due to the lack of access to historical task data. To address this issue, we propose Fed-CPrompt based on prompt learning techniques to obtain task-specific prompts in a communication-efficient way. Fed-CPrompt introduces two key components, asynchronous prompt learning, and contrastive continual loss, to handle asynchronous task arrival and heterogeneous data distributions in FCL, respectively. Extensive experiments demonstrate the effectiveness of Fed-CPrompt in achieving SOTA rehearsal-free FCL performance.

1. Introduction

Federated learning (FL) has been a popular collaborative machine learning paradigm enabling multiple clients to learn a shared model without exposing private client data (McMahan et al., 2017). While successful, existing FL algorithms are mainly designed for a single task with fixed datasets on clients (McMahan et al., 2017; Li et al., 2020; 2021), which becomes ineffective in handling non-stationary data distribution over time. Therefore, recent efforts have been put into federated continual learning (FCL) to learn tasks that are presented sequentially. Since the model in continual learning (CL) may overfit data from the current task and suffer from catastrophic forgetting (Kirkpatrick et al., 2017), the mainstream research to address the forgetting issue can be roughly divided into two categories: rehearsal-based and rehearsal-free FCL.

Although rehearsal-based approaches achieve state-of-the-art (SOTA) performance by using the rehearsal buffer to store and retrain data from previous tasks, the buffer size needs to be large enough to effectively mitigate forgetting (Wang et al., 2022b; Dong et al., 2022), leading to scalability and data storage constraints in FL. Moreover, many applications do not allow this buffer due to privacy concerns (Smith et al., 2022b), further restricting their adoption in practice. Hence, this work focuses on rehearsal-free FCL. Existing efforts along this line regularize the global model with knowledge from previous tasks when learning a new task (Shoham et al., 2019; Yoon et al., 2021; Casado et al., 2022; Usmanova et al., 2021; 2022; Ma et al., 2022). Unfortunately, they have substantially deteriorated performance compared to rehearsal-based approaches (Wang et al., 2022b). Moreover, existing research requires continuously exchanging the entire model to learn incremental tasks in FCL, leading to significant communication overhead. In view of these, it is critical to developing *innovative rehearsal-free FCL in a communication-efficient way to address the forgetting issue while maintaining the model plasticity for new tasks*.

Enlightened by the recent advance of prompting techniques (Lester et al., 2021; Liu et al., 2022; Li & Liang, 2021), in this work, we leverage prompt learning to achieve the above goal. As one promising transfer learning approach, prompt learning uses insertable embeddings called prompts to condition a pre-trained model for downstream tasks. Recent research enables prompt-based CL by using key-query mechanisms, which achieves SOTA rehearsal-free performance, even outperforming rehearsal-based CL (Wang et al., 2022b;a; Smith et al., 2022a). Due to the small size of prompt parameters, the communication efficiency of FCL is expected to be improved significantly. However, existing prompt-based CL is designed for centralized datasets, which becomes ineffective in FL with distributed and confidential datasets. The main limitation is due to the inherent heterogeneity of distributed clients. On the one hand, clients may observe heterogeneous data for the same task, leading to biased learning performance and slow convergence. On the other hand, incremental tasks may arrive asynchronously on clients, further deteriorating the overall learning perfor-

[1]Department of ECE, Michigan Technological University, Houghton, MI, USA [2]College of Computing, Michigan Technological University, Houghton, MI, USA [3]Department of ECE, University of Houston, Houston, TX, USA. Correspondence to: Gaurav Bagwe <grbagwe@mtu.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

mance. Therefore, to unleash the potential of prompting for rehearsal-free FCL, we propose Fed-CPrompt to facilitate inter-task and inter-client prompt-based knowledge transfer while addressing the heterogeneity concerns of data distribution and task arrival over clients. Our key contributions are summarized below:

- We propose Fed-CPrompt, an innovative rehearsal-free FCL framework based on prompting techniques. Fed-CPrompt achieves SOTA FCL performance to handle the stability-plasticity dilemma under heterogeneous FL environments in a communication-efficient way.
- We introduce two key components to Fed-CPrompt: *asynchronous prompt learning* takes advantage of task asynchronicity to strengthen the task-specific prompts; *C2L loss* alleviates inter-task forgetting and inter-client data heterogeneity via a contrastive and continual loss.
- We conduct extensive experiments to demonstrate the effectiveness of Fed-CPrompt in various challenging FCL settings, such as heterogeneous data distribution and asynchronous task arrival.

2. Proposed Method

2.1. Problem Statement

In a standard FCL setting, a central server coordinates a set of distributed clients $\mathcal{C}$ to learn incremental tasks $\mathcal{T}_1, \ldots, \mathcal{T}_n$ over time. The training data for each task is distributed to clients and cannot be shared. FCL aims to obtain a global model parameterized by $\mathbf{w}$ to perform all existing tasks. In this work, we consider a challenging CL problem, class-incremental CL, where the task labels are unknown during inference (Dong et al., 2022). Our design can be easily extended to the task- or domain-incremental FCL problems. The optimization objective can be written as

$$\min_{\mathbf{w}} \sum_{i\in\{1,\ldots,n\}} \sum_{c\in\mathcal{C}} \frac{\mathrm{n}_c^{\mathcal{T}_i}}{\mathrm{n}^{\mathcal{T}_i}} \mathcal{L}(\mathcal{D}_c^{\mathcal{T}_i}; \mathbf{w}), \tag{1}$$

where $\mathrm{n}_c^{\mathcal{T}_i}$ and $\mathrm{n}^{\mathcal{T}_i}$ represent the number of training samples from client c and all clients for task $\mathcal{T}_i$, respectively. $\mathcal{D}_c^{\mathcal{T}_i}$ is the training dataset of $\mathcal{T}_i$ on client c.

This objective function uses data from all existing tasks, making it a rehearsal-based FCL problem. This work focuses on rehearsal-free FCL. Specifically, each client can only observe the training data of the current task, i.e., when training on task $\mathcal{T}_n$, training data of all previous tasks are unseen. However, due to the unavailability of historical task data, training the current task can overwrite previous task information of the model $\mathbf{w}$ in (1), deteriorating the forgetting issues in CL. Thus, existing rehearsal-free FCL approaches cannot achieve comparable performance to rehearsal-based approaches (Wang et al., 2023).

2.2. Design Principle

In this work, we aim to accommodate the forgetting issue for rehearsal-free FCL. Inspired by the success of the prompt-based rehearsal-free CL that achieves SOTA performance, we intend to implement prompting techniques in our design. Existing prompt-based CL (Smith et al., 2022a; Wang et al., 2022b;a) use insertable embeddings, called prompts p, to condition a frozen pre-trained model θ to perform incremental tasks. Due to the small size of prompt parameters, a task-specific prompt is created and stored for each task to avoid overwriting previous knowledge. Here, we refer readers to Appendix A for more details. While successful, the above prompt-based CL research is designed for centralized datasets, which becomes ineffective in FL settings.

The main challenge of implementing prompting techniques in FCL is the inherent heterogeneity of distributed clients. On the one hand, the data heterogeneity among clients leads to biased local updates and slow convergence. On the other hand, the sequential tasks may appear asynchronously over clients, further delaying convergence. Due to the small size of learnable parameters in prompt learning, it is essential to improve their learning capacity by facilitating knowledge transfer between tasks and clients. Therefore, we propose Fed-CPrompt, an innovative prompt-based rehearsal-free FCL framework. As shown in Figure 1, Fed-CPrompt introduces two key components, asynchronous prompt learning and contrastive and continual loss, to address the aforementioned task arrival and data heterogeneity concerns. In the following, we first introduce these two components and then present the overall training of Fed-CPrompt.

2.3. Asynchronous Prompt Learning

We adopt the existing prompt-based CL approach (CODA-P (Smith et al., 2022a)) on clients to learn incremental tasks based on their local data. In CODA-P, the prompt for the current task is re-weighted based on previous task information to refine task-specific representation via attention mechanisms (see Appendix A). In Fed-CPrompt, when client $c \in \mathcal{C}$ learns task $\mathcal{T}_m$, $p_c^{\mathcal{T}_m} = \sum_{i\in[1,m-1]} \alpha_s^{\mathcal{T}_i} P_s^{\mathcal{T}_i} + \alpha_c^{\mathcal{T}_m} P_c^{\mathcal{T}_m}$, where $\alpha_b^{\mathcal{T}_i}$ and $P_b^{\mathcal{T}_i}$ are the $\mathcal{T}_i$-specific attention and prompt at the server ($b = s$) and client c ($b = c$), respectively. The updated client-side $p_c^{\mathcal{T}_m}$ will be uploaded to the server and aggregated based on classical FL (McMahan et al., 2017) to obtain server-side prompt $p_s^{\mathcal{T}_m}$. However, such naive aggregation becomes inefficient in asynchronous task arrival. When client c is training task $\mathcal{T}_m$, the latest task observed by other clients might be task $\mathcal{T}_n$ ($m < n$), and $\mathcal{T}_n$ will be observed by client c later. Hence, to handle this condition, Fed-CPrompt introduces asynchronous prompt learning.

Instead of waiting for updated prompts of the current task $\mathcal{T}_n$ from all clients before aggregation, we allow task-specific

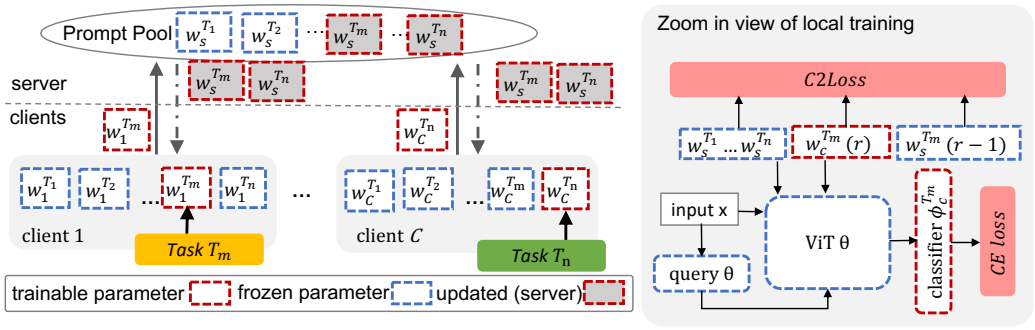


Figure 1. Overview of Fed-CPrompt. Left: overall system framework. A server maintains a pool of task-specific prompts to coordinate multiple clients for FCL. The clients may observe different tasks at the same time. Fed-CPrompt allows asynchronous prompt learning. Right: zoom in view of client c's local training for task $\mathcal{T}_m$. Since only the trainable parameters ($w_c^{\mathcal{T}_m}$ and $\phi_c^{\mathcal{T}_m}$) need to be exchanged, Fed-CPrompt is a communication-efficient FCL approach.

prompt aggregation in parallel. In this way, the previously learned prompt at the server $p_s^{\mathcal{T}_m}$ can be refined by $p_c^{\mathcal{T}_m}$. Moreover, taking advantage of the task arrival heterogeneity, the training of $p_c^{\mathcal{T}_m}$ becomes

$$p_c^{\mathcal{T}_m} = \sum_{i=1}^{m-1} \alpha_s^{\mathcal{T}_i} P_s^{\mathcal{T}_i} + \alpha_c^{\mathcal{T}_m} P_c^{\mathcal{T}_m} + \sum_{j=m+1}^{n} \alpha_s^{\mathcal{T}_j} P_s^{\mathcal{T}_j}, \quad (2)$$

where the first and the third terms are task knowledge from the server, which are frozen when training task $\mathcal{T}_m$. It should be mentioned that although the newest task for client c is $\mathcal{T}_m$, the asynchronous task arrival in FCL allows client c to leverage unseen task knowledge to navigate the local training. By incorporating past and future task representations, we increase the capacity of prompts to learn task-specific instructions.

2.4. C2Loss: Contrastive and Continual Loss

To address the data heterogeneity issue while alleviating forgetting in FCL, we introduce a new loss function, contrastive and continual loss (C2Loss), to regularize local training on clients. The goal of C2Loss is mainly twofold. First, C2Loss accommodates disagreements between clients due to biased local training with heterogeneous data distribution. Second, C2Loss enforces distinct task-specific prompts construction, which facilitates CL to avoid the forgetting effect. Specifically, when learning task $\mathcal{T}_m$ at communication round r, we have the C2Loss on client $c \in \mathcal{C}$ given by

$$\mathcal{L}_{C2L}(P_c^{\mathcal{T}_m}(r)) = \max\big(||P_c^{\mathcal{T}_m}(r) - P_s^{\mathcal{T}_m}(r-1)||_2 \quad (3)$$
$$-\gamma \min\{||P_c^{\mathcal{T}_m}(r) - P_s^{\mathcal{T}_i}||_2, i \in [1,n], i \neq m\} + \alpha, 0\big),$$

where the first term within the $\max()$ calculates the change of the current prompt compared to that in the previous round. By restricting this change, C2Loss smooths the local update to achieve the first goal. The second term within the $\max()$ finds the most similar prompt to the current prompt based on the distance between the current and all previous prompts. By increasing this distance, C2Loss enforces the discrimination between task-specific prompts to achieve the second goal. Besides, $\gamma > 0$ is the hyperparameter to balance the impact between the first two terms. $\alpha \in [0,1]$ represents a margin value that encourages a separation between the first two terms (Schroff et al., 2015).

2.5. Overall Training

In Fed-CPrompt, client $c \in \mathcal{C}$ conducts local training with dataset $\mathcal{D}_c^{\mathcal{T}_m}$ for the current task $\mathcal{T}_m$. As discussed in (2), a prompt is constructed based on attention mechanisms, and thus the learnable prompt parameter for client c is defined by $\mathbf{w}_c^{\mathcal{T}_m} = \{P_c^{\mathcal{T}_m}, K_c^{\mathcal{T}_m}, A_c^{\mathcal{T}_m}\}$ (K and A composite the α in (2), detailed in Appendix A). By incorporating the C2Loss to the cross-entropy loss, we have the local optimization function of client c by

$$\min_{\mathbf{w}_c^{\mathcal{T}_m}, \phi_c^{\mathcal{T}_m}} \mathcal{L}_{CE}(f_{\phi_c}(x;\theta,\mathbf{w}_c), y) + \lambda\, \mathcal{L}_{C2L}(\mathbf{w}_c^{\mathcal{T}_m}), \quad (4)$$

where $\phi_c^{\mathcal{T}_m}$ represents the classifier parameter for task $\mathcal{T}_m$. Note that $\mathbf{w}_c$ and ϕ_c concatenate both the frozen previous task parameter and the current task learnable parameter as discussed in (2). Besides, θ is the frozen pretrained model parameters; $(x,y) \in \mathcal{D}_c^{\mathcal{T}_m}$; $\lambda \in [0,1]$ is the hyperparameter balancing losses. Both prompt parameters $\mathbf{w}_c^{\mathcal{T}_m}$ and classifier parameters $\phi_c^{\mathcal{T}_m}$ will be uploaded to the server. The server handles asynchronous task arrival by conducting parallel aggregation following classical FL (McMahan et al., 2017). The overall training is illustrated in Algorithm 1 of Appendix C.

3. Experiments

3.1. Experimental Setup

We evaluate the proposed Fed-CPrompt based on the CIFAR-100 dataset (Krizhevsky et al., 2009), a widely used dataset

Table 1. Performance comparison of rehearsal-free FCL methods under iid settings.

Model	Accuracy	Forgetting
Fed-EWC	9.79	84.85
Fed-LwF	60.92	33.75
Fed-L2P	73.19	9.80
Fed-DualP	76.00	9.84
Fed-CODAP	77.28	6.42
Fed-CPrompt	**79.43**	**4.75**

Table 2. Performance comparison of prompt-based FCL methods under different non-iid settings (label and quantity skew).

	Model	Accuracy	Forgetting
Label Skew	Fed-DualP	46.28	13.16
	Fed-CODAP	54.80	11.55
	Fed-CPrompt	**65.45**	**9.15**
Quantity Skew	Fed-DualP	77.57	8.38
	Fed-CODAP	78.56	**5.00**
	Fed-CPrompt	**81.12**	7.75

in continual learning for classification tasks. We consider a total of 10 clients in FCL. The server-side knowledge aggregation is based on FedAvg (McMahan et al., 2017). The evaluation metrics include average accuracy and average forgetting, which are standard metrics used in previous CL research (Wang et al., 2023; Huang et al., 2022). To comprehensively evaluate Fed-CPrompt, we consider baseline approaches, including rehearsal-free FL approaches (i.e., Fed-EWC and Fed-LWF) and recent prompt-based CL approaches (i.e., Fed-CODAP, Fed-DualP, Fed-L2P). Further details on the dataset setup, FL settings, evaluation metrics, and baseline approaches can be found in Appendix D.

3.2. Experimental Results

Effectiveness of Fed-CPrompt. We evaluate the effectiveness of Fed-CPrompt under iid and non-iid FL settings. We report the average test accuracy and forgetting over all ten tasks. As illustrated in Table 1, Fed-CPrompt gains a significant performance improvement over all rehearsal-free FCL methods under iid settings. Compared with the best of existing works, Fed-CPrompt achieves around a 2% increase in Top-1 accuracy and around a 2% drop in Forgetting. It should be mentioned that non-prompt-based methods (Fed-EWC and Fed-LwF) optimize about 86 million parameters, while Fed-CPrompt optimizes only 4 million ($\approx 4.18\%$) to achieve better performance. Moreover, Fed-CPrompt has better convergence, which significantly reduces the communication cost for FCL.

Besides, we further compare the Fed-CPrompt with other prompt-based FCL baselines under non-iid settings. In the experiments following (Li et al., 2020), we consider two non-iid settings: label skew and quantity skew. As illustrated in Table 2, Fed-CPrompt outperforms the existing prompt-based methods under non-iid settings. In particular, under a challenging label-skew setting, Fed-CPrompt achieves a significant performance improvement by 10.65%.

Table 3. Performance comparison of asynchronous continual learning under iid and non-iid (label skew) settings.

	Model	Accuracy	Forgetting
iid	Fed-DualP	70.27	17.36
	Fed-CODAP	73.63	10.94
	Fed-CPrompt	**75.70**	**9.82**
non-iid	Fed-DualP	46.80	**3.89**
	Fed-CODAP	50.81	8.05
	Fed-CPrompt	**62.60**	10.55

Table 4. Impact of C2Loss under the non-iid (label skew) setting.

Model	Accuracy	Forgetting
$\mathcal{L}_{CE}$ w/ FedProx	50.81	9.16
$\mathcal{L}_{CE}$ w/o $\mathcal{L}_{C2L}$	71.30	13.52
$\mathcal{L}_{CE}$ w/ $\mathcal{L}_{C2L}$	**79.30**	**4.50**

Impact of Asynchronous Continual Learning Tasks. We demonstrate the effectiveness of Fed-CPrompt under asynchronous task arrival, where the clients train the models on different tasks at the same time. As illustrated in Table 3, the average test accuracy of Fed-CPrompt significantly outperforms the existing methods by 2.07% and 11.79% under iid and non-iid settings, respectively. Our findings suggest jointly considering past and future task information can improve the training efficiency of FCL. It should be noted that the forgetting of Fed-CPrompt is comparable to or higher than the existing works; however this is due to the high accuracy gained by Fed-CPrompt on the first task. Besides the impact of high accuracy on the first task, we can still observe the substantial advantage of Fed-CPrompt in mitigating catastrophic forgetting, as the average accuracy on all ten tasks achieved by Fed-CPrompt is much higher than the existing works.

Impact of C2Loss. We further perform ablation studies to evaluate the effectiveness of the proposed C2Loss. We compare the performance between FedProx and Fed-CPrompt with and without C2Loss. As shown in Table 4, Fed-CPrompt with C2Loss achieves the highest accuracy and lowest forgetting compared with the rest two methods. This is mainly due to that C2Loss handles inter-task and inter-client knowledge transfer, thereby leading to better task discrimination and improved accuracy.

4. Conclusion

This paper proposed Fed-CPrompt, an innovative rehearsal-free FCL framework to alleviate catastrophic forgetting over

incremental tasks and facilitate knowledge transfer among distributed and heterogeneous clients. Fed-CPrompt introduces two key components: asynchronous prompt learning to handle asynchronous arrival, and a simple yet effective contrastive continual loss that optimizes prompt parameters while providing additional supervision for learning distinct task-specific prompts. Extensive experiments demonstrate the effectiveness of our proposal.

Acknowledgment

The authors thank all anonymous reviewers for their insightful feedback. This work was supported by the National Science Foundation under Grants CCF-2106754, CCF-2221741, CCF-2153381, and CCF-2151238. The work of Miao Pan was supported in part by the US National Science Foundation under Grants CNS-2107057 and CNS-2318664.

References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Casado, F. E., Lema, D., Criado, M. F., Iglesias, R., Regueiro, C. V., and Barro, S. Concept drift detection and adaptation for federated and continual learning. *Multimedia Tools and Applications*, pp. 1–23, 2022.

Dong, J., Wang, L., Fang, Z., Sun, G., Xu, S., Wang, X., and Zhu, Q. Federated class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10164–10173, 2022.

Guo, T., Guo, S., Wang, J., and Xu, W. Promptfl: Let federated participants cooperatively learn prompts instead of models–federated learning in age of foundation model. *arXiv preprint arXiv:2208.11625*, 2022.

Huang, W., Ye, M., and Du, B. Learn from others and be yourself in heterogeneous federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10143–10153, 2022.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Li, Q., He, B., and Song, D. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10713–10722, 2021.

Li, Q., Diao, Y., Chen, Q., and He, B. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 965–978. IEEE, 2022.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

Li, X., Zhou, Y., Wu, T., Socher, R., and Xiong, C. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pp. 3925–3934. PMLR, 2019.

Li, X. L. and Liang, P. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Liu, X., Ji, K., Fu, Y., Tam, W., Du, Z., Yang, Z., and Tang, J. 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, 2022.

Ma, Y., Xie, Z., Wang, J., Chen, K., and Shou, L. Continual federated learning based on knowledge distillation. In Raedt, L. D. (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2182–2188. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/303. URL https://doi.org/10.24963/ijcai.2022/303. Main Track.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823, 2015.

Shenaj, D., Toldo, M., Rigon, A., and Zanuttigh, P. Asynchronous federated continual learning. *arXiv preprint arXiv:2304.03626*, 2023.

Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., and Singh, S. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

Shoham, N., Avidor, T., Keren, A., Israel, N., Benditkis, D., Mor-Yosef, L., and Zeitak, I. Overcoming forgetting in federated learning on non-iid data. *arXiv preprint arXiv:1910.07796*, 2019.

Smith, J. S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., and Kira, Z. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2211.13218*, 2022a.

Smith, J. S., Tian, J., Hsu, Y.-C., and Kira, Z. A closer look at rehearsal-free continual learning. *arXiv preprint arXiv:2203.17269*, 2022b.

Usmanova, A., Portet, F., Lalanda, P., and Vega, G. A distillation-based approach integrating continual learning and federated learning for pervasive services. *arXiv preprint arXiv:2109.04197*, 2021.

Usmanova, A., Portet, F., Lalanda, P., and Vega, G. Federated continual learning through distillation in pervasive computing. In *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 86–91. IEEE, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.-Y., Ren, X., Su, G., Perot, V., Dy, J., and Pfister, T. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pp. 631–648. Springer, 2022a.

Wang, Z., Zhang, Z., Lee, C.-Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., and Pfister, T. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022b.

Wang, Z., Zhang, Y., Xu, X., Fu, Z., Yang, H., and Du, W. Federated probability memory recall for federated continual learning. *Information Sciences*, 2023.

Yoon, J., Jeong, W., Lee, G., Yang, E., and Hwang, S. J. Federated continual learning with weighted inter-client transfer. In *International Conference on Machine Learning*, pp. 12073–12086. PMLR, 2021.

Zhao, H., Du, W., Li, F., Li, P., and Liu, G. Reduce communication costs and preserve privacy: Prompt tuning method in federated learning. *arXiv preprint arXiv:2208.12268*, 2022.

Zizzo, G., Rawat, A., Holohan, N., and Tirupathi, S. Federated continual learning with differentially private data sharing. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

A. Preliminaries for Prompt-based Continual Learning

In this work, we build upon the technical foundations of prompt-based methods from the prior centralized continual learning research (Smith et al., 2022a; Wang et al., 2022a;b) to introduce prompts, which can collaboratively learn in heterogeneous federated settings. As done in CODA-P, prompt parameters are attached to several multi-head self-attention (MSA) layers in a pre-trained ViT. Define a task-specific prompt parameter for task $\mathcal{T}_m$ as $P^{\mathcal{T}_m} \in \mathbb{R}^{L_P \times D \times \mathcal{T}_m}$, where L_P, D, and $\mathcal{T}_m$ are the prompt lengths, embedding dimension, and the number of prompts for each task, respectively. We consider prefix-tuning to attach prompts to the keys and values of an MSA layer with input $h \in \mathbb{R}^{L \times D}$ and the query, key, and value as h_Q, h_K, and h_V. A prompt p is split into $\{P_K, P_V\} \in \mathbb{R}^{\frac{L_p}{2} \times D}$, which are respectively attached to the key and the value of this layer, i.e., $MSA(h_Q, [P_K; h_K], [P_V; h_V])$. where $[\cdot;\cdot]$ is a concatenation operation. Since CODA-P achieves SOTA centralized continual learning performance, we adopt the weighted prompt for local training, and the prompt for task $\mathcal{T}_m$ can be calculated by

$$p^{\mathcal{T}_m} = \sum_{i \in [1,m]} \alpha^{\mathcal{T}_i} P^{\mathcal{T}_i}, \tag{5}$$

where $P^{\mathcal{T}_m}$ is a learnable prompt to the current task $\mathcal{T}_m$, $\alpha^{\mathcal{T}_m} = \gamma(q(x) \odot A^{\mathcal{T}_m}, K^{\mathcal{T}_m})$ measures the cosine similarity γ between the attended query and the key, where the attended query defined by the element-wise product $\odot$ between query and learnable attention parameter. The query is produced as $q(x) \in \mathbb{R}^D = f(x;\theta)$, where $f(\cdot;\theta)$ is the encoder of the pre-trained ViT[1]. For training task $\mathcal{T}_m$, the learnable parameters include $P^{\mathcal{T}_m}$, $K^{\mathcal{T}_m}$, $A^{\mathcal{T}_m}$, and the classification head $\phi^{\mathcal{T}_m}$, whereas $(\alpha^{\mathcal{T}_i}, P^{\mathcal{T}_i})\ \forall i \in [1, m-1]$ is frozen but contributes to the training as in Equation (5). In addition, the classification head of the previous task $\mathcal{T}_1, \cdots, \mathcal{T}_{m-1}$, *i.e* are frozen.

B. Related Work

Federated Continual Learning (FCL). FCL performs addresses catastrophic forgetting across multiple clients trained on their private sequential tasks, where a global model is obtained by exchanging task-specific knowledge via a global server. The mainstream FCL research can be roughly divided into two categories: rehearsal-based and rehearsal-free FCL. The rehearsal-based research stores and replays information from previous tasks to mitigate the global model's forgetting over time (Dong et al., 2022; Huang et al., 2022; Zizzo et al., 2022; Wang et al., 2023). For example, Huang *et al.* proposed FCCL to address the heterogeneity and catastrophic forgetting in federated learning based on buffered data for intra- and inter-domain knowledge distillation (Huang et al., 2022). Similarly, Zizzo *et al.* and Wang *et al.* leveraged replay buffers and novel data-sharing approaches based on differential privacy to mitigate forgetting (Zizzo et al., 2022; Wang et al., 2023). To tackle the global model's forgetting brought by heterogeneous clients, Dong *et al.* introduced a proxy server to store and select the best old models to assist clients' local training (Dong et al., 2022). While successful, the above rehearsal-based FCL research requires large storage space and complex data-sharing strategies to replay past information, making it challenging to scale over time.

Another category of FCL research is rehearsal-free approaches without storing past information. One group of rehearsal-free continual learning (CL) expands the model architecture when encountering new tasks (Rusu et al., 2016; Li et al., 2019). However, most architecture-based approaches require task identity to condition the network during inference, leading to their ineffectiveness for class-incremental or task-agnostic CL scenarios, i.e., the task identity is unknown. In this work, we focus on the practical but more challenging class-incremental FCL in a rehearsal-free manner. Existing FCL research along this line proposed regularizing the model with respect to the previous task knowledge when training a new task. For example, Shoham *et al.* and Yoon *et al.* leveraged the weight consolidation method to restrict the updates of the important parameters regarding previous tasks while improving the training performance for the new task (Shoham et al., 2019; Yoon et al., 2021). Similarly, several recent works implemented knowledge distillation methods to transfer knowledge from the model for the old task to that for the current task (Casado et al., 2022; Usmanova et al., 2021; 2022; Ma et al., 2022). In addition, (Shenaj et al., 2023) investigates the asynchronous-task FCL while using representation loss and a modified aggregation strategy to address the forgetting across multiple clients asynchronously learning respective tasks. While the aforementioned research enables class-incremental FCL without the rehearsal buffer, they rely on optimizing the entire model on the client side, leading to heavy communication overhead when iteratively exchanging distributed client knowledge in FL, especially for CL scenarios. To address the limitations of existing research, in this work, we propose a novel rehearsal-free FCL approach for class-incremental learning problems based on prompt learning techniques.

[1]We refer the reader to sections 4.1 and 4.2 of the CODA-p (Smith et al., 2022a) paper for more details.

Prompt Learning. Prompt learning has been a popular transfer learning approach that modifies the input sample with input embedding called prompts, aiming to provide additional information to condition the model to perform downstream tasks (Brown et al., 2020; Jiang et al., 2020; Shin et al., 2020). However, designing the prompt function for various downstream tasks is challenging. Recent research has introduced "soft prompts" to automatically train the learnable prompt parameters to replace the heuristic manual selection, such as the prompt tuning, p-tuning, and prefix tuning (Lester et al., 2021; Liu et al., 2022; Li & Liang, 2021). Prompt learning has shown great potential for parameter-efficient transfer learning with a small set of prompt parameters. Taking advantage of the small parameter size, Zhao *et al.* (Zhao et al., 2022) and Guo *et al.* (Guo et al., 2022) adopted prompt learning to improve federated learning efficiency.

Some recent works have implemented prompt learning techniques in CL. Wang *et al.* proposed L2P by using the key-query-based similarity method to select prompts from a prompt pool to instruct different tasks in CL (Wang et al., 2022b). Later, DualPrompt was introduced as the follow-up to L2P with better CL performance, which learns two sets of disjoint prompt spaces to encode task-specific and task-invariant instructions, respectively (Wang et al., 2022a). More recently, CODA-Prompt was proposed using an attention-based end-to-end key-query method, which produces the input-conditioned prompts to further improve CL performance (Smith et al., 2022a). Nevertheless, the above prompt-based CL approaches designed for centralized datasets cannot be directly used for federated learning scenarios, as they ignore the unique challenges raised by distributed nature of clients, such as the heterogeneous data distribution and asynchronous task arrival over clients. To the best of our knowledge, none of the existing prompt learning research has been done for FCL.

C. Algorithm

The overall training process includes four main steps (a-d) as shown in Algorithm 1. (a) The server distributes the prompts and model to each new participating device. (b) Each user first freezes the previous prompt parameters. (c) Each user optimizes the local prompt parameters and classifier head following CE loss and Equation (3). (d) The clients return the locally trained model to the server. Further, the server aggregates the model following classical FedAvg (McMahan et al., 2017). Algorithm 1 follows steps (a) - (d) until convergence.

D. Implementation details

In this section, we conduct extensive experiments to evaluate the proposed Fed-CPrompt. We first introduce the experimental setup, followed by the experimental results. Additionally, the same random seed is used to conduct all experiments for reproducibility.

D.1. Dataset Setup.

The CIFAR-100 dataset consists of 100 classes with 600 samples per class. In the experiments, we divide the dataset into 10 disjoint tasks with 10 classes per task ($5,000$ training samples per task). We divide the samples on each task among clients following a uniform distribution for the iid settings in federated learning. We implement label-based and quantity-based distribution skew (i.e., label skew and quantity skew) for non-iid settings with non-iid degree $\beta = 0.5$ following (Li et al., 2022). The test dataset consists of $10,000$ samples, with $1,000$ samples per task.

D.2. Federated Learning Settings.

The learning rate is set to $lr = 0.0001$. We also deploy an early-stopping mechanism in each task using a validation set. We consider $\mathcal{C} = 10$ clients, $R = 40$ communication rounds, and local epochs $l_{epochs} = 5$. The network parameters are optimized using Adam optimizer and a batch size of 128 images. We split the CIFAR100 dataset into 10 tasks, each with 10 classes. This is distributed among the 10 clients following § D.1.

D.3. Asynchronous Tasks.

We consider an asynchronous scenario where different clients learn from different tasks at the same time. Specifically, we select a random set of 5 clients to participate in the following task $\mathcal{T}_n = \mathcal{T}_{m+1}$, while the remaining 5 clients remain on task $\mathcal{T}_m$.

Algorithm 1 Fed-CPrompt

Input: Set of $\mathcal{C}$ clients, R communication rounds, total tasks $\mathcal{T}$, trainable parameters $\mathbf{w}_s = \{P_s, A_s, K_s\}, \phi_s$, frozen pretrained large model parameters θ, local epochs E, learning rate η
Output: $\mathbf{w}, \phi$

Server executes:
Send the pretrained model $\phi, \theta, \mathbf{w}_s$ parameters to clients. **(a)**
for $r \in \{1, \cdots, R\}$ **do**
  for each client $c \in \mathcal{C}$ **do**
    $\mathbf{w}_c \leftarrow \mathbf{w}_s$
    $\phi_c \leftarrow \phi_s$
    $\mathbf{w_c}, \phi_c \leftarrow$ **Client Update**$(\mathbf{w_c}, \phi_c)$
  end for
  Aggregate $\mathbf{w}_c^{\mathcal{T}_m}$ and $\phi_c^{\mathcal{T}_m}$ using FedAvg **(d)**
end for
return θ_{r+1}

Client Update $(\mathbf{w_c}, \phi)$:
for $i \in [1, n]$ **do**
  if $m \neq n$ **then**
    Freeze $w_c^{\mathcal{T}_i}, \phi_c^{\mathcal{T}_i}$ **(b)**
  end if
end for
for $e \in E$ **do**
  $\mathcal{L}_{CE} \leftarrow$ Calculate *cross entropy* loss $\mathcal{L}_{CE}$
  $\mathcal{L}_{C2L} \leftarrow$ Calculate *C2Loss* using (3)
  Update $\mathbf{w}_c^{\mathcal{T}_m}, \phi_c^{\mathcal{T}_m}$ **(c)**
end for
return $\mathbf{w}_c^{\mathcal{T}_m}, \phi_c^{\mathcal{T}_m}$

D.4. Baseline Approaches.

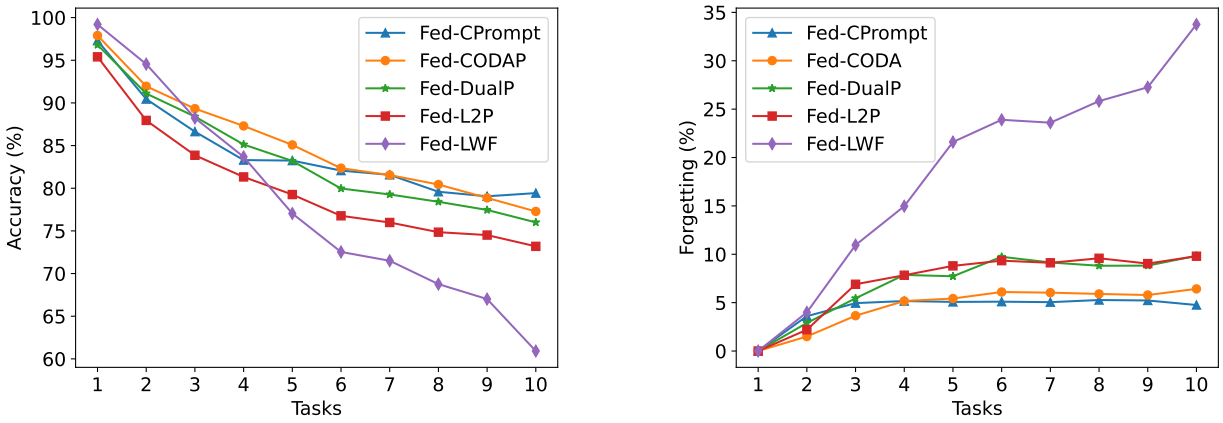
We compare our proposed Fed-CPrompt with CODA-Prompt (Smith et al., 2022a), Dual prompt (Wang et al., 2022a), L2P (Wang et al., 2022b) applied to federated settings. These prompt-based methods have shown potential parameter-efficient SOTA solutions in continual learning. Additionally, we consider conventional non-prompt-based rehearsal-free methods to demonstrate the advantages of prompt-based methods. Specifically, Fed-EWC (Shoham et al., 2019) and Fed-LWF (Usmanova et al., 2022) provide a fair representation of conventional non-prompt-based rehearsal-free methods in continual learning (Yoon et al., 2021; Casado et al., 2022; Usmanova et al., 2021). This comparison allows us to show the potential of a prompt-based approach to other rehearsal-free methods. The same set of hyper-parameters, such as the learning rate, batch size, and number of rounds is adopted in the baselines and our proposed Fed-CPrompt.

D.5. Prompt parameters.

We use prefix-tuning (Li & Liang, 2021) to attach the prompts to layers (1-5) of the pretrained ViT network (Vaswani et al., 2017). The total prompt size is $n = 100$, and 10 prompts per task. Each prompt is set with length $L_p = 8$ and embedding dimension $D = 768$.

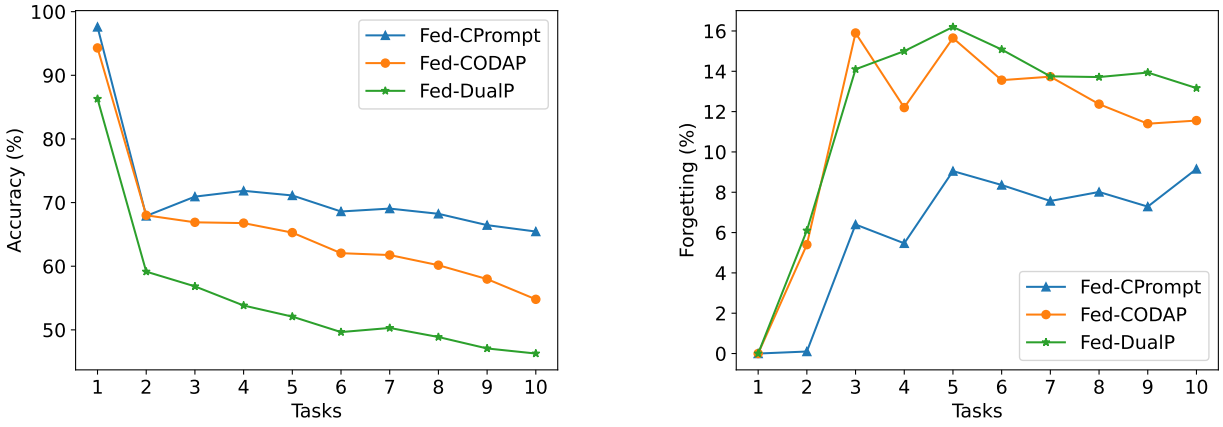
D.6. Evaluation metrics.

We evaluate our model on the standard continual learning metrics, including average accuracy and average forgetting, which are widely used in previous works (Li & Liang, 2021; Usmanova et al., 2022; Wang et al., 2023). We follow the standard definition of accuracy and forgetting mentioned in (Smith et al., 2022a).



(a) Total Test Accuracy. (b) Overall Forgetting.

Figure 2. Total test accuracy and forgetting of the global model after training each incremental task in standard iid settings. Note that the total test accuracy is the average test accuracy of the current task and all the previous tasks.



(a) Total Test Accuracy. (b) Forgetting.

Figure 3. Total test accuracy and forgetting after training each incremental task on non-iid (label-skew).

E. Additional Results

Training Efficiency. Figure 2(a) and Figure 2(b) demonstrate the effect of catastrophic forgetting when training new incremental tasks. Overall, we observe that in Fed-CPrompt retains knowledge from previous tasks, mitigating the catastrophic forgetting issues. Additionally, the accuracy per task is higher due to the increased capacity of prompts compared to prompts used in Dual Prompt and L2P. Overall, our findings suggest that prompt-based algorithms, especially Fed-CPrompt, can effectively mitigate the problem of catastrophic forgetting and improve the training efficiency of lifelong learning systems.

Impact of Asynchronous Continual Learning Tasks. To investigate the impact of client pacing on the training efficiency of our lifelong learning system, we conduct experiments with varying degrees of client pacing. Specifically, we compare the system's performance when all clients move to the next task simultaneously versus when some clients move to the next task while others are still at the current task. Our results show that when some clients move to the next task, the knowledge of the next task can benefit the current task prompt by providing additional context and improving the convergence speed

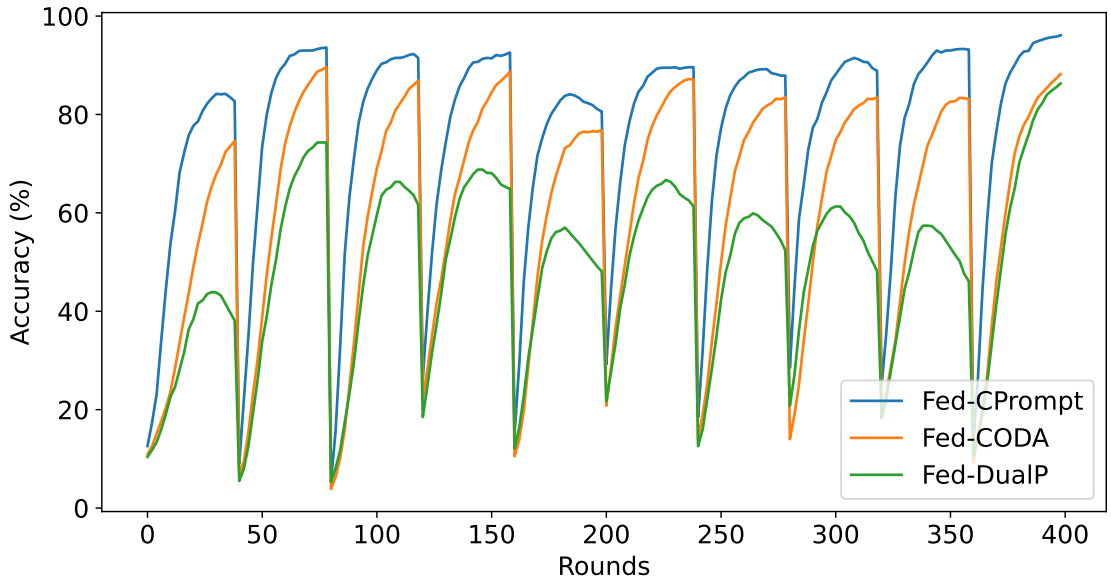


Figure 4. Test accuracy of the prompt-based methods at each round of federated training.

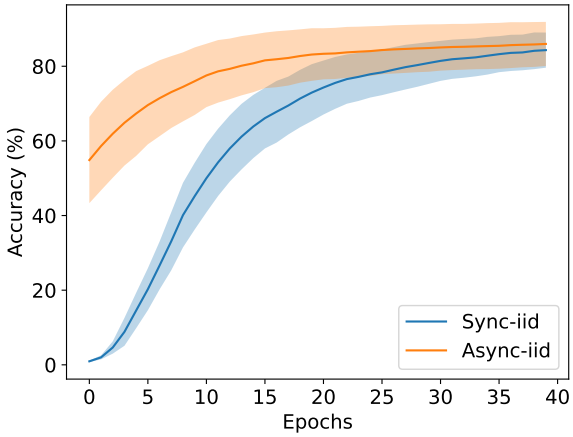


Figure 5. Test accuracy of first 5 tasks in synchronous (sync) and asynchronous (async) iid settings. Under asynchronous settings, prompts in our approach leverage information from other tasks, thus increasing the convergence speed and better initialization.

(Figure 5). The model can leverage the knowledge learned from the next task to understand the current task better, leading to faster convergence and improved accuracy.

Moreover, we also explore the idea of leveraging the prompts from other tasks for example in our design, the clients on task $\mathcal{T}_{m+1}$ leverage prompts from task $\mathcal{T}_m$ to improve the convergence speed of the current task. In addition, clients on task $\mathcal{T}_{\Updownarrow}$ have an increased capacity due to the additional prompt from task $\mathcal{T}_{m+1}$. Our experiments show that incorporating prompts from previous tasks into the current task prompt can significantly improve the convergence speed and reduce the training time, as shown in Figure 5. This is because the model can reuse the knowledge learned from previous tasks and incorporate it into the current task prompt to improve its understanding of the new task.