

# PCL: Prompt-based Continual Learning for User Modeling in Recommender Systems

Mingdai Yang  
myang72@uic.edu  
University of Illinois at Chicago  
Chicago, USA

Shaoyuan Xu  
shaoyux@amazon.com  
Amazon  
Seattle, USA

Simone Shao  
simengsh@amazon.com  
Amazon  
Seattle, USA

Fan Yang  
fnam@amazon.com  
Amazon  
Seattle, USA

Tianchen Zhou  
tiancz@amazon.com  
Amazon  
Seattle, USA

Jia Liu  
liu@ece.osu.edu  
Amazon & The Ohio State University  
Columbus, USA

Yanhui Guo  
yanhuig@amazon.com  
Amazon  
Seattle, USA

Yetian Chen  
yetichen@amazon.com  
Amazon  
Seattle, USA

Yan Gao  
yanngao@amazon.com  
Amazon  
Seattle, USA

## Abstract

User modeling in large e-commerce platforms aims to optimize user experiences by incorporating various customer activities. Traditional models targeting a single task often focus on specific business metrics, neglecting the comprehensive user behavior, and thus limiting their effectiveness. To develop more generalized user representations, some existing work adopts Multi-task Learning (MTL) approaches. But they all face the challenges of optimization imbalance and inefficiency in adapting to new tasks. Continual Learning (CL), which allows models to learn new tasks incrementally and independently, has emerged as a solution to MTL's limitations. However, CL faces the challenge of catastrophic forgetting, where previously learned knowledge is lost when the model is learning the new task. Inspired by the success of prompt tuning in Pre-trained Language Models (PLMs), we propose PCL, a Prompt-based Continual Learning framework for user modeling, which utilizes position-wise prompts as external memory for each task, preserving knowledge and mitigating catastrophic forgetting. Additionally, we design contextual prompts to capture and leverage inter-task relationships during prompt tuning. We conduct extensive experiments on real-world datasets to demonstrate PCL's effectiveness.

## CCS Concepts

• Information systems → Retrieval models and ranking.

## Keywords

Prompt Tuning; Continual Learning; User Modeling

## ACM Reference Format:

Mingdai Yang, Fan Yang, Yanhui Guo, Shaoyuan Xu, Tianchen Zhou, Yetian Chen, Simone Shao, Jia Liu, and Yan Gao. 2025. PCL: Prompt-based Continual Learning for User Modeling in Recommender Systems. In *Proceedings of Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3701716.3715589>

## 1 Introduction

User modeling has been widely applied in large e-commerce platforms to interpret and optimize user experience. Most machine learning models for user modeling optimize for their respective business metrics without jointly considering all activities performed by customers [3, 4], resulting in suboptimal performance. For instance, a sequential recommendation model trained only on users' purchase sequences could significantly benefit from incorporating auxiliary relations and multiple behaviors from users and items (e.g., users' delivery preferences, co-clicks, and return behaviors) to optimize its tasks [7, 16, 20]. Thus, for a seamless and consistent customer experience throughout the shopping journey, it is highly desirable to build a universal representation that incorporates all data modes to easily adapt to a large variety of applications and scale across multiple marketplaces at an affordable cost. To date, most existing works adopt multi-task learning (MTL) to learn a generalized representation by simultaneously training a single model on multiple tasks [8, 9, 15]. However, MTL methods often suffer from optimization imbalance [10, 21]. What's worse is that, when training on a new incoming task, MTL requires full model retraining together with all previous tasks, which is inefficient or even infeasible.

To address the challenges of sequentially arriving new tasks under MTL, Continual Learning (CL) has recently been introduced [5, 18], which aims to learn each new task incrementally without retraining from previous tasks. However, in CL, a major challenge is to mitigate the catastrophic forgetting (CF) problem, i.e., the learned knowledge from previous tasks is lost when the model is learning a new task. CONURE [18] addresses CF by training part of the model

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*WWW Companion '25, Sydney, NSW, Australia*

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1331-6/25/04

<https://doi.org/10.1145/3701716.3715589>

parameters for each task, but this restricts the model’s learning capability due to the reduction of available per-task learnable parameters. TERACON [5] retains learned knowledge by regularizing the model updates between consecutive tasks. However, it requires continually updating the backbone and the adapters of all previous tasks, which is computationally inefficient.

With the success of prompt tuning in Pretrained Language Models (PLMs), prompt tuning has been leveraged in CL for mitigating CF of sequential text-based and image-based classification tasks [2, 12]. Prompt tuning updates a pretrained model with low memory and computational costs, since only a small number of parameters corresponding to the prompt of each task are fine-tuned [14, 17]. However, so far, how to apply prompt tuning for continual user modeling in the recommender systems design remains under-explored. Notably, tasks in user modeling, such as user age prediction and next thumbs-up item prediction, are more heterogeneous than classifying text or images with different groups of labels. Many features in each task are unique and could vary dramatically, such as user behaviors, user profiles, and item categories. In this work, we propose a Prompt-based Continual Learning (PCL) framework for continual user modeling tasks in recommender systems, which leverages prompts to enable task adaptation and knowledge retention among consecutive tasks. We design position-wise prompts in the form of external memory for each individual task. The position-wise prompt for each task inherits and retains the knowledge from previously trained tasks during new task adaptation. To guide prompt tuning with inter-task contextual relations, we propose a new contextual prompt-based on the attention between tasks learned during downstream finetuning. Our main contributions to this work are summarized as follows:

- We propose a new PCL framework that leverages prompts for continual user modeling in recommender systems design. To our knowledge, PCL is the first work to mitigate catastrophic forgetting by prompt tuning in continual user modeling.
- To capture and exploit relationships between tasks during prompt tuning, we design simple and effective contextual prompts based on the attention mechanism.
- We conduct extensive experiments on two real-world datasets to i) validate the efficacy of PCL and ii) demonstrate its robustness to different task orders and cold-start settings.

## 2 Problem Formulation

We use  $\mathcal{T} = \{T_1, T_2, \dots, T_{|\mathcal{T}|}\}$  to denote the set of sequentially arriving tasks. We let  $\mathcal{I}$  and  $\mathcal{U}$  represent the item set and the user set, respectively. In each task, every user  $u \in \mathcal{U}$  is associated with a behavior sequence  $s^u = \{i_1^u, i_2^u, \dots, i_n^u\}$ , where  $i_j^u \in \mathcal{I}$  is the  $j$ -th item interacted by the user  $u$ . Each task  $T_k$  is associated with a set of labels  $Y^k$  with  $|Y^k| = |\mathcal{U}|$ . In other words, each user has a label to be predicted in each task, such as the category of the next clicked item in next-category prediction. Our goal is to train a single model for all tasks in a sequential manner to predict the label of each user in each task. Upon the completion of training on all tasks, this model will be applied in the inference stage without further training. In this work, next item prediction is always used as the first task  $T_1$  due to its self-supervised nature. A recommendation backbone model is pretrained on  $T_1$  to capture and preserve

user-item interaction information before prompt tuning for other downstream tasks.

## 3 The Proposed PCL Framework

### 3.1 Position-wise Prompts

With a frozen backbone model and its item embeddings pretrained in a self-supervised fashion on the next-click prediction task  $T_1$ , we expand the model’s capability by introducing a prompt for each of the downstream tasks. Formally, let  $\mathbf{p}_k \in \mathbb{R}^{n \times d}$  denote the learnable position-wise prompt sequence for task  $T_k$ , where  $n$  is the maximum length of behavior sequences and  $d$  is the dimension of the pretrained item embedding. In user modeling based on behavior sequences, items interacted by a user more recently reflect more recent preferences and features for this user. Thus, for each behavior sequence, we only fuse the last  $t$  item embeddings with their corresponding position-wise prompts. Assuming the backbone model has been pretrained by a behavior sequence  $s^u$  in self-supervised task  $T_1$ , we prompt the behavior embedding sequence as follows:

$$\mathbf{E}_k^{s^u} = \mathbf{E}_1^{s^u} [:-t, :] \| (\mathbf{E}_1^{s^u} [-t :, :] \oplus \mathbf{p}_k [-t :, :]), \quad (1)$$

where  $\mathbf{E}_1^{s^u} [-t :, :]$  represents the embeddings of the last  $t$  items in the behavior sequence and  $\mathbf{E}_1^{s^u} [:-t, :]$  denotes the embeddings of the remaining items in the sequence. We fuse the last  $t$  prompts of the position-wise prompt sequence  $\mathbf{p}_k [-t :, :]$  with the last  $t$  item embeddings by element-wise addition  $\oplus$ , and then concatenate the fused embedding with the frozen first  $n - t$  item embeddings. By doing so, the most relevant prior knowledge for the task  $k$  is extracted by prompt tuning.

The total memory cost of prompts for all tasks is  $|\mathcal{T}| \times n \times d$ , which is negligible in comparison to the number of backbone model parameters and the item embedding table size  $|\mathcal{I}| \times d$ .

### 3.2 Contextual Prompts

In addition to the position-wise prompts that allow a frozen model for task adaptation by expanding learning capacity, we further introduce contextual prompts to guide the prompt tuning by both task information and inter-task relations. A pretrained sentence transformer [13] is deployed to encode task descriptions as text embeddings  $\mathbf{E}_{\mathcal{T}} \in \mathbb{R}^{|\mathcal{T}| \times d}$ , which are frozen in all downstream tasks. Each task description consists of the task name, the input data format, the output data format and the evaluation metric. A multi-head attention layer [11] is used to adaptively learn the relations among all the tasks:

$$\mathbf{P}_{\mathcal{T}} = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O, \quad (2)$$

$$\text{head}_m = \text{Attention}(\mathbf{E}_{\mathcal{T}} W_m^Q, \mathbf{E}_{\mathcal{T}} W_m^K, \mathbf{E}_{\mathcal{T}} W_m^V),$$

where  $W_m^Q, W_m^K, W_m^V \in \mathbb{R}^{d \times d_h}$  and  $W^O \in \mathbb{R}^{d \times d}$  are projection matrices. We obtain the contextual prompts  $\mathbf{P}_{\mathcal{T}} = [\mathbf{p}'_1, \dots, \mathbf{p}'_{|\mathcal{T}|}]$  for all tasks in  $\mathcal{T}$ . Similar to the fusion between pretrained item embeddings and position-wise prompts, we add the contextual prompts to each of the last  $t$  embeddings in the behavior sequence:

$$\mathbf{E}'_k^{s^u} = \mathbf{E}_k^{s^u} [:-t, :] \| (\mathbf{E}_k^{s^u} [-t :, :] + \lambda \mathbf{p}'_k), \quad (3)$$

where  $\lambda$  denotes the prompting intensity parameter. The effectiveness of the proposed contextual prompts is verified in Section 4.3.

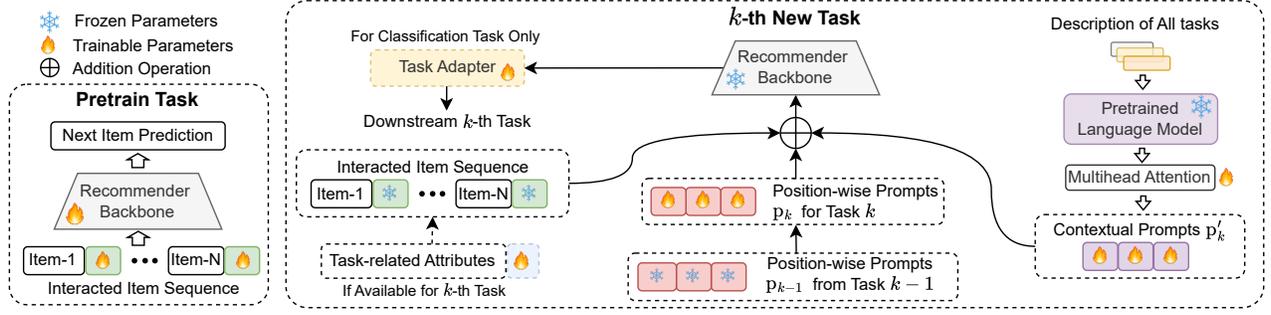


Figure 1: The overall framework of PCL. The backbone model and item embeddings are frozen in the downstream tasks.

### 3.3 Pretraining and Prompt Tuning

Our PCL framework is illustrated in Figure 1. To capture temporal dynamics in user behavior, we first pretrain a sequential recommendation backbone model  $\mathcal{M}$  with all item embeddings  $\mathbf{N} \in \mathbb{R}^{|\mathcal{I}| \times d}$  on the self-supervised next-item prediction task  $T_1$ . For link prediction tasks, we adopt the binary cross entropy loss for sequence recommendation [4] as the objective function. For attribute classification tasks, the cross entropy loss  $\mathcal{L}_{CE}(f_k(\mathcal{M}(\mathbf{E}'_k^{s_u})), \mathbf{y}_u^k)$  is adopted as the objective function, where  $\mathbf{y}_u^k$  is the ground-truth label,  $f_k(\cdot)$  represents the task adapter, which can be a linear layer or an MLP model to map the dimension of latent vectors to the number of classes for classification.

During the prompt tuning stage, the position-wise prompts are randomly initialized in  $T_2$ . From  $T_3$  to  $T_{|\mathcal{T}|}$ , the prompts  $\mathbf{p}_k$  are initialized as  $\mathbf{p}_{k-1}$ , which is the optimized prompt for the previous task. In addition, our framework is designed to leverage task-related attributes when they are available in downstream tasks. This capability allows the model to incorporate additional contextual information, enhancing its adaptability to specific task requirements. For example, in the classification of the next item’s category in a behavior sequence, a feature vector sequence containing each item’s category embedding is added to and finetuned with the frozen item sequence. In this work, we use SASRec [4] as the backbone model and the Adam [6] method as the optimizer.

## 4 Experiment

### 4.1 Experiment Setup

**1) Dataset:** We conduct experiments on two real-world datasets: Tenrec [19] and MovieLens<sup>1</sup>. We sample the datasets with fewer user-item interactions to provide a more challenging evaluation environment. Tenrec involves two link prediction tasks and two attribute classification tasks on 15,046 users and 16,475 items from the Kandian platform. For Tenrec,  $T_1$  contains the users’ recent 50 news & video-watching interactions,  $T_2$  aims to predict next thumb-up items for users,  $T_3$  aims to classify categories of next-click items, and  $T_4$  is to predict users’ ages. MovieLens contains three link prediction tasks and one classification task on 20,741 users and 10,033 items from the MovieLens website. Specifically,  $T_1$  contains users’ recent 50 clicking interactions,  $T_2$  and  $T_3$  aim to

predict next four-starred and five-starred movies rated by users, and  $T_4$  is trained to classify genres of the next clicked movies. We select Hit Ratio (HR@5 and HR@10) and accuracy (Acc) as the metrics for link prediction and attribute classification, respectively. We use the batch-ranking [4] and all-ranking protocols in MovieLens and Tenrec, respectively, for a comprehensive HR evaluation.

**2) Baselines:** To show the effectiveness of PCL, we compare it with four single-objective baselines. *a) SASRec* trains a single model for each task from scratch without knowledge transfer between tasks. *b) SinMo* trains a single model for all tasks without saving parameters from previous tasks. *c) FineAll* finetunes all the model for each task after training on  $T_1$  for transfer learning. *d) Teracon* [5] is a CL-based method leveraging soft masks and function regularization for knowledge transfer. We also compare PCL with four MTL baselines that optimize all tasks simultaneously. *MTL* [1] is a standard multi-task optimization method via parameter sharing. *MMOE* [8] is an MTL method based on mixture-of-experts. *MTL+* and *MMOE+* apply a multi-gradient descent algorithm [10] to dynamically determine the weight of each objective to address gradient conflicts. In addition to overall performance comparison, we design experiments to answer the following three research questions:

- **RQ1:** Does the proposed contextual prompts benefit PCL?
- **RQ2:** Is the performance of PCL robust to the order of tasks?
- **RQ3:** Does PCL improve downstream finetuning with cold-start items?
- **RQ4:** Can PCL generate universal user representations for unseen models and tasks?

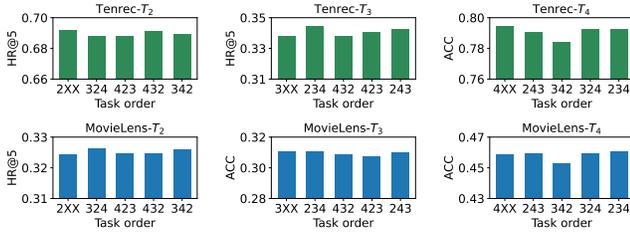
### 4.2 Overall Performance

We show the results in Table 1. The best and second-best single-objective methods are in boldface and underlined. The best MTL method better/worse than PCL is marked with  $\Delta/\nabla$ . We have the following observations: 1) With only one copy of backbone parameters preserved, PCL achieves the best in 6 out of 8 tasks compared with other single-objective baselines. This justifies the advantages of prompt tuning for continual user modeling; and 2) Even if the supervised signals from other tasks are not available when training on the target task, PCL is generally better than all MTL baselines that leverage labels in all tasks for optimization. This verifies the efficacy of prompts as external memory for task adaptation.

<sup>1</sup><https://grouplens.org/datasets/movielens/25m/>

**Table 1: Overall performance comparison. #B is the number of backbone models to be preserved for inference on all tasks. #F is the number of objective tasks to be optimized simultaneously. Multi-task learning methods simultaneously supervised by all tasks with #F= 4 are compared as the upper bounds of CL baselines.**

Method	Tenrec						MovieLens						#B	#F	
	$T_1$		$T_2$		$T_3$	$T_4$	$T_1$		$T_2$		$T_3$	$T_4$			
	HR@5	HR@10	HR@5	HR@10	ACC	ACC	HR@5	HR@10	HR@5	HR@10	HR@5	HR@10			ACC
SASRec	0.6582	0.8112	0.6613	0.8107	0.3329	0.7722	0.3018	0.4450	0.3007	0.4627	0.2859	0.4369	0.4783	4	1
SimMo	0.1028	0.1777	0.1083	0.1874	0.0581	0.7553	0.1157	0.1876	0.0937	0.1651	0.0890	0.1558	0.4696	1	1
FineAll	-	-	0.6832	0.8292	0.3409	0.7837	-	-	0.3291	0.4785	0.3015	0.4547	<b>0.4812</b>	4	1
Teracon	0.5703	0.7388	0.6390	0.7804	0.3071	0.7725	0.1177	0.1704	0.2878	0.3986	<b>0.4297</b>	<b>0.5594</b>	0.3540	1	1
PCL	<b>0.6582</b>	<b>0.8112</b>	<b>0.6936</b>	<b>0.8372</b>	<b>0.3466</b>	<b>0.7974</b>	<b>0.3018</b>	<b>0.4450</b>	<b>0.3315</b>	<b>0.4845</b>	0.3155	0.4615	0.4634	1	1
MTL	0.6361	0.7965	0.6968	0.8355	0.3285	0.7447 <sup>∇</sup>	0.2816 <sup>∇</sup>	0.4364	0.3036	0.4603	0.3002	0.4414	0.4744	1	4
MMOE	0.6565 <sup>∇</sup>	0.8085 <sup>∇</sup>	0.6884	0.8351	0.3227	0.7385	0.2517	0.3631	0.2521	0.3734	0.2460	0.3682	0.4609	1	4
MTL+	0.6489	0.8045	0.7043 <sup>Δ</sup>	0.8434 <sup>Δ</sup>	0.3528	0.7434	0.2811	0.4503 <sup>Δ</sup>	0.3106 <sup>∇</sup>	0.4707 <sup>∇</sup>	0.3020 <sup>∇</sup>	0.4522 <sup>∇</sup>	0.4759 <sup>Δ</sup>	1	4
MMOE+	0.6157	0.7899	0.6823	0.8331	0.3551 <sup>Δ</sup>	0.6166	0.2536	0.3732	0.2659	0.3952	0.2481	0.3725	0.4716	1	4



**Figure 2: Performance of each task w.r.t the task order.**

### 4.3 RQ1: Impact of Contextual Prompts

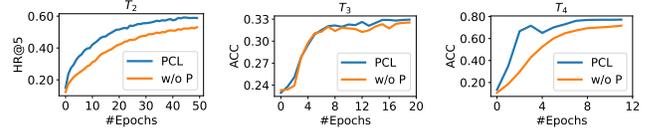
**Table 2: Ablation study of contextual prompts.**

Method	Tenrec			MovieLens		
	$T_2$	$T_3$	$T_4$	$T_2$	$T_3$	$T_4$
	HR@5	ACC	ACC	HR@5	HR@5	ACC
w/o $P_{\mathcal{T}}$	0.6927	0.3404	0.7908	0.3259	0.3083	0.4431
w/o PLM	0.6871	0.3347	0.7886	0.3269	0.3055	0.4566
PCL	<b>0.6936</b>	<b>0.3449</b>	<b>0.7974</b>	<b>0.3315</b>	<b>0.3155</b>	<b>0.4634</b>

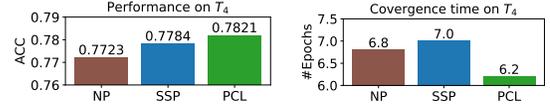
To quantify the contribution of contextual prompts in our framework, we compare the performance of PCL to its variants (i) without contextual prompts  $P_{\mathcal{T}}$  and (ii) without PLM to initialize contextual prompts from task descriptions. For the variant of PCL without PLM, we randomly initialize the contextual prompts. The results are shown in Table 2. We find that employing PLM to encode task descriptions leads to stable enhancement in both datasets. This implies contextual prompts in PCL effectively capture and leverage task relations during sequential task learning.

### 4.4 RQ2: Robustness against Task Orders

Considering that the sequence of tasks cannot be arbitrarily determined in real-world scenarios, we verify the robustness of PCL against task orders by conducting experiments on all permutations of downstream tasks. The results are shown in Figure 2, where each coordinate on the x-axis represents a task order. For example, 4XX means the performance of  $T_4$  when  $T_4$  is the first downstream task after pretraining on the self-supervised  $T_1$ . We observe that the



**Figure 3: Performance w.r.t. the number of training epochs with 50% cold-start items in Tenrec dataset when  $lr = 0.01$ .**



**Figure 4: Performance and convergence time of an MLP model on  $T_4$  in Tenrec with user features obtained from different pretraining methods.**

average performance degradation from the best to the worst task order on all tasks is only 1.20% in both datasets. Since each downstream task is trained with independent prompts, the performance of each task is robust to the training order.

### 4.5 RQ3: Finetuning with Cold-start Items

We investigate the efficacy of PCL in finetuning with cold-start items by masking 50% items in  $T_1$  as cold-start items. The performance in the following tasks is shown as curves in Figure 3, compared with only finetuning cold-start item embeddings and task adapters without prompts. Benefiting from the adaptability of prompt tuning, PCL converges to better performance without extensive retraining. The necessity of prompt tuning stands out more in  $T_2$  without available task adapters.

### 4.6 RQ4: Universal User Representation

We further conduct a case study to verify that PCL can generate a universal user representation through prompt tuning. Besides the downstream tasks where the prompts are finetuned, this representation can be leveraged as an informative user feature in new tasks with a model other than the backbone. To be concrete, we finetune

the prompts on  $T_2$  and  $T_3$  in Tenrec, and the behavior latent vector output from the model is used as the universal representation of each user. We concatenate these universal representations as fixed user features to one-hot user vectors and feed them into a simple MLP with one hidden layer for  $T_4$ . For comparison, we also evaluate the performance of (i) *NP*: randomly initialized vectors as user features, and (ii) *SSP*: behavior latent vectors pretrained on self-supervised  $T_1$  as user features. The results in Figure 4 indicates that the universal user representations pretrained by PCL achieves better accuracy with faster convergence speed.

## 5 Conclusion

In this paper, we proposed a framework called PCL, which leverages prompts for task adaptation and knowledge retention in continual user modeling in recommender systems design. We proposed position-wise prompts to inherit knowledge from observed tasks and preserve the knowledge learned from each downstream task. We also introduced contextual prompts to guide prompt tuning with relations between tasks. Our experiments demonstrated the benefits of prompt tuning in continual user modeling and the effectiveness of PCL in various scenarios.

## References

- [1] Rich Caruana. 1998. Multitask Learning. In *Learning to Learn*, Sebastian Thrun and Lorien Y. Pratt (Eds.). Springer, 95–133.
- [2] Yanhui Guo, Shaoyuan Xu, Jimmiao Fu, Jia Liu, Chaosheng Dong, and Bryan Wang. 2024. Q-Tuning: Queue-based Prompt Tuning for Lifelong Few-shot Language Learning. In *Findings of the Association for Computational Linguistics: NAACL 2024*. ACL, Mexico City, Mexico, 2595–2622.
- [3] Yang Jiao, Fan Yang, Yetian Chen, Yan Gao, Jia Liu, and Yi Sun. 2024. Rethinking Sequential Relationships: Improving Sequential Recommenders with Inter-Sequence Data Augmentation. In *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*. 641–645.
- [4] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*. IEEE Computer Society, 197–206.
- [5] Sein Kim, Namkyeong Lee, Donghyun Kim, Min-Chul Yang, and Chanyoung Park. 2023. Task Relation-aware Continual User Representation Learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*. ACM, 1107–1119.
- [6] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- [7] Gang Liu, Fan Yang, Yang Jiao, Ali Bagheri Garakani, Tian Tong, Yan Gao, and Meng Jiang. 2025. Learning attribute as explicit relation for sequential recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
- [8] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. 2018. Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. ACM, 1930–1939.
- [9] Yabo Ni, Dan Ou, Shichen Liu, Xiang Li, Wenwu Ou, Anxiang Zeng, and Luo Si. 2018. Perceive Your Users in Depth: Learning Universal User Representations from Multiple E-commerce Tasks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. ACM, 596–605.
- [10] Ozan Sener and Vladlen Koltun. 2018. Multi-Task Learning as Multi-Objective Optimization. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*. 525–536.
- [11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 5998–6008.
- [12] Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. 2023. Hierarchical Decomposition of Prompt-Based Continual Learning: Rethinking Obscured Sub-optimality. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- [13] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MINLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (Vancouver, BC, Canada) (NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 485, 13 pages.
- [14] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S. Yu. 2024. Instruction-based Hypergraph Pretraining. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*. ACM, 501–511.
- [15] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S. Yu. 2024. Unified Pretraining for Recommendation via Task Hypergraphs. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. ACM, 891–900.
- [16] Yuhao Yang, Chao Huang, Lianghao Xia, Yuxuan Liang, Yanwei Yu, and Chenliang Li. 2022. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 2263–2274.
- [17] Kingtong Yu, Chang Zhou, Yuan Fang, and Xinming Zhang. 2024. MultiGPrompt for Multi-Task Pre-Training and Prompting on Graphs. In *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee (Eds.). ACM, 515–526. <https://doi.org/10.1145/3589334.3645423>
- [18] Fajie Yuan, Guoxiao Zhang, Alexandros Karatzoglou, Joemon M. Jose, Beibei Kong, and Yudong Li. 2021. One Person, One Model, One World: Learning Continual User Representation without Forgetting. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 696–705.
- [19] Guanghu Yuan, Fajie Yuan, Yudong Li, Beibei Kong, Shujie Li, Lei Chen, Min Yang, Chenyun Yu, Bo Hu, Zang Li, Yu Xu, and Xiaohu Qie. 2022. Tenrec: A Large-scale Multipurpose Benchmark Dataset for Recommender Systems. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- [20] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, et al. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *IJCAI*. 4320–4326.
- [21] Tianchen Zhou, FNU Hairi, Haibo Yang, Jia Liu, Tian Tong, Fan Yang, Michinari Momma, and Yan Gao. 2024. Finite-time convergence and sample complexity of actor-critic multi-objective reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning (Vienna, Austria) (ICML'24)*. JMLR.org, Article 2562, 21 pages.