Embracing Plasticity: Balancing Stability and Plasticity in Continual Recommender Systems

Hyunsik Yoo
hy40@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, IL, USA

SeongKu Kang
seongkukang@korea.ac.kr
Korea University
Seoul, Republic of Korea

Ruizhong Qiu
rq5@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, IL, USA

Charlie Xu
caizhx@amazon.com
Amazon.com, Inc.
Seattle, WA, USA

Fei Wang
feiww@amazon.com
Amazon.com, Inc.
Sunnyvale, CA, USA

Hanghang Tong
htong@illinois.edu
University of Illinois
Urbana-Champaign
Urbana, IL, USA

Abstract

In the era of big data and AI, recommender systems must adapt to evolving user preferences and new users/items to maintain high-quality recommendations. Fine-tuning, which updates model parameters using only new data, offers an efficient alternative to full retraining but struggles to balance stability (retaining past knowledge) and plasticity (adapting to new knowledge). While existing methods prioritize stability to address catastrophic forgetting, we argue that plasticity must also be explicitly strengthened, especially for users with rapidly changing preferences. In this work, we propose PlastIcity and StAbility balancing continual recommender systems (PISA), a novel framework that adaptively balances stability and plasticity based on user preference shifts. PISA quantifies preference shifts as changes in user distances to item clusters, and then guides user embeddings by prioritizing stability for stable users and plasticity for dynamic users. To achieve this, PISA leverages backward knowledge from the previous model and forward knowledge from fine-tuning on current data. During training, PISA maximizes mutual information between user-specific parameters and the relevant reference knowledge. Theoretically, we show that enhancing plasticity mitigates distribution shifts more effectively than fine-tuning alone. Empirically, extensive experiments on three real-world datasets validate PISA's superiority over existing methods and highlight the contributions of its components.[1]

CCS Concepts

• **Information systems → Data mining**; • **Computing methodologies → Machine learning**.

Keywords

recommender systems; continual learning; balance between stability and plasticity

[1]The GitHub repository is available at https://github.com/hsyoo32/pisa.

SIGIR '25, Padua, Italy

ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3729964

ACM Reference Format:
Hyunsik Yoo, SeongKu Kang, Ruizhong Qiu, Charlie Xu, Fei Wang, and Hanghang Tong. 2025. Embracing Plasticity: Balancing Stability and Plasticity in Continual Recommender Systems. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '25), July 13–18, 2025, Padua, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3726302.3729964

1 Introduction

In the era of big data and AI, recommender systems continuously gather vast amounts of user feedback, such as product purchases on e-commerce platforms or movie ratings on streaming sites. As new data continually flows in, models can quickly become stale and outdated. To maintain high recommendation quality, models must adapt to capture evolving user preferences and accommodate new users and items. While periodic retraining from scratch using the entire dataset (both historical data and new data) is an option, this approach is highly time-consuming given the likely large volume of historical data. A more efficient alternative is *fine-tuning*, which updates existing model parameters *using only newly collected data* [3, 5, 7, 11, 15, 16, 29, 32, 33]. For example, a model pre-trained on one year's data can be fine-tuned weekly with the latest data.

Continual learning in recommender systems builds on fine-tuning as a foundational strategy to effectively process new data. A core property of continual learning is the ability to balance *stability* (retaining past knowledge) and *plasticity* (adapting to new knowledge) [2, 19, 24, 28, 37]. Although fine-tuning inherits past model parameters, implicitly preserving stability, it may still sacrifice stability for plasticity by focusing solely on new data. This trade-off can result in the loss of past knowledge, a phenomenon known as catastrophic forgetting [31]. Most existing methods [1, 21, 25–27, 34] are designed to address this issue and can be broadly categorized into two strategies: (1) experience replay and (2) knowledge distillation. Experience replay methods selectively reuse past *data* as additional input for current training. For example, [21] uses uniform sampling of past data, while [1] samples past data inversely proportional to its historical popularity. In contrast, knowledge distillation methods integrate past *model parameters* into the current model during training by imposing constraints between past and current parameters. For example, [25–27] minimize the discrepancy between past and current embeddings of users/items, as well as their neighbors in interaction networks.

In this work, while the focus of existing methods on explicitly enhancing fine-tuning's stability remains important, we argue that its plasticity can and should also be explicitly strengthened when needed. Specifically, unlike stable users with small preference shifts, dynamic users with significant preference shifts may benefit more from focusing on new data, implicitly forgetting past knowledge that may no longer be relevant to current preferences. For example, some users may rapidly change interests or show high sensitivity to trends and new items. Such dynamic users can appear on any platforms but are especially common on platforms like short-form video services or seasonal e-commerce (e.g., during Christmas). For these users, past consumption patterns may have little relevance—or even contradict—current preferences. At the very least, current consumption patterns would likely be more important than past ones, suggesting the potential need to prioritize current data.

Our empirical observations further support the need for explicitly enhancing fine-tuning's plasticity. Specifically, we find that, unlike shuffled data, fine-tuning outperforms retraining in real-world temporal settings, highlighting the benefit of prioritizing current data. Additionally, dynamic users benefit more from fine-tuning than stable users, emphasizing the importance of ensuring plasticity for users with high preference shifts. These findings may warrant an even greater emphasis on focusing exclusively on current data. Detailed analyses are in Figure 2/Table 2 in Section 2.

Motivated by these insights, our goal is to enhance *both aspects* of fine-tuning by adaptively balancing stability and plasticity based on user preference shifts. The bottom part of Figure 1 illustrates this philosophy: for a user, the *"center part"* represents the parameters resulting from pure fine-tuning, the *"left part"* reflects the enhancement of stability, and the *"right part"* represents the enhancement of plasticity. Here, the plasticity axis is a novel dimension that is absent in prior methods. While some existing methods may adaptively impose stability based on user preference shifts–thereby implicitly achieving relative plasticity for dynamic users compared to stable ones–they are limited to adjustments along the stability axis, determining only the extent of past knowledge retention [26]. In contrast, our approach aims to go beyond fine-tuning's inherent plasticity, explicitly enhancing it for users who need it.

To achieve this goal, we propose PlastIcity and StAbility balancing continual recommender systems, named PISA, a novel continual learning framework that involves two general steps: **(1) quantifying user preference shifts:** preference shifts are measured as changes in user distances to item clusters across successive time stages, and **(2) adjusting user embeddings:** based on the degree of preference shifts, user embeddings are adjusted to prioritize stability or plasticity.

A core aspect of plasticity enhancement is improving adaptability to rapid shifts in user preferences or the emergence of new entities. To achieve this, we introduce reference parameters for plasticity, termed *"forward knowledge,"* and adjust user embeddings toward this knowledge. Forward knowledge is obtained through pure fine-tuning on the current data and offers several advantages: First, it captures latent patterns in the data, providing richer supervision for enhancing plasticity (e.g., similarities between entities in the embedding space or potential interactions) compared to relying

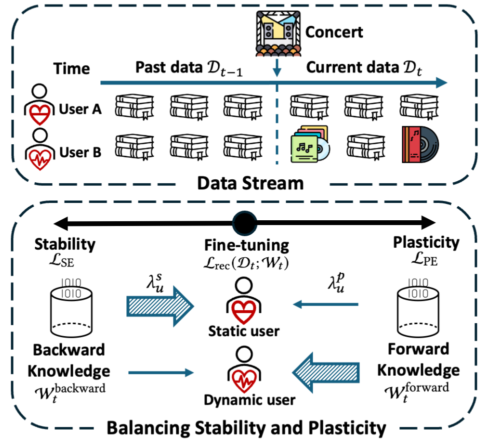

Figure 1: Concepts and ideas of our PISA framework.

solely on input data-level signals. This is particularly valuable given the high sparsity of user interaction data, especially in incremental data blocks used for fine-tuning. Second, we theoretically demonstrate that using forward knowledge for plasticity guidance more effectively mitigates the impact of distribution shifts compared to fine-tuning alone. Symmetrically, for stability enhancement, PISA maintains *"backward knowledge,"* which corresponds to parameters derived from the model at the previous time stage.

During training, PISA dynamically balances stability and plasticity by maximizing mutual information [6, 35] (e.g., InfoNCE [9, 20]) between the current user embeddings—optimized with the recommendation loss—and the corresponding backward or forward knowledge, based on the user's degree of preference shifts. Also, PISA selectively applies such stability/plasticity enhancement to stable and dynamic users with the lowest and highest degrees of preference shifts, respectively, reducing uncertainty for intermediate users. Furthermore, PISA incorporates the backward/forward knowledge of a user's neighboring items, providing richer and more comprehensive guidance. Extensive experiments on three real-world recommendation datasets empirically validate the effectiveness of PISA and highlight the contributions of its individual components.

In summary, the paper makes the following key contributions:

- **Perspective.** We introduce a new perspective that explicitly enhancing the plasticity of fine-tuning is as critical as prior efforts to enhance stability.
- **Theory.** We provide a theoretical analysis demonstrating that fine-tuning with plasticity enhancement mitigates the impact of distribution shifts more effectively than fine-tuning alone, highlighting the need to explicitly emphasize plasticity.
- **Algorithm.** We propose PISA, a novel continual learning framework that adaptively balances stability and plasticity based on user preference shifts.
- **Experiments.** Extensive experiments on real-world continual recommendation datasets validate the effectiveness of PISA, showing an average increase of 7.25% compared to the best-performing recent competitors.

2 Preliminaries and Related Works

In this section, we introduce key notations, review continual recommendation settings and related works, present motivational

Table 1: Main symbols used in this paper.

Symbol	Description
$\mathcal{D}_t$	Dataset collected at time stage t
$\mathcal{U}_t, \mathcal{I}_t, \mathcal{E}_t$	Sets of users, items, and their interactions at time stage t
Y_t	User-item interaction matrix at time stage t
$\mathcal{W}_t$	Set of model parameters at time stage t
$\mathbf{e}_u^t, \mathbf{e}_i^t$	User and item embeddings of u and i at t, respectively
$\mathcal{L}_{\text{rec}}$	Recommendation loss
$\mathcal{M}_{ui}$	Set of negative items for (u, i) for $\mathcal{L}_{\text{rec}}$
$\mathcal{L}_{\text{SE}}, \mathcal{L}_{\text{PE}}$	Stability and plasticity enhancement losses, respectively
$\mathcal{W}_t^{\text{backward}}$	Backward knowledge at time stage t
$\mathcal{W}_t^{\text{forward}}$	Forward knowledge at time stage t
$\mathcal{N}(u)$	Set of neighboring items of u up to t for $\mathcal{L}_{\text{SE}}$ and $\mathcal{L}_{\text{PE}}$
λ_u^s, λ_u^p	User-specific stability and plasticity weights, respectively
$\mathbf{p}_u^t$	Preference vector of user u at time t
L	Top-L selection percentage for stability/plasticity weights

analyses, and formally define the problem of balancing stability and plasticity for each user in continual recommendations.

Notations. Table 1 provides a list of symbols used in this paper. The dataset collected at time stage t is denoted as $\mathcal{D}_t = \{\mathcal{U}_t, \mathcal{I}_t, \mathcal{E}_t, \mathrm{Y}_t\}, \forall t \in \{1, \ldots, T\}$, where $\mathcal{U}_t$ is the user set, $\mathcal{I}_t$ is the item set, $\mathcal{E}_t$ is the user-item interaction set, and Y_t is the user-item interaction matrix. We consider binary user-item interactions in this work, where $\mathrm{Y}_t[u, i] = 1$ if user u has interacted with item i within the t-th time stage, and $\mathrm{Y}_t[u, i] = 0$ otherwise. The initial user set, item set, user-item interaction set, and user-item interaction matrix prior to the first time stage (i.e., pretrain data) are denoted as $\mathcal{U}_0$, $\mathcal{I}_0$, $\mathcal{E}_0$, and Y_0, respectively. Lastly, the subscript $_{:t}$ represents data from the beginning up to time period t.

Continual recommendations. We assume that an initial recommendation model has been pre-trained on the dataset $\mathcal{D}_0 = \{\mathcal{U}_0, \mathcal{I}_0, \mathcal{E}_0, \mathrm{Y}_0\}$ in an offline manner. Subsequently, a continual model is updated in an online manner using the *fine-tuning* training strategy, relying solely on newly collected data $\mathcal{D}_t$ at each time stage t, $\forall t \in \{1, \ldots, T\}$. Formally, the loss of fine-tuning at each t can be represented as follows: $\mathcal{L} := \mathcal{L}_{\text{rec}}(\mathcal{D}_t; \mathcal{W}_t)$, where $\mathcal{L}_{\text{rec}}$ represents the recommendation loss (e.g., BPR loss [23]), and $\mathcal{W}_t$ denotes the current model parameters (e.g., user/item embeddings) being optimized. At the beginning of training at each t, $\mathcal{W}_t$ is initialized to $\mathcal{W}_{t-1}$ (the parameters from the previous stage), implicitly preserving stability. However, despite this implicit stability, the model inevitably forgets some past knowledge as it learns solely from the current data. To address this issue, existing methods explicitly enhance stability using two main techniques: experience replay [1, 21, 36] and knowledge distillation [25–27].

<u>**Experience replay**</u>-based methods use a subset of past data and can be formally represented as follows:

$$\mathcal{L} := \mathcal{L}_{\text{rec}}(\mathcal{D}_t \cup \mathcal{D}^*_{:t-1}; \mathcal{W}_t), |\mathcal{D}^*_{:t-1}| = \gamma |D_t|, \quad (1)$$

where $\mathcal{D}^*_{:t-1}$ is a selection of representative historical data (i.e., $\mathcal{D}^*_{:t-1} \subset \mathcal{D}_{:t-1}$), typically much smaller than the historical data (e.g., $\gamma = 1$, making the selected data the same size as the incremental data). Different methods vary in how they construct $\mathcal{D}^*_{:t-1}$.

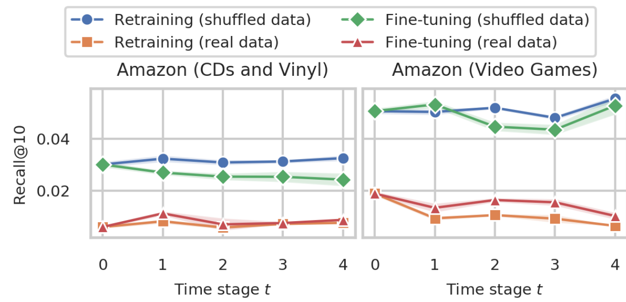

Figure 2: Performance comparison of Retraining and Fine-tuning on shuffled and real temporal data. Fine-tuning outperforms Retraining in the temporal setting, showing the benefit of prioritizing current data in real-world scenarios.

Table 2: Recall@10(in %) of Fine-tuning and Retraining, along with their difference, for stable and dynamic users. Dynamic users gain more from Fine-tuning, highlighting the need for plasticity to adapt to rapidly evolving user preferences.

Datasets	User group	Retraining	Fine-tuning	Diff. (C2-C1)
Amazon (CDs and Vinyls)	Stable	$0.97_{\pm 0.09}$	$1.07_{\pm 0.18}$	$0.10_{\pm 0.23}$
	Dynamic	$0.81_{\pm 0.02}$	$1.17_{\pm 0.18}$	$\mathbf{0.36_{\pm 0.18}}$
Amazon (Video Games)	Stable	$1.07_{\pm 0.19}$	$1.33_{\pm 0.19}$	$0.26_{\pm 0.19}$
	Dynamic	$0.77_{\pm 0.10}$	$1.21_{\pm 0.17}$	$\mathbf{0.44_{\pm 0.16}}$

<u>**Knowledge distillation**</u>-based methods use past model parameters and are represented as follows:

$$\mathcal{L} := \mathcal{L}_{\text{rec}}(\mathcal{D}_t; \mathcal{W}_t) + \lambda \cdot \sum_{u \in \mathcal{U}_{:t}} \mathcal{L}_{KD}(\mathcal{W}_t(u), \mathcal{W}_{t-1}(u)), \quad (2)$$

where $\mathcal{L}_{KD}$ is the knowledge distillation loss, which minimizes discrepancies (e.g., differences or negative mutual information) between past and current parameters, $\mathcal{W}(u)$ is user-specific parameters, and λ is a scaling factor for the distillation loss.[2] Different methods vary in how they design the distillation loss. For instance, [27] incorporates local and global structure, and self-information from interaction networks, while [25] updates layer-wise parameters of graph-based models for users and their neighbors via contrastive distillation. Recent work [26] proposes a personalized approach, assigning distinct stability weights λ_u to each user, i.e., $\lambda_u \cdot \sum_{u \in \mathcal{U}_{:t}} \mathcal{L}_{KD}$, where λ_u is expected to learned as inversely proportional to each user's degree of preference shifts. The intuition is that stable users, whose preferences change minimally, benefit more from retaining past knowledge (i.e., stability). While this method personalizes stability enhancement, it lacks an explicit mechanism to enhance plasticity for users who need it. Our work introduces a new dimension of plasticity enhancement, complementing stability, to better personalize the balance between the two.

After training at each t using the corresponding loss, the model generates a top-N recommendation list $[i_1, \ldots, i_N]$ for each user u, ranked by predicted scores $\widehat{\mathrm{Y}}_t[u, i], \forall i$.

Motivational analyses. As mentioned in Section 1, we provide motivational analyses highlighting the need for explicitly enhancing fine-tuning's plasticity. First, Figure 2 shows that with shuffled interaction data (i.e., no temporal sequence), retraining outperforms fine-tuning at each time stage on datasets like Amazon-CDs and

[2]We omit the item-side KD loss for brevity, but it can be implemented analogously.

Video Games. However, this trend reverses with chronologically ordered data, indicating that in real-world temporal scenarios, fine-tuning benefits from prioritizing current data rather than treating all data equally—a limitation when temporal information is ignored.

Second, we analyze stable and dynamic user groups (see Section 4.3 for classification details). Table 2 shows the average performance over time for those groups, along with their difference (Fine-tuning - Retraining) in the last column. The results reveal that both groups benefit from fine-tuning, with dynamic users gaining more than stable users, underscoring the importance of ensuring plasticity, particularly for users with rapidly evolving preferences.

Problem definition. Our problem is formally defined as follows:

PROBLEM 1 (Personalized Balance between Stability and Plasticity). ***Input:*** *(1) a pre-trained recommendation model with parameters $\mathcal{W}_0$; (2) a continually collected dataset $\mathcal{D}_t = \{\mathcal{U}_t, \mathcal{I}_t, \mathcal{E}_t, \mathrm{Y}_t\}, \forall t \in \{1, \ldots, T\}$;* ***Output:*** *a continual model that achieves high-quality recommendation at each time stage t by adaptively balancing stability and plasticity for each user during fine-tuning.*

3 Proposed Framework

In this section, we propose PlastIcity and StAbility balancing continual recommender systems, named PISA, a novel continual learning framework designed to explicitly enhance both stability and plasticity for each user. An illustration of PISA is shown in Figure 1.

3.1 Overview and Outline

We begin by outlining the high-level objective of PISA and provide an overview of its key components. The overall loss is written as:

$$\mathcal{L} := \underbrace{\mathcal{L}_{\text{rec}}(\mathcal{D}_t; \mathcal{W}_t)}_{\text{fine-tuning}} + \alpha \left[\sum_{u \in \mathcal{U}_{:t}} \underbrace{\lambda_u^s \mathcal{L}_{\text{SE}}(\mathcal{W}_t, \mathcal{W}_t^{\text{backward}}; u)}_{\text{stability adjustment}} + \underbrace{\lambda_u^p \mathcal{L}_{\text{PE}}(\mathcal{W}_t, \mathcal{W}_t^{\text{forward}}; u)}_{\text{plasticity adjustment}} \right], \tag{3}$$

where α is global scaling parameter and $\mathcal{L}_{\text{rec}}(\mathcal{D}_t; \mathcal{W}_t)$ corresponds to fine-tuning with the current data. In this work, we use Bayesian personalized ranking (BPR) loss [23], defined as:

$$\mathcal{L}_{\text{rec}} := -\frac{1}{|\mathcal{E}_t|} \sum_{(u,i,t) \in \mathcal{E}_t} \frac{1}{|\mathcal{M}_{ui}|} \sum_{i' \in \mathcal{M}_{ui}} \log(\sigma(s_{ui} - s_{ui'})), \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function, and $\mathcal{M}_{ui}$ is a set of sampled negative items for (u, i). The recommendation scores s_{ui} between a user u and an item i is modeled as the dot product of their embeddings. The remaining terms in Eq. (3) represent PISA's effort to balance stability and plasticity for each user through two key components: (1) the losses for stability ($\mathcal{L}_{\text{SE}}$) and plasticity ($\mathcal{L}_{\text{PE}}$) enhancement, respectively, and (2) adjusting the degree of those enhancements using the corresponding weights for stability (λ_u^s) and plasticity (λ_u^p), based on each user's preference shifts. In the subsequent subsections, we introduce each component of PISA and relevant analyses in the following order:

- (Section 3.2) A core contribution of PISA is the novel design to explicitly enhance plasticity through $\mathcal{L}_{\text{PE}}$. This involves the careful design of "forward knowledge" ($\mathcal{W}_t^{\text{forward}}$) as reference parameters, providing richer supervision for plasticity. For stability enhancement ($\mathcal{L}_{\text{SE}}$), PISA maintains "backward knowledge" ($\mathcal{W}_t^{\text{backward}}$). To leverage these two types of knowledge, PISA maximizes mutual information between the current user embeddings and their corresponding backward or forward knowledge, as well as the knowledge of neighboring items.
- (Section 3.3) Our theoretical analysis further shows that fine-tuning with plasticity enhancement more effectively mitigates the impact of distribution shifts compared to fine-tuning alone.
- (Section 3.4) To personalize the degree of stability and plasticity enhancements, we introduce user-specific stability and plasticity weights (λ_u^s and λ_u^p), based on individual preference shifts.
- (Section 3.5) Finally, we detail the training procedure based on the final loss function of PISA.

3.2 Stability and Plasticity Enhancements

First, we define backward/forward knowledge and introduce a formulation for adjusting user embeddings toward these types of knowledge. Backward knowledge captures historical context, represented by the parameters from the previous time stage $(t-1)$, while forward knowledge reflects recent as well as latent preferences, obtained via pure fine-tuning on the current data at t, as follows:

$$\mathcal{W}_t^{\text{backward}} := \mathcal{W}_{t-1}, \tag{5}$$

$$\mathcal{W}_t^{\text{forward}} := \text{fine-tune}_{\mathcal{W}_t}\, \mathcal{L}_{\text{rec}}(\mathcal{D}_t; \mathcal{W}_t). \tag{6}$$

Intuitively, backward knowledge ensures that embeddings retain the user's past preferences, preserving long-term behavior. In contrast, forward knowledge captures recent preference shifts and infers latent preferences that may not be directly observable in the data (e.g., similarities between entities in the embedding space or potential interactions). It also accounts for newly emerged users and items, which is absent in backward knowledge, enabling the embeddings to adapt effectively to new information.

To incorporate these references into user embeddings, we maximize a lower bound of the mutual information (MI), known as InfoNCE [10, 20], between the reference knowledge ($\mathcal{W}_t^{\text{backward}}$ or $\mathcal{W}_t^{\text{forward}}$) and the user embeddings. Specifically, for each user u, we minimize the negative mutual information between its embedding and (1) its self-corresponding reference, and (2) the references of its neighboring items. The first part involves minimizing $\mathcal{L}_{\text{MI-S}}$, referred to as the self-reference-loss:

$$\mathcal{L}_{\text{MI-S}}(\mathcal{W}_t, \mathcal{W}_t^{\text{ref}}; u) = -\log \frac{\exp(\text{sim}(\mathbf{e}_u^{\text{ref}}, \mathbf{e}_u^t))}{\frac{1}{|\hat{\mathcal{U}}|} \sum_{u' \in \hat{\mathcal{U}}}^{|\hat{\mathcal{U}}|} \exp(\text{sim}(\mathbf{e}_u^{\text{ref}}, \mathbf{e}_{u'}^t))}, \tag{7}$$

where ref $\in$ {backward, forward}, $\mathbf{e}_u^{\text{ref}}$ is the embedding for user u derived from the reference knowledge, $\text{sim}(\cdot, \cdot)$ denotes a similarity function (e.g., cosine similarity), and $\hat{\mathcal{U}}$ represents the set of negative users sampled from $\mathcal{U}_{:t}$. The second part involves minimizing $\mathcal{L}_{\text{MI-N}}$, referred to as the neighboring-reference-loss.

$$\mathcal{L}_{\text{MI-N}}(\mathcal{W}_t, \mathcal{W}_t^{\text{ref}}; u) = \sum_{i \in \mathcal{N}(u)} -\log \frac{\exp(\text{sim}(\mathbf{e}_i^{\text{ref}}, \mathbf{e}_u^t))}{\frac{1}{|\hat{\mathcal{U}}|} \sum_{u' \in \hat{\mathcal{U}}}^{|\hat{\mathcal{U}}|} \exp(\text{sim}(\mathbf{e}_i^{\text{ref}}, \mathbf{e}_{u'}^t))}, \tag{8}$$

where $\mathcal{N}(u)$ is the set of neighboring items of user u up to time t. These neighboring item references provide richer context to refine the user embedding beyond its self-reference. Combining these components, the final stability and plasticity enhancement losses (from Eq. (3)) are:

$$\mathcal{L}_{\text{SE}} = \mathcal{L}_{\text{MI-S}}(\mathcal{W}_t, \mathcal{W}_t^{\text{backward}}; u) + \mathcal{L}_{\text{MI-N}}(\mathcal{W}_t, \mathcal{W}_t^{\text{backward}}; \mathcal{N}(u)), \tag{9}$$

$$\mathcal{L}_{\text{PE}} = \mathcal{L}_{\text{MI-S}}(\mathcal{W}_t, \mathcal{W}_t^{\text{forward}}; u) + \mathcal{L}_{\text{MI-N}}(\mathcal{W}_t, \mathcal{W}_t^{\text{forward}}; \mathcal{N}(u)). \tag{10}$$

With these losses, we can guide user embeddings toward stability or plasticity as needed.

3.3 Theory on Plasticity Enhancement

Plasticity enhancement is a novel component introduced by PISA. We provide a theoretical justification for its effectiveness in addressing distribution shifts. Specifically, we build on Theorem 3.1 of [30], which provides a generalization bound for fine-tuning under distribution shift and shows that fine-tuning outperforms full retraining. We extend this result by demonstrating that *fine-tuning with plasticity enhancement* can be even more effective than fine-tuning alone.

Assumption 1 (Losses and Data Size). *Consider a time-horizon of t_{te} (test time) stages. Suppose the model is trained on $\mathcal{D}_0 \cup \cdots \cup \mathcal{D}_{t_{\text{te}}-1}$ and tested on $\mathcal{D}_{t_{\text{te}}}$.*

- *Let $\mathcal{L}_t(\mathcal{W}) := \mathbb{E}_{\mathcal{D}_t}\left[\mathcal{L}_{\text{rec}}^{\mathcal{D}_t}(\mathcal{W})\right]$ denote the true generalization error of the recommendation loss.*
- *Let $\mathcal{L}_t^* := \inf_{\mathcal{W}} \mathcal{L}_t(\mathcal{W})$ denote the optimal $\mathcal{L}_t(\mathcal{W})$ loss value.*
- *For each time stage t, let $m_t := |\mathcal{E}_t|$ be the size of dataset $\mathcal{D}_t$. In realistic scenarios, we assume $m_1 = \cdots = m_{t_{\text{te}}-1} \ll m_0$, meaning the pre-training dataset $\mathcal{D}_0$ is significantly larger than each of the incremental data blocks.*

(See Appendix A.1 for further assumptions on distribution shifts, proximal fine-tuning, and plasticity enhancement.)

We use a theoretical measure of the distribution shift between a given stage t and the test time t_{te}, denoted by $d_{t,t_{\text{te}}}$. This measure captures both *covariate shift* (e.g., changes in user/item attributes) and *concept drift* (e.g., changes in user preferences). See Definition 1 in Section A.1 for details. In addition, we assume that larger time gaps lead to larger distribution shifts: $d_{0,t_{\text{te}}} > d_{1,t_{\text{te}}} > \cdots > d_{t_{\text{te}}-1,t_{\text{te}}}$. Using this measure of distribution shift, we theoretically analyze the generalization bound of fine-tuning with plasticity enhancement, and compare it to that of fine-tuning alone (Theorem 3.1).

Theorem 3.1 (Fine-tuning with Plasticity Enhancement). *Let $\mathcal{L}_{t_{\text{te}}}(\beta)$ denote the recommendation loss (e.g., BPR loss) on the test set $\mathcal{D}_{t_{\text{te}}}$ when the pre-trained model is fine-tuned with $\mathcal{L}_{\text{rec}} + \beta \cdot \mathcal{L}_{\text{PE}}$ on $\mathcal{D}_1, \ldots, \mathcal{D}_{t_{\text{te}}-1}$. Suppose that at each time period $t \geq 1$, the number of fine-tuning epochs is chosen according to the proximity assumption [22] with some $0 < \gamma < 1$, reflecting the influence of prior knowledge on new training. Under assumptions in Appendix A.1, including a constant $C > 0$, with probability at least $1 - \delta$,*

$$\mathcal{L}_{t_{\text{te}}}(\beta) \leq \mathcal{L}_{t_{\text{te}}}^* + 2\tilde{\gamma}^{t_{\text{te}}-1} d_{0,t_{\text{te}}} + 2\sum_{t=1}^{t_{\text{te}}-1}(1-\tilde{\gamma})\tilde{\gamma}^{t_{\text{te}}-t-1} d_{t,t_{\text{te}}}$$
$$+ 4\sqrt{\left(\frac{\tilde{\gamma}^{2t_{\text{te}}-2}}{\frac{m_0}{\log m_0}} + \frac{(1+\tilde{\gamma})(1-\tilde{\gamma}^{2t_{\text{te}}-4})}{(1-\tilde{\gamma})\frac{m_1}{\log m_1}}\right)\log\frac{2}{\delta}}, \text{ where } \tilde{\gamma} = \frac{\gamma}{1 + C\beta + \gamma}. \tag{11}$$

Note that $\mathcal{L}_{t_{\text{te}}}(0)$ corresponds to the test loss when the pre-trained model is fine-tuned only with $\mathcal{L}_{\text{rec}}$ (i.e., pure fine-tuning). Thus, fine-tuning with plasticity enhancement (i.e., $\beta > 0$, thus smaller $\tilde{\gamma}$) more effectively mitigates the distribution shifts $\{d_{t,t_{\text{te}}}\}_{t=0}^{t_{\text{te}}-1}$ than pure fine-tuning alone (i.e., $\beta = 0$, thus bigger $\tilde{\gamma}$).

Proof of Theorem 3.1 is in Appendix A.2. Note that the conclusion of Theorem 3.1 assume significant distribution shifts for all users over time. However, real-world scenarios often involve users with varying degrees of preference shifts. Thus, we further quantify the degree of these shifts and use them to weight stability/plasticity enhancements, as detailed in the next subsection.

3.4 Balancing Stability and Plasticity Using Preference Shifts

We quantify preference shifts for each user, and personalize the balance between stability and plasticity based on those preference shifts. Intuitively, users with larger shifts lean toward plasticity, while those with smaller shifts favor stability.

3.4.1 Quantifying Preference Shifts. In PISA, preference shifts are measured by tracking changes in user distances to item clusters across successive time steps. Formally, the preference vector of user u at time t ($\mathbf{p}_u^t \in \mathbb{R}^{1\times m}$) is defined as:

$$\mathbf{p}_u^t := [\mathbf{e}_u^t \cdot (\mathbf{c}_1)^\top, \cdots \mathbf{e}_u^t \cdot (\mathbf{c}_m)^\top], \tag{12}$$

where $\mathbf{e}_u^t \in \mathbb{R}^{1\times d}$ represents the user embedding of u at t with dimension d, derived from the current model parameters $\mathcal{W}_t$, and $[\mathbf{c}_1, \cdots, \mathbf{c}_m]$ are m cluster centroids of current item embeddings, $\{\mathbf{e}_i^t\}_{i\in\mathcal{I}_t}$, obtained via an off-the-shelf clustering method such as K-means, with each centroid having the same d as the user embedding.

Then, the plasticity weight λ_u^p, proportional to the degree of preference shifts, is defined as the sigmoid of the mean-adjusted Jensen-Shannon Divergence (JSD) [8, 18] between preference vector at t and $t-1$; and the stability weight λ_u^s complements λ_u^p as follows:

$$\lambda_u^p := \sigma(\text{JSD}(\mathbf{p}_u^t, \mathbf{p}_u^{t-1}) - \mathbb{E}_{u\in\mathcal{U}}[\text{JSD}(\mathbf{p}_u^t, \mathbf{p}_u^{t-1})]), \tag{13}$$

$$\lambda_u^s := 1 - \lambda_u^p, \tag{14}$$

where $\sigma(\cdot)$ is a sigmoid function. When successive preference vectors are identical, the JSD is 0, while larger shifts result in higher JSD values. The sigmoid function ensures that λ_u^p lies within $[0, 1]$. By subtracting the mean JSD across users, the median plasticity weight is normalized to 0.5. This adjustment allows PISA to differentiate effectively between dynamic users ($\lambda_u^p > 0.5$) and stable users ($\lambda_u^p < 0.5$). The complementary condition $\lambda_u^p + \lambda_u^s = 1$ balances plasticity and stability.

3.4.2 Balancing Stability and Plasticity. PISA leverages these weights to capture both short-term and long-term preferences while reflecting their relative importance. A key challenge in continual recommendation is handling the diverse spectrum of user behaviors: highly dynamic users, highly stable users, and intermediate users. Therefore, applying these weights universally may be suboptimal, especially when many users exhibit intermediate behavior, with only a small fraction being highly dynamic or stable. To address this, we propose selectively applying weights to the top-$L\%$ most dynamic and stable users:

$$\lambda_u^p := \begin{cases} \lambda_u^p & \text{if } \lambda_u^p \geq \text{Top-}L\%(\lambda^p) \\ 0 & \text{otherwise,} \end{cases} \quad \lambda_u^s := \begin{cases} \lambda_u^s & \text{if } \lambda_u^s \geq \text{Top-}L\%(\lambda^s) \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

where L is a hyperparameter. This approach offers two key benefits. First, for highly dynamic or stable users, adjustments focus exclusively on plasticity or stability, ensuring targeted optimization. Second, for intermediate users, it avoids applying either weight, relying instead on fine-tuning to prevent over-adjustment and mitigate uncertainty. This is possible because fine-tuning inherently achieves a partial balance between stability and plasticity by inheriting parameters and updating them with new data. Note that when $L = 100$, PISA applies both weights to all users.

3.5 Training Procedure

The overall procedure is detailed in Algorithm 1. Here, we highlight the essential aspects of the training process. The current model parameters to be optimized, $\mathcal{W}_t$, and the backward knowledge, $\mathcal{W}_t^{\text{backward}}$, are both initialized with the previous model parameters, $\mathcal{W}_{t-1}$ (Line 3). Forward knowledge, $\mathcal{W}_t^{\text{forward}}$, is obtained through one full process of pure fine-tuning on the current dataset $\mathcal{D}_t$ (Line 4). At each epoch, clustering is performed on the current item embeddings, $\{\mathbf{e}_i^t\}_{i \in \mathcal{I}_{:t}}$, to yield up-to-date centroids that capture evolving item embeddings (Line 6).

Initially, since the current model parameters and backward knowledge are identical ($\mathbf{e}_u^t = \mathbf{e}_u^{t-1}$), the preference vectors are also identical ($\mathbf{p}_u^t = \mathbf{p}_u^{t-1}$). This leads to equal stability and plasticity weights, $\lambda_u^s = \lambda_u^p = 0.5$ (before applying top-L% selection). As training progresses, these weights adjust to reflect users' preference shifts, as captured by the evolving embeddings. The top-L% selection further refines these weights, targeting adjustments for only highly dynamic or stable users (Lines 18–19). This automatic adjustment enables PISA to reflect each user's changing behavior over time. By integrating these weights (λ_u^s and λ_u^p) with the stability/plasticity enhancement losses ($\mathcal{L}_{\text{SE}}$ and $\mathcal{L}_{\text{PE}}$) (Line 22), along with fine-tuning using the recommendation loss ($\mathcal{L}_{\text{rec}}$) (Line 21), PISA effectively balances stability and plasticity in the embeddings (Line 23).

Algorithm 1 Training procedure of PISA at time stage t

1: **Input:** Model parameters $\mathcal{W}_{t-1}$, scaling parameter α, incoming dataset $\mathcal{D}_t = \{\mathcal{U}_t, \mathcal{I}_t, \mathcal{E}_t, \mathrm{Y}_t\}$
2: **Output:** Updated model parameters $\mathcal{W}_t$
3: $\mathcal{W}_t \leftarrow \mathcal{W}_{t-1}$; $\mathcal{W}_t^{\text{backward}} \leftarrow \mathcal{W}_{t-1}$;
4: $\mathcal{W}_t^{\text{forward}} \leftarrow$ fine-tune$_{\mathcal{W}_t}$ $\mathcal{L}_{\text{rec}}(\mathcal{D}_t; \mathcal{W}_t)$;
5: **for** epoch **do**
6:     Perform clustering of $\mathcal{W}_t(\mathcal{I})$ and get $[\mathbf{c}_1, \cdots, \mathbf{c}_m]$;
7:     **for** mini-batch $\mathcal{B}$ obtained from $\mathcal{E}_t$ **do**
8:         Negative interaction set $\mathcal{M} \leftarrow \{\}$;
9:         User set $\mathcal{U} \leftarrow \{\}$;
10:         Negative user set $\hat{\mathcal{U}} \leftarrow \{\}$;
11:         **for** user-item interaction $(u, i) \in \mathcal{B}$ **do**
12:             Sample negative items as $\mathcal{M}_{ui}$;
13:             Update $\mathcal{M} \leftarrow \mathcal{M} \cup \{(u, i')\}_{i' \in \mathcal{M}_{ui}}$;
14:             Update $\mathcal{U} \leftarrow \mathcal{U} \cup \{u\}$
15:         **end for**
16:         **for** user $u \in \mathcal{U}$ **do**
17:             Sample negative users and add to $\hat{\mathcal{U}}$;
18:             Compute stability weight λ_u^s by Eq. (12)-(15);
19:             Compute plasticity weight λ_p^s by Eq. (12)-(15);
20:         **end for**
21:         Compute the rec loss $\mathcal{L}_{\text{rec}}$ with $\mathcal{B} \cup \mathcal{M}$ by Eq. (4);
22:         Compute $\mathcal{L}_{\text{SE}}$ and $\mathcal{L}_{\text{PE}}$ with $\mathcal{U} \cup \hat{\mathcal{U}} \cup \mathcal{N}(u)$ by Eq. (9) and Eq. (10);
23:         Update $\mathcal{W}_t$ based on α, $\mathcal{L}_{\text{rec}}$, λ_u^s, $\mathcal{L}_{\text{SE}}$, λ_u^p, $\mathcal{L}_{\text{PE}}$;
24:     **end for**
25: **end for**

4 Experiments

We design experiments to answer the key research questions (RQs):

RQ1. To what extent does PISA outperform its competitors?
RQ2. How does PISA benefit both stable and dynamic users?
RQ3. How does each component in PISA affects its performance?

4.1 Experimental Settings

4.1.1 Datasets. We use the following real-world temporal datasets.

- **Amazon (Video Games)**[3]: This contains 93,471 ratings provided by 9,826 users on 16,172 products in the Video Games category. It spans 2,064 days, from Jan. 1, 2018, to Aug. 27, 2023. The avg. number of new users/items per new data block is 373/570.
- **Amazon (CDs and Vinyls)**: This contains 74,093 ratings provided by 5,488 users on 14,622 products in the CDs and Vinyls category. It spans 2,050 days, from Jan. 1, 2018, to Aug. 13, 2023. The avg. number of new users/items per new data block is 182/985.
- **Gowalla**[4]: This location-based social network dataset contains 1,716,274 check-in interactions shared by 61,302 users on 636,558 locations. It spans 52 days, from Sept. 1, 2010, to Oct. 23, 2010. The avg. number of new users/items per new data block is 1,725/36,826.

[3]https://amazon-reviews-2023.github.io/
[4]https://snap.stanford.edu/data/loc-gowalla.html

Following [1, 26, 27], we preprocess the Amazon datasets by binarizing the 5-star ratings. Specifically, we set $\mathrm{Y}[u, i] = 1$ if user u rates item i higher than 2, and $\mathrm{Y}[u, i] = 0$ otherwise. We use the latest 5.5 years of data, starting from January 1 2018, and remove users or items with fewer than ten interactions. To simulate continual learning, we sort the interactions in each dataset $\mathcal{D}$ chronologically. Following [1, 26, 27], we split 60% of the interactions as pre-training data $\mathcal{D}_0$ and the remaining 40% as incremental data. The incremental data is further divided into five stages ($T = 5$), each containing an equal number of interactions, resulting in $\{\mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_T\}$. For evaluation, after training the model on $\mathcal{D}_t$ ($t = 0, ..., T-1$), we use the first half of the next incremental data block $\mathcal{D}_{t+1}$ as the validation set and the second half as the test set.

4.1.2 Compared methods. We compare PISA with several competitors designed to handle incremental data effectively in continual learning settings. For a fair comparison, we use LightGCN [12] as the base model for all methods, and employ the Bayesian Personalized Ranking (BPR) loss [23] as the recommendation loss $\mathcal{L}_{\text{rec}}$.

We first include basic baselines: full-retraining and fine-tuning.

- Retraining: This refers to retraining with all historical and new data $\{\mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_t\}$ with $\mathcal{L}_{\text{rec}}$ at each time stage t.
- Fine-tuning: This fine-tunes only with new data $\mathcal{D}_t$ with $\mathcal{L}_{\text{rec}}$.

The continual learning competitors are divided into two categories. First, experience replay-based competitors aim to enhance stability by selectively reusing past input data:

- UNIFORM [21]: This method samples a fixed number of past interactions uniformly in random.
- INV-DEGREE [1]: This method samples past interactions with selection probabilities proportional to the inverse degree of their users, prioritizing stability for nodes with fewer interactions.

Second, knowledge distillation-based competitors aim to enhance stability by integrating past model parameters through distillation:

- GRAPHSAIL [27]: It integrates three types of past knowledge for each user–local structure, global structure, and self-information–into the current model.
- LWC-KD [25]: It integrates past model parameters of each user and their neighbors into the current model using contrastive distillation loss.
- LWC-KD-PIW [26]: It builds upon LWC-KD by using adaptively learned personalized weights for knowledge distillation for each user. These weights are trained via a multi-layer perceptron that captures user preference shifts in an end-to-end manner.

For a fair comparison, all knowledge distillation-based competitors, as well as PISA, employ only user-side distillation, excluding item-side distillation. Additionally, to conduct an ablation study of PISA, we introduce two variants of PISA as follows:

- PISA(S): A special version of PISA that only uses stability enhancement loss (Eq. (9)) with fine-tuning.
- PISA(P): A special version of PISA that only uses plasticity enhancement loss (Eq. (10)) with fine-tuning.

For the main comparison, we report both PISA and PISA(P) as our proposed methods that employs explicit plasticity enhancement, which is a novel contribution of this work.

4.1.3 Evaluation metrics. We use the all-item-ranking method, where all items a user has not interacted with are treated as candidate items for recommendation, to evaluate top-N recommendation accuracy [14]. The evaluation metrics include Recall@N and NDCG@N (normalized discounted cumulative gain), with $N = 10$ or 20. We report results averaged over all T incremental data blocks.

4.1.4 Implementation details. For all compared methods, we ensure consistency by using four negative user-item pairs for each positive user-item pair in the BPR loss, setting the learning rate to 0.001, and applying L2 regularization with a value of 0.00001. Model parameters are optimized using the Adam algorithm [13], and the dimension of user and item embeddings is set to 64.

For the experience replay-based method, we use a subset of past data equal in size to the incremental data block (i.e., $\gamma = 1$ in Eq. (1)). For the knowledge distillation-based method, we choose the global weight for the knowledge distillation loss from [0.1,0.5,1.0] (i.e., λ in Eq. (2)). Note that LWC-KD-PIW automatically learns the personalized weight for different users. For PISA and PISA(P), we choose L values from [0.2, 0.4, 0.6, 0.8, 1.0] and α values from [0.1,0.5,1.0]. Specifically, we set $L = 1.0$ for Amazon (CDs and Vinyl), and $L = 0.2$ for Amazon (Video Games) and Gowalla, with $\alpha = 0.5$ for all cases in both PISA and PISA(P). To ensure reproducibility, we use five fixed random seeds for experimentation and report the mean and standard deviation values across these five runs. The source code will be released upon the publication of the paper.

4.2 Main Results

To address RQ1, we compare PISA with all seven competitors, evaluating their recommendation performance in continual settings. Table 3 presents the results across four metrics and three datasets. For clarity, we use double-underline, single-underline, and bold text to highlight the 1st, 2nd, and 3rd best results, respectively. First, regarding the basic baselines, FINE-TUNING consistently outperforms RETRAINING, demonstrating the inherent adaptability of fine-tuning in capturing continuously evolving user preferences. This aligns with the motivational analyses in Figure 2 and Table 2.

Second, although several competitors surpass FINE-TUNING, none consistently dominates across all metrics and datasets. In contrast, PISA (or PISA(P)) consistently outperforms all competitors, achieving an average improvement of 6.79% over the *best* competitor in each case, with specific improvements of 11.68% and 7.25% over INV-DEGREE and LWC-KD-PIW, respectively. Existing continual learning approaches—whether based on experience replay or knowledge distillation—exhibit two key limitations: (1) they focus primarily on enhancing stability, neglecting plasticity, and (2) they apply this enhancement uniformly across all users. Although LWC-KD-PIW introduces personalized weights for users, its personalization is limited to stability enhancement, falling short to explicitly address plasticity. In contrast, PISA effectively balances stability and plasticity based on stability and plasticity losses combined with dynamic user-specific adjustments.

Lastly, in some cases, PISA(P) is comparable to or slightly outperforms PISA for certain metrics on Amazon (Video Games) and Gowalla. This reveals two insights: (1) Even without explicit stability enhancement, fine-tuning alone can preserve stability to some extent through parameter inheritance. For example, FINE-TUNING outperforms all competitors in R@20 and N@20 on Amazon (Video Games). When combined with plasticity enhancement, this achieves an effective balance between stability and plasticity. (2) Some datasets may contain highly dynamic users who require targeted plasticity enhancement. This observation will be explored further in Section 4.4 (RQ3).

4.3 Results on Stable/Dynamic Users

To answer RQ2, we conduct a fine-grained evaluation of user groups with varying behaviors. Users' preference shifts are measured using Eq.(13) after pure fine-tuning and classified into two groups: stable users (lowest half of shifts) and dynamic users (highest half). This classification is updated at each time stage. Figure 3 shows the results for each user group on Amazon (CDs and Vinyl) and Amazon (Video Games) for FINE-TUNING, two knowledge distillation-based methods (LWC-KD, LWC-KD-PIW), and PISA.

First, the results demonstrate that PISA achieves superior performance for *both* stable and dynamic users on both datasets, compared to LWC-KD and LWC-KD-PIW, which focus solely on stability enhancement. While LWC-KD-PIW provides more benefits to dynamic users than LWC-KD by assigning different weights to stability enhancement, it sometimes sacrifices performance for stable users (e.g., on Amazon (CDs and Vinyl)), indicating a sub-optimal balance between stability and plasticity. In contrast, PISA effectively balances these two aspects. Moreover, PISA outperforms FINE-TUNING in most cases (with one slight exception) while the other competitors perform worse than FINE-TUNING for either the

Table 3: Recommendation performance averaged across time stages for PISA, PISA(P), and seven competitors. PISA or PISA(P) consistently outperforms all competitors across all metrics on both datasets, showcasing its superior ability to balance between stability and plasticity for each user in continual scenarios (values are in %, e.g., 0.71±0.03 represents 0.0071 with 0.0003 stdev.).

	Amazon (CDs and Vinyls)				Amazon (Video Games)				Gowalla			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
Retraining	0.71±0.03	0.46±0.01	1.40±0.03	0.66±0.01	0.88±0.05	0.55±0.03	1.44±0.09	0.73±0.03	2.15±0.02	1.57±0.01	3.07±0.02	1.86±0.01
Fine-tuning	0.86±0.07	0.56±0.05	1.39±0.11	0.72±0.05	1.38±0.10	0.93±0.04	**2.42±0.14**	**1.24±0.04**	2.69±0.03	2.01±0.02	3.99±0.01	2.41±0.01
Uniform	0.90±0.07	0.55±0.03	1.50±0.05	0.73±0.03	1.22±0.06	0.72±0.02	2.07±0.16	0.98±0.04	2.60±0.03	1.92±0.01	3.79±0.01	2.29±0.01
Inv-degree	**0.93±0.04**	**0.63±0.02**	1.49±0.05	0.80±0.03	1.29±0.07	0.81±0.06	2.14±0.11	1.07±0.07	2.87±0.02	2.11±0.02	**4.20±0.02**	2.52±0.02
GraphSAIL	0.77±0.09	0.49±0.06	1.31±0.07	0.65±0.05	1.26±0.13	0.79±0.07	2.14±0.14	1.05±0.08	1.98±0.00	1.44±0.01	2.98±0.01	1.75±0.00
LWC-KD	0.92±0.08	0.60±0.04	1.47±0.10	0.77±0.04	1.35±0.20	0.91±0.10	2.03±0.11	1.11±0.07	2.55±0.04	1.89±0.04	3.74±0.07	2.26±0.04
LWC-KD-PIW	**0.95±0.05**	0.60±0.02	**1.53±0.08**	**0.77±0.03**	**1.42±0.11**	**0.94±0.04**	2.31±0.14	1.22±0.05	**2.88±0.02**	**2.16±0.00**	4.19±0.01	**2.56±0.00**
PISA(P)	0.92±0.08	**0.63±0.02**	**1.52±0.10**	**0.81±0.03**	**1.61±0.04**	**1.07±0.03**	**2.40±0.12**	**1.32±0.06**	**2.91±0.04**	**2.16±0.03**	**4.35±0.04**	**2.60±0.02**
PISA	**1.03±0.09**	**0.67±0.03**	**1.65±0.07**	**0.86±0.02**	**1.57±0.10**	**1.06±0.06**	**2.42±0.14**	**1.33±0.07**	**2.90±0.01**	**2.13±0.01**	**4.26±0.03**	**2.55±0.00**
Imp. / best	8.42%	6.35%	7.84%	11.69%	13.38%	13.83%	4.76%	9.02%	1.04%	0.00%	3.57%	1.56%

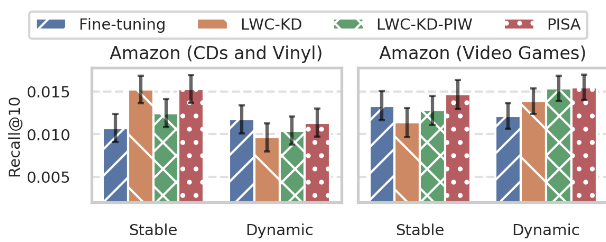

Figure 3: Performance comparison for stable and dynamic user groups. PISA consistently outperforms the stability-focused competitors (LWC-KD and LWC-KD-PIW) for both user groups, highlighting the advantage of explicitly enhancing both stability and plasticity.

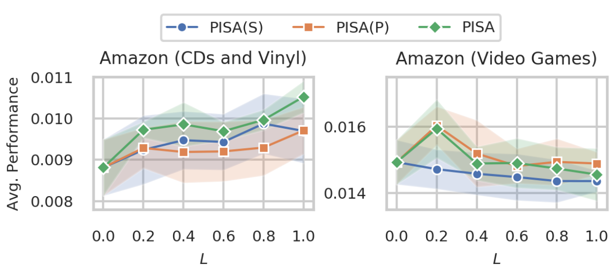

Figure 4: Impact of the weight selection parameter (L) on the performance of PISA(S), PISA(P), and PISA.

stable or dynamic group. This further confirms PISA's effectiveness in achieving balance within the fine-tuning training strategy.

4.4 Component/Hyperparameter Analysis

To answer RQ3, we present the average performance across all four metrics (R@10, N@10, R@20, N@20) for PISA(S), PISA(P), and PISA with varying values of weight selection parameter L (Eq.(15)), where $L \in [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$, in Figure 4. Here, $L = 0$ represents pure fine-tuning without stability or plasticity losses.

For PISA, the results show that $L > 0$ consistently outperforms $L = 0$, indicating that incorporating stability and plasticity enhancement losses improves the effectiveness of fine-tuning. On Amazon (CDs and Vinyl), as L increases, the performance of PISA tends to steadily improve. In contrast, on Amazon (Video Games), performance tends to decrease as L increases.

The steady changes suggest that the stability and plasticity weights accurately capture users' preference shifts. Additionally, the differing trends between datasets provide further insights. On Amazon (CDs and Vinyl), the distribution of users' preference shifts appears to be relatively balanced, with no extreme outliers in terms of stability or dynamism. This allows stability and plasticity enhancements to benefit all users synergistically, resulting in PISA outperforming both PISA(S) and PISA(P).

In contrast, for Amazon (Video Games), the dataset likely includes a small subset of highly dynamic users alongside intermediate users. This is evidenced by two observations: (1) the best performance occurs when plasticity enhancement is applied only to the top 20% most dynamic users ($L = 0.2$), and (2) while PISA(P) performs comparably to PISA, PISA(S) performs significantly worse. Notably, PISA(S) performs even worse than pure fine-tuning, highlighting the importance of explicitly enhancing plasticity to adapt to user preference shifts or dataset-specific characteristics.

5 Conclusion

In this work, we highlight the importance of balancing stability and plasticity in continual recommender systems. To address the limitations of existing methods that focus only on stability, we propose PISA, a novel framework that adaptively enhances both stability and plasticity based on user preference shifts. By introducing forward knowledge to guide plasticity and backward knowledge to maintain stability, PISA dynamically adjusts user embeddings to optimize recommendation quality for both stable and dynamic users. Our theoretical analysis demonstrates the benefits of plasticity enhancement in mitigating distribution shifts, while extensive experiments on real-world datasets confirm the superiority of PISA over existing methods, validating its effectiveness in balancing stability and plasticity for each user.

A Theoretical Analyses

A.1 Assumptions and Setup

Our main theorem builds on Theorem 3.1 of [30], which provides a generalization error for fine-tuning under distribution shift and shows its advantage over full retraining. We extend this result to show that *fine-tuning with plasticity enhancement* is even more effective than fine-tuning alone.

Definition 1 (Measures of Distribution Shifts). *Distribution shifts over time can arise from two main sources:*

- *Covariate shift: Changes in the distribution of user/item attributes (i.e., the distribution of $(\mathcal{U}_t, \mathcal{I}_t, \mathcal{E}_t)$).*
- *Concept drift: Changes in user preferences (i.e., the conditional distribution $\Upsilon_t \mid (\mathcal{U}_t, \mathcal{I}_t, \mathcal{E}_t)$).*

A classic measure of covariate shift is the discrepancy distance [17], a generalization of the $\mathcal{H}\Delta\mathcal{H}$ distance [4]:

$$d^{\mathcal{H}\Delta\mathcal{H}}_{t,t_{\mathrm{te}}} := \sup_{\mathcal{W},\mathcal{W}'} \Big| \big(\mathcal{L}_t(\mathcal{W}) - \mathcal{L}_t(\mathcal{W}')\big) - \big(\mathcal{L}_{t_{\mathrm{te}}}(\mathcal{W}) - \mathcal{L}_{t_{\mathrm{te}}}(\mathcal{W}')\big)\Big|.$$

If no covariate shift occurs between t and t_{te}, then for any two models $\mathcal{W}, \mathcal{W}'$, the difference in $\mathcal{L}$ remains unchanged, leading to $d^{\mathcal{H}\Delta\mathcal{H}}_{t,t_{\mathrm{te}}}=0$.

A classic measure of concept drift is the combined error [4]:

$$d^{\mathrm{comb}}_{t,t_{\mathrm{te}}} := \inf_{\mathcal{W}} \big(\mathcal{L}_t(\mathcal{W}) + \mathcal{L}_{t_{\mathrm{te}}}(\mathcal{W})\big) - \mathcal{L}^*_t - \mathcal{L}^*_{t_{\mathrm{te}}}.$$

If no concept drift exists, then $d^{\mathrm{comb}}_{t,t_{\mathrm{te}}} = 0$ because a single model can achieve the optimal losses at both times. Combining these, we define the unified distribution shift as follows: $d_{t,t_{\mathrm{te}}} := d^{\mathcal{H}\Delta\mathcal{H}}_{t,t_{\mathrm{te}}} + d^{\mathrm{comb}}_{t,t_{\mathrm{te}}}$.

Assumption 2 (Distribution Shifts Over Time). *Distribution shifts grow with larger time gaps, i.e., $d_{0,t_{\mathrm{te}}} > d_{1,t_{\mathrm{te}}} > \cdots > d_{t_{\mathrm{te}}-1,t_{\mathrm{te}}}$.*

Assumption 3 (Loss with PE). *For training, we use both recommendation and plasticity enhancement terms: for $t \geq 1$,*

$$\mathcal{L}^{\mathcal{D}_t}_{\mathrm{rec}}(\mathcal{W}) + \beta \cdot \mathcal{L}^{\mathcal{D}_t}_{\mathrm{PE}}(\mathcal{W}), \tag{16}$$

where $\mathcal{L}^{\mathcal{D}_t}_{\mathrm{PE}}(\mathcal{W})$ is the plasticity enhancement loss that mitigates distribution shifts by aligning user embeddings to forward knowledge, and $\beta \geq 0$ is the weight of the plasticity enhancement loss. When $\beta = 0$, the loss function is the pure recommendation loss.

Assumption 4 (Plasticity). *We assume that the plasticity enhancement loss $\mathcal{L}^{\mathcal{D}_t}_{\mathrm{PE}}$ is positively correlated with the recommendation loss $\mathcal{L}^{\mathcal{D}_t}_{\mathrm{rec}}$ at the same time step. Formally, there exists a constant $C > 0$ such that, for all t and all $\mathcal{W}$,*

$$\mathcal{L}^{\mathcal{D}_t}_{\mathrm{PE}}(\mathcal{W}) \geq C\mathcal{L}^{\mathcal{D}_t}_{\mathrm{rec}}(\mathcal{W}). \tag{17}$$

Under this assumption, the loss with PE can be lower bounded as

$$\mathcal{L}^{\mathcal{D}_t}_{\mathrm{rec}}(\mathcal{W}) + \beta \cdot \mathcal{L}^{\mathcal{D}_t}_{\mathrm{PE}}(\mathcal{W}) \geq (1 + C\beta)\mathcal{L}_{\mathrm{rec}}(\mathcal{W}). \tag{18}$$

Thus, fine-tuning with plasticity enhancement can be viewed as implicitly optimizing $(1 + C\beta)$ times the recommendation loss.

Assumption 5 (Proximal Fine-tuning). *We adopt the proximal fine-tuning assumption from [22, 30]. For each $t \geq 1$, let $\mathcal{W}^{\mathrm{ft}}_t$ denote the model parameters obtained by fine-tuning on $\mathcal{D}_t$. To ensure that $\mathcal{W}^{\mathrm{ft}}_t$ does not completely forget knowledge from $\mathcal{W}^{\mathrm{ft}}_{t-1}$, we assume all time periods share the same parameter space. Furthermore, there is a fixed $0 < \gamma < 1$, reflecting the influence of prior knowledge on new training, such that, for each $t \geq 1$, the number of fine-tuning epochs is chosen so that $\mathcal{W}^{\mathrm{ft}}_t$ minimizes*

$$\ell_t(\mathcal{W}) := \frac{(1 + C\beta)\mathcal{L}^{\mathcal{D}_t}_{\mathrm{rec}}(\mathcal{W}) + \gamma\, \ell_{t-1}(\mathcal{W})}{1 + C\beta + \gamma}, \tag{19}$$

where $\ell_0(\mathcal{W}) := \mathcal{L}^{\mathcal{D}_0}_{\mathrm{rec}}(\mathcal{W})$ is the pre-training loss.

A.2 Proof of Theorem 3.1

Our proof of Theorem 3.1 relies on Lemma A.1 and Corollary A.2 from [30], which provide bounds on the generalization loss under distribution shifts. We restate these results here for completeness; see [30] for additional details.

Lemma A.1. *Let $\boldsymbol{\alpha} \in \mathbb{R}^{t_{\mathrm{te}}}_{\geq 0}$ satisfy $\sum_{t=0}^{t_{\mathrm{te}}-1} \alpha_t = 1$, and fix any $\epsilon > 0$. Suppose there exist model parameters $\mathcal{W}^{\boldsymbol{\alpha},\epsilon}_{t_{\mathrm{te}}-1}$ such that*

$$\sum_{t=0}^{t_{\mathrm{te}}-1} \alpha_t\, \mathcal{L}_t(\mathcal{W}^{\boldsymbol{\alpha},\epsilon}_{t_{\mathrm{te}}-1}) \leq \epsilon + \inf_{\mathcal{W}} \sum_{t=0}^{t_{\mathrm{te}}-1} \alpha_t\, \mathcal{L}^{\mathcal{D}_t}_{\mathrm{rec}}(\mathcal{W}).$$

Then, with probability at least $1 - \delta$,

$$\mathcal{L}_{t_{\mathrm{te}}}(\mathcal{W}^{\boldsymbol{\alpha},\epsilon}_{t_{\mathrm{te}}-1}) \leq \mathcal{L}^*_{t_{\mathrm{te}}} + \epsilon + 2\sum_{t=0}^{t_{\mathrm{te}}-1} \alpha_t\, d_{t,t_{\mathrm{te}}} + 4\sqrt{\sum_{t=0}^{t_{\mathrm{te}}-1} \frac{\alpha_t^2}{\frac{m_t}{\log m_t}} \log\frac{2}{\delta}}.$$

Corollary A.2. *Under the assumptions of Lemma A.1, define*

$$\mathcal{L}^{\boldsymbol{\alpha}}_{t_{\mathrm{te}}} := \inf_{\epsilon \in \mathbb{Q}_{>0}} \mathcal{L}_{t_{\mathrm{te}}}(\mathcal{W}^{\boldsymbol{\alpha},\epsilon}_{t_{\mathrm{te}}-1}).$$

Then, with probability at least $1 - \delta$,

$$\mathcal{L}^{\boldsymbol{\alpha}}_{t_{\mathrm{te}}} \leq \mathcal{L}^*_{t_{\mathrm{te}}} + 2\sum_{t=0}^{t_{\mathrm{te}}-1} \alpha_t\, d_{t,t_{\mathrm{te}}} + 4\sqrt{\sum_{t=0}^{t_{\mathrm{te}}-1} \frac{\alpha_t^2}{\frac{m_t}{\log m_t}} \log\frac{2}{\delta}}.$$

Proof of Theorem 3.1. We first define, for conciseness,

$$\tilde{\gamma} = \frac{\gamma}{1 + C\beta + \gamma}. \tag{20}$$

Then, by Assumption 5, the loss for fine-tuning with plasticity enhancement at stage $(t_{\mathrm{te}} - 1)$ is

$$\begin{aligned} \ell_{t_{\mathrm{te}}-1}(\mathcal{W}) &= (1 - \tilde{\gamma})\mathcal{L}^{\mathcal{D}_t}_{\mathrm{rec}}(\mathcal{W})\tilde{\gamma}\, \ell_{t_{\mathrm{te}}-2}(\mathcal{W}) \\ &= \tilde{\gamma}^{\,t_{\mathrm{te}}-1}\, \mathcal{L}^{\mathcal{D}_0}_{\mathrm{rec}}(\mathcal{W}) + \sum_{t=1}^{t_{\mathrm{te}}-1} (1 - \tilde{\gamma})\tilde{\gamma}^{\,t_{\mathrm{te}}-t-1}\, \mathcal{L}^{\mathcal{D}_t}_{\mathrm{rec}}(\mathcal{W}). \end{aligned} \tag{21}$$

Thus, the coefficients are given by

$$\alpha_t := \begin{cases} \tilde{\gamma}^{\,t_{\mathrm{te}}-1}, & \text{for } t = 0, \\ (1 - \tilde{\gamma})\tilde{\gamma}^{\,t_{\mathrm{te}}-t-1}, & \text{for } t = 1, \ldots, t_{\mathrm{te}} - 1. \end{cases} \tag{22}$$

Applying Corollary A.2, we obtain

$$\begin{aligned} \mathcal{L}^{\boldsymbol{\alpha}}_{t_{\mathrm{te}}}(\beta) \leq\ & \mathcal{L}^*_{t_{\mathrm{te}}} + 2\tilde{\gamma}^{t_{\mathrm{te}}-1} d_{0,t_{\mathrm{te}}} + 2\sum_{t=1}^{t_{\mathrm{te}}-1} (1 - \tilde{\gamma})\tilde{\gamma}^{t_{\mathrm{te}}-t-1} d_{t,t_{\mathrm{te}}} \\ & + 4\sqrt{\left(\frac{\tilde{\gamma}^{2t_{\mathrm{te}}-2}}{\frac{m_0}{\log m_0}} + \frac{(1 + \tilde{\gamma})(1 - \tilde{\gamma}^{2t_{\mathrm{te}}-4})}{(1 - \tilde{\gamma})\frac{m_1}{\log m_1}}\right) \log\frac{2}{\delta}},\ \text{where } \tilde{\gamma} = \frac{\gamma}{1 + C\beta + \gamma}. \end{aligned} \tag{23}$$

This implies that fine-tuning with plasticity enhancement (i.e., $\beta > 0$, thus smaller $\tilde{\gamma}$) more effectively mitigates the distribution shifts $\{d_{t,t_{\mathrm{te}}}\}_{t=0}^{t_{\mathrm{te}}-1}$ than pure fine-tuning (i.e., $\beta = 0$, thus bigger $\tilde{\gamma}$). □

Acknowledgments

This work is partially supported by NSF (2416070), NIFA (2020-67021-32799), and IBM-Illinois Discovery Accelerator Institute. The content of the information in this document does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] Kian Ahrabian, Yishi Xu, Yingxue Zhang, Jiapeng Wu, Yuening Wang, and Mark Coates. 2021. Structure aware experience replay for incremental learning in graph-based recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2832–2836.

[2] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. 2022. Learning fast, learning slow: A general continual learning method based on complementary learning system. *arXiv preprint arXiv:2201.12604* (2022).

[3] Jiameng Bai, Sai Wu, Jie Song, Junbo Zhao, and Gang Chen. 2024. Pre-Trained Model Recommendation for Downstream Fine-tuning. *arXiv preprint arXiv:2403.06382* (2024).

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning* 79 (2010), 151–175.

[5] Lei Chen, Fajie Yuan, Jiaxi Yang, Xiangnan He, Chengming Li, and Min Yang. 2021. User-specific adaptive fine-tuning for cross-domain recommendations. *IEEE Transactions on Knowledge and Data Engineering* 35, 3 (2021), 3239–3252.

[6] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. 2020. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*. PMLR, 1779–1788.

[7] Jaime Hieu Do and Hady W Lauw. 2023. Continual Collaborative Filtering Through Gradient Alignment. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1133–1138.

[8] Bent Fuglede and Flemming Topsoe. 2004. Jensen-Shannon divergence and Hilbert space embedding. In *International symposium onInformation theory, 2004. ISIT 2004. Proceedings.* IEEE, 31.

[9] Umang Gupta, Aaron M Ferber, Bistra Dilkina, and Greg Ver Steeg. 2021. Controllable guarantees for fair outcomes via contrastive information estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 7610–7619.

[10] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 297–304.

[11] Bowei He, Xu He, Yingxue Zhang, Ruiming Tang, and Chen Ma. 2023. Dynamically expandable graph convolution for streaming recommendation. In *Proceedings of the ACM Web Conference 2023*. 1457–1467.

[12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[14] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 1748–1757.

[15] Gyuseok Lee, SeongKu Kang, Wonbin Kweon, and Hwanjo Yu. 2024. Continual Collaborative Distillation for Recommender System. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1495–1505.

[16] Xinyu Lin, Wenjie Wang, Yongqi Li, Shuo Yang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2024. Data-efficient Fine-tuning for LLM-based Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 365–374.

[17] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. 2009. Domain adaptation: Learning bounds and algorithms. In *Proceedings of The 22nd Annual Conference on Learning Theory*.

[18] María Luisa Menéndez, JA Pardo, L Pardo, and MC Pardo. 1997. The jensen-shannon divergence. *Journal of the Franklin Institute* 334, 2 (1997), 307–318.

[19] Fei Mi, Xiaoyu Lin, and Boi Faltings. 2020. Ader: Adaptively distilled exemplar replay towards continual learning for session-based recommendation. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 408–413.

[20] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[21] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. 2020. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 524–540.

[22] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*, Vol. 32.

[23] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[24] Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. 2021. Afec: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems* 34 (2021), 22379–22391.

[25] Yuening Wang, Yingxue Zhang, and Mark Coates. 2021. Graph structure aware contrastive knowledge distillation for incremental learning in recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3518–3522.

[26] Yuening Wang, Yingxue Zhang, Antonios Valkanas, Ruiming Tang, Chen Ma, Jianye Hao, and Mark Coates. 2023. Structure aware incremental learning with personalized imitation weights for recommender systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 4711–4719.

[27] Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark Coates. 2020. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2861–2868.

[28] Mao Ye, Ruichen Jiang, Haoxiang Wang, Dhruv Choudhary, Xiaocong Du, Bhargav Bhushanam, Aryan Mokhtari, Arun Kejariwal, and Qiang Liu. 2022. Future gradient descent for adapting the temporal shifting data distribution in online recommendation systems. In *Uncertainty in Artificial Intelligence*. PMLR, 2256–2266.

[29] Hyunsik Yoo, Ruizhong Qiu, Charlie Xu, Fei Wang, and Hanghang Tong. 2025. Generalizable recommender system during temporal popularity distribution shifts. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[30] Hyunsik Yoo, Zhichen Zeng, Jian Kang, Ruizhong Qiu, David Zhou, Zhining Liu, Fei Wang, Charlie Xu, Eunice Chan, and Hanghang Tong. 2024. Ensuring user-side fairness in dynamic recommender systems. In *Proceedings of the ACM on Web Conference 2024*. 3667–3678.

[31] Fajie Yuan, Guoxiao Zhang, Alexandros Karatzoglou, Joemon Jose, Beibei Kong, and Yudong Li. 2021. One person, one model, one world: Learning continual user representation without forgetting. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 696–705.

[32] Kexin Zhang, Yichao Wang, Xiu Li, Ruiming Tang, and Rui Zhang. 2024. IncMSR: An Incremental Learning Approach for Multi-Scenario Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*. 939–948.

[33] Peiyan Zhang, Yuchen Yan, Chaozhuo Li, Senzhang Wang, Xing Xie, Guojie Song, and Sunghun Kim. 2023. Continual learning on dynamic graphs via parameter isolation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 601–611.

[34] Xinni Zhang, Yankai Chen, Chenhao Ma, Yixiang Fang, and Irwin King. 2024. Influential Exemplar Replay for Incremental Learning in Recommender Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9368–9376.

[35] Chen Zhao, Le Wu, Pengyang Shao, Kun Zhang, Richang Hong, and Meng Wang. 2023. Fair representation learning for recommendation: A mutual information perspective. In *Proceedings of the AAAI Conference on artificial intelligence*, Vol. 37. 4911–4919.

[36] Jieming Zhu, Guohao Cai, Junjie Huang, Zhenhua Dong, Ruiming Tang, and Weinan Zhang. 2023. ReLoop2: Building Self-Adaptive Recommendation Models via Responsive Error Compensation Loop. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5728–5738.

[37] Jianing Zhu, Jiangchao Yao, Bo Han, Jingfeng Zhang, Tongliang Liu, Gang Niu, Jingren Zhou, Jianliang Xu, and Hongxia Yang. 2021. Reliable adversarial distillation with unreliable teachers. *arXiv preprint arXiv:2106.04928* (2021).