Mitigating Distribution Shifts in Sequential Recommendation: An Invariance Perspective

Yuxin Liao
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
Hefei, Anhui, China
yuxinliao314@gmail.com

Yonghui Yang
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
Hefei, Anhui, China
yyh.hfut@gmail.com

Min Hou
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
Hefei, Anhui, China
hmhoumin@gmail.com

Le Wu*
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
Hefei, Anhui, China
Institute of Dataspace, Hefei Comprehensive National Science Center
Hefei, Anhui, China
lewu.ustc@gmail.com

Hefei Xu
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
Hefei, Anhui, China
hefeixu604@gmail.com

Hao Liu
Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology
Hefei, Anhui, China
hliu7879@163.com

Abstract

Sequential recommendation aims to learn users' dynamic preferences from their historical interactions and predict the next item they are most likely to engage with. In real-world scenarios, time-varying factors (e.g., product promotions, seasonal changes) induce distribution shifts in user interactions. Despite the demonstrated success of existing models, their generalization capability remains limited under such dynamic conditions. Current methods tackle this challenge by leveraging distributionally robust optimization (DRO) to optimize the "worst-case" loss or by employing manually designed data augmentation to enrich the training distribution. Despite their effectiveness, DRO-based approaches are inherently constrained by the sparsity of training data, limiting the range of distributions they can model, while manually designed augmentations risk introducing noise or irrelevant information that could distort user preference learning. Furthermore, these methods often overlook the sensitivity of user interactions to distribution shifts, which is essential for capturing the stable factors in the evolution of user preferences in real-world settings.

In this work, we tackle the distribution shifting problem from the perspective of invariant learning. We propose a novel framework called ***I**nvariant Learning for **D**istribution Shifts in S**E**quential Recommend**A**tion (**IDEA**)* to develop robust sequential recommendation. The key of *IDEA* lies on learning stable preferences across various distribution-aware environments. Since explicit environments are unavailable, we first extract multiple subsequences by dropping potential noise items, then extend environments with our proposed subsequence mixup. Given the simulated environments, *IDEA* then learns stable user preferences through invariant risk minimization (IRM) across various environments. To encourage the diversity of simulated environments, *IDEA* employs an adversarial training strategy to explore potential diverse environments, and further enhance the model's generalization to unseen test distributions. It is worth mentioning that *IDEA* is a flexible model-agnostic framework, which is applicable to various sequential recommendation models. Extensive experimental results on three public datasets clearly demonstrate the effectiveness of the proposed framework. Our code is available at: https://github.com/hermione314/IDEA.

CCS Concepts

• **Information systems** → **Information retrieval**; • **Human-centered computing** → **Collaborative and social computing**.

Keywords

Sequential Recommendation, Robustness, Invariant Learning

ACM Reference Format:
Yuxin Liao, Yonghui Yang, Min Hou, Le Wu, Hefei Xu, and Hao Liu. 2025. Mitigating Distribution Shifts in Sequential Recommendation: An Invariance Perspective. 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, 11 pages. https://doi.org/10.1145/3726302.3730036

1 Introduction

Sequential recommendation focuses on modeling sequential dependencies in users' historical interactions to uncover their evolving

*Le Wu is the corresponding author.

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.
SIGIR '25, July 13–18, 2025, Padua, Italy
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1592-1/2025/07
https://doi.org/10.1145/3726302.3730036

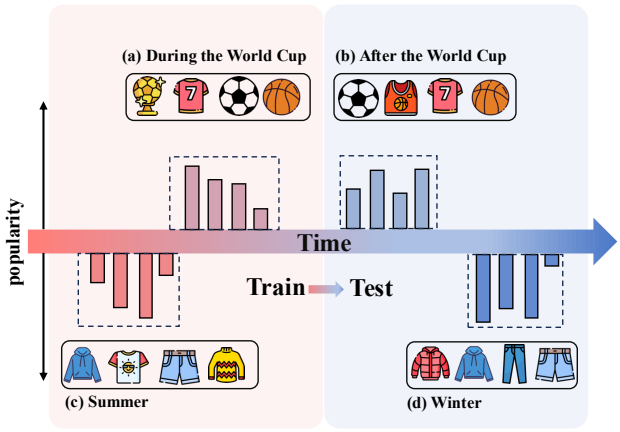


Figure 1: Illustration of real-world distribution shifts. Solid boxes indicate user interaction sequences while dashed boxes indicate item popularity. (a) and (b) indicate event-driven factor(the World Cup). (c) and (d) present time-varying factors(the seasonal changes).

preferences [9, 13, 29, 39, 47]. As a cornerstone of personalized online services, sequential recommendation systems enhance user experiences by predicting future interactions based on past behaviors. With the development of deep learning, increasingly advanced architectures have been successfully applied to sequential recommendation models, such as CNNs [37], RNNs [19], GNNs [6], self-attention mechanisms [23, 36] and selective state-space-models [25], effectively capturing intricate user behavior patterns, achieving advanced performance.

Although the models mentioned above have achieved considerable success, most assume that users' historical and future interactions follow the same distribution, i.e. *the IID assumption* [14]. However, this idealized assumption hardly holds in real-world scenarios, as users' interests evolve dynamically over time and in response to varying contexts. Certain interactions of users are highly sensitive to time-dependent or event-driven factors. As illustrated in Figure 1, during the World Cup (Figure 1 (a)), users typically buy football jerseys to support their favorite teams, but the demand significantly decreases once the tournament concludes (Figure 1 (b)). Meanwhile, in summer (Figure 1 (c)), users tend to purchase T-shirts and shorts to keep cool. However, this fashion trend shifts in winter (Figure 1 (d)). These types of interactions are sensitive to distribution shifts. Models based on the *IID assumption* may be disproportionately influenced by such interactions, overfitting on short-term trends [4, 5], thus struggling to generalize effectively in distribution shifts [7].

To mitigate the performance degradation of sequential recommendation caused by distribution shifts, previous work has made significant efforts by employing techniques such as distributionally robust optimization (DRO) [32, 50, 52], uncertainty modeling [11, 12], and contrastive learning [8, 46]. While these methods have advanced sequential recommendation generalization, they have certain limitations: **(i)** DRO-based methods heavily rely on identifying the worst distribution. RSR [52] treats the worst distribution as the subdistribution with the highest loss, which may overemphasize high-loss, potentially noisy samples. DROS [50] defines robust optimization within a certain shift from the nominal distribution, potentially leading to overly conservative solutions. **(ii)** Uncertainty modeling methods use uncertainty to model possible changes in user interest, but lack causal explanation [43]. **(iii)** Contrastive learning-based methods often rely on arbitrary data augmentation, which may drop important interaction data and retain noise information, misleading user preference modeling [27].

To ensure model generalization under distribution shifts, it is crucial to increase attention on items that users consistently interact with across diverse distributions, reflecting stable preferences. If we can learn users' stable preferences from the reconstructed training distribution, this issue can be readily resolved. Invariant learning is an effective technique to tackle domain generalization challenges by learning invariant representations that can transfer across diverse environments, or intervention distributions [34]. Since explicit environment labels are not available in real-world scenarios, we simulate diverse environments by reconstructing the distribution of users' historical interactions.

In this work, we propose a novel solution from the perspective of invariant learning. Specifically, we propose a framework ***I**nvariant Learning for **D**istribution Shifts in S**E**quential Recommend**A**tion (**IDEA**)* to achieve robust sequential recommendation. *IDEA* consists of two modules, the environment simulation module for simulating more diverse environments that mimic real-world scenarios and the invariance-based optimization module for capturing users' invariant preferences. Specifically, in the environment simulation module, we apply subsequence extractors to systematically remain the interactions that reflect users' stable preferences. To extend the reconstructed training data to unseen distributions, the extracted subsequences are fed into subsequence mixup to generate sequences, which we refer to as environments. In invariance-based optimization stage, *IDEA* employs the invariant optimization objective on simulated heterogeneous environments to learn invariant user preferences. To further enhance the diversity of environments, *IDEA* integrates an adversarial training strategy, enabling the model to explore potential environments and improving its robustness to unseen distributions. Notably, *IDEA* is a model-agnostic framework, making it applicable to a wide range of sequential recommendation models without being tied to specific architectures. Extensive experimental results on three public datasets validate the superior performance and generalization capability of *IDEA*.

We summarize the contributions of this work as follows: (1) We investigate distribution shifts in sequential recommendation from an invariance perspective, and propose a model-agnostic robust sequential recommendation framework *IDEA* to enhance model robustness. (2) *IDEA* first simulates potential environments with the subsequence extractors and uses mixup method to expand environments, then learns stable user preferences via invariance-guided optimization. To further enhance model generalization under potential test distribution, *IDEA* employs an adversarial training strategy to promote environment diversity. (3) We conduct experiments on several benchmarks, theoretical analysis and extensive experiments validate the effectiveness and compatibility of our proposed *IDEA*.

2 Related Works

2.1 Sequential Recommendation

In recent years, sequential recommendation has attracted significant attention due to its focus on modeling the dynamic nature of user behavior. Unlike traditional methods like collaborative or content-based filtering, sequential recommendation captures sequential dependencies in user actions, evolving preferences, and shifting item popularity, enabling more accurate and context-aware recommendations [39].

In the early stages of development, sequential recommendation was largely driven by Markov chain-based models [18, 33]. These approaches primarily focused on modeling the immediate item-to-item transition probabilities between consecutive items but struggled with capturing longer-term patterns. With the advent of deep learning, neural networks provided a powerful tool to overcome these limitations, enabling more expressive modeling of sequential behaviors [19, 23, 25, 36, 37, 45]. GRU4Rec [19] introduced GRUs to sequential recommendation, effectively capturing long-term dependencies and complex patterns. Caser [37] leverages CNN to capture the item transition patterns. To better leverage the history information, attention-based models like SASRec [23] and BERT4Rec [36] improved sequential recommendation by focusing on the most relevant parts of sequences but suffered from computational inefficiencies with long sequences. To address this, recent methods like Mamba4Rec [25] leverages state space models for efficient long-sequence modeling, improving scalability and performance.

2.2 Robust Sequential Recommendation

While these models have succeeded, they assume training and testing data share the same distribution. However, real-world user preferences are diverse and dynamic, causing distribution shifts that often undermine performance.

To enhance robustness and improve generalization in sequential recommendation, researchers have proposed several SOTA techniques [8, 11, 12, 42, 46, 50, 52]. Inspired by contrastive learning, CL4SRec[46] and ICL[8] design auxiliary contrastive tasks to improve the learning of users' sequential features. Distributionally Robust Optimization [42, 50, 52] aims to mitigate performance degradation under distributional shifts by prioritizing hard-to-predict or underrepresented samples, ensuring more stable predictions across different user groups. Additionally, uncertainty modeling [11, 12, 28] addresses the limitations of deterministic embeddings by representing dynamic user interests using stochastic Gaussian distributions. For instance, STOSA [12] employs transformers to jointly model mean and covariance embeddings, effectively capturing the inherent uncertainty in user behaviors and mitigating the effects of ambiguous or noisy interaction data.

Despite these advancements, existing methods still exist limitations when generalizing to unseen test domains. Current strategies for enriching training data often lack effective guidance or are constrained by the training data itself. In particular, many approaches rely on patterns observed in the training data, such as item co-occurrence statistics, which may no longer hold when distributions shifts. To address these limitations, *IDEA* leverages invariant learning to encourage the model to capture stable user preferences, enabling more reliable recommendations in dynamic scenarios.

2.3 Invariant Learning

Invariant learning seeks to learn stable representations that remain robust across heterogeneous environments, aiding out-of-distribution (OOD) generalization. A common example of image recognition is that in the training set, cows are predominantly observed on green grasslands, whereas camels typically appear in yellow deserts. However, the test set suffers distribution-shifted instances when cows in deserts or camels on grasslands. By constructing environments where animals appear under different backgrounds, invariant learning imposes an invariance penalty on ERM loss across environments to encourage the model to learn invariant features of animals. Techniques such as IRM [1] and REx [24] optimize performance via regularization across environments, while EIIL [10] and HRM [26] automatically identify and split environments in the absence of explicit labels. These methods have been applied successfully to image classification, graph learning, and natural language processing. However, applying such automatic environment splitting techniques directly to sequential recommendation remains challenging. Unlike static data scenarios, user behaviors are temporally continuous and context-dependent, while distribution shifts are often driven by complex latent factors (e.g., sudden interest shifts or external events), which existing methods struggle to disentangle without explicit guidance.

In recommendation systems, invariant learning has shown potential in addressing challenges such as OOD generalization [30] and denoising [49]. Works like InvPref [41] apply invariant learning to collaborative filtering to mitigate selection bias and improve debiasing. InvRL [21] and Milk [2] focus on multimedia recommendations, learning invariant multimodal representations for better robustness and accuracy. KGIL [40] enhances recommendations by capturing invariant knowledge graph subgraphs. Our work builds on these advancements by tailoring invariant learning specifically for sequential recommendation. By focusing on capturing user preferences that persist across diverse temporal and contextual environments, we aim to enhance sequential recommendation models' generalization to new scenarios.

3 Preliminaries

3.1 Sequential Recommendation

Notations. Suppose we have a user set $\mathcal{U}$ and an item set $\mathcal{V}$, where $|\mathcal{U}|$ and $|\mathcal{V}|$ denote the number of users and items, respectively. We represent $u \in \mathcal{U}$ and $v \in \mathcal{V}$ as specific instances of a user and an item. The users' historical behaviors are chronologically ordered and organized into an interaction sequence set $\mathcal{S} = \{S^u \mid u \in \mathcal{U}\}$. The interaction sequence of user u is $S^u = \{v_1^u, v_2^u, \cdots, v_T^u\}$, where v_t^u is the item interacted at the timestamp t and T denotes the length of the sequence. We utilize an item embedding matrix $\mathcal{I} \in \mathbb{R}^{|\mathcal{V}| \times d}$ to represent items, where d is the latent dimensionality.

Problem Formulation. Formally, given $\{\mathcal{U}, \mathcal{V}, \mathcal{S}\}$, sequential recommendation aims to learn a model f_θ, which focuses on predicting the most possible item which the user u will interact with at the timestamp $T + 1$ based on his/her current interests.

In general, a sequential recommender generally consists of following three key components:

- *Embedding Layer.* In embedding layer, given a user's interaction sequence S^u, we transform users' sequences in fixed length L and

retrieve the input embedding $\mathbf{E} \in \mathbb{R}^{L \times d}$ from the item embedding matrix $\mathcal{I}$.

- *Sequence Encoder.* Sequence encoder takes input embedding and learns the relationship between items in the historical interaction sequences. Different sequential recommendation models employ distinct sequence encoders, RNN based (e.g., GRU4Rec [19]), self-attention based (e.g., SASRec [23]), and Mamba based (e.g., Mamba4Rec [25]) are widely used.
- *Prediction Layer.* After the sequence encoder has adaptively modeled the item transition patterns, we feed its output into a prediction layer to leverage encoded sequential representation to generate relevance scores for candidate items, typically using inner-product layer.

Typically, the model is trained using Empirical Risk Minimization (ERM), which minimizes global risk by optimizing the expected loss over the observed distribution $\mathbf{P}_{train}$:

$$\theta^* = \arg\min_{\theta} \mathbf{E}_{\mathcal{S} \sim \mathbf{P}_{train}} \mathcal{L}\left(f_\theta\left(v \mid \mathcal{S}\right)\right), \tag{1}$$

where $f_\theta(v|\mathcal{S})$ denotes the prediction score of potential next item. In practice, we use BCE loss [3, 20]:

$$\mathcal{L}_{\text{BCE}} = -\sum_{v \in \mathcal{V}^+} \log \sigma(f_\theta\left(v \mid \mathcal{S}\right)) - \sum_{w \in \mathcal{V}^-} \log(1 - \sigma(f_\theta\left(w \mid \mathcal{S}\right))) \tag{2}$$

where $\mathcal{V}^+$ and $\mathcal{V}^-$ represent the dataset of positive and negative samples respectively, σ denotes the sigmoid function, and we use the ERM loss $\mathcal{L}_e$ to simplify notation.

3.2 Distribution Shifting Issue

Despite significant advancements in sequential recommendation models, these models are typically trained under the ERM paradigm, which assumes that both the training and test data are Independent and Identically Distributed (IID), implying that $\mathbf{P}_{test} = \mathbf{P}_{train}$. However, this idealized assumption is often difficult to satisfy in real-world scenarios, where users' preferences and item popularity are dynamic due to temporal changes or contextual factors, leading to $\mathbf{P}_{test} \neq \mathbf{P}_{train}$, i.e., Out-of-Distribution (OOD) issues. These OOD phenomena can result in a degradation in model performance when the distribution changes. We expect to ensure that the model remains effective in the presence of such distributional shifts.

Our goal is to develop an optimal sequential recommendation model capable of generalizing well to the test distribution $\mathbf{P}_{test}$, where $\mathbf{P}_{test} \neq \mathbf{P}_{train}$. Formally, the optimization objective is to minimize the loss $\mathcal{L}_e(f_\theta\left(v \mid \mathcal{S}\right)$, where the expectation is taken over the unknown test data distribution $\mathbf{P}_{test}$:

$$\arg\min_{\theta} \mathbf{E}_{\mathcal{S} \sim \mathbf{P}_{test}} \mathcal{L}_e\left(f_\theta(v \mid \mathcal{S})\right). \tag{3}$$

3.3 Invariant Learning

Invariant Learning is a mechanism encourages models to capture representations that maintain predictive ability by penalizing performance differences across heterogeneous environments.

Formally, we suppose that training data $\mathcal{D}_{train}$ are sampled from multiple environments $\mathcal{E}$, i.e., $\mathcal{D}_{train} = \{D_m\}_{m \in \mathcal{E}}$, IL technique is encouraging representations that remain constant across environments and label relevance, which can be defined as:

$$\mathbb{E}(Y \mid f(S), E = m) = \mathbb{E}\left(Y \mid f(S), E = m'\right), \forall m, m' \in \mathcal{E}. \tag{4}$$

The constraint is termed as Environment Invariance Constraint (EIC) [10]. The constraint can be incorporated into the ERM optimization objectives via a penalty term [1, 24]:

$$\mathcal{L}_{IRM} = \sum_{m \in \mathcal{E}} \mathcal{L}_e^m(\theta) + \lambda \left\|\nabla_\theta \mathcal{L}_e^m(\theta)\right\|_2^2 \text{ (IRM-v1)} \tag{5}$$

$$\mathcal{L}_{REx} = \sum_{m \in \mathcal{E}} \mathcal{L}_e^m(\theta) + \lambda \operatorname{Var}\left(\left\{\mathcal{L}_e^1(\theta), \ldots, \mathcal{L}_e^m(\theta)\right\}\right) \text{ (V-REx)}, \tag{6}$$

where θ denotes the model parameters, $\mathcal{L}_e^m(\theta)$ denotes the ERM loss in the environment m, realized by Equation (2). The performance of invariant learning is heavily dependent on the quality of the environments. In ideal scenarios, the training data can be explicitly segmented into multiple environments. However, the lack of explicit environmental labels in sequence recommendation settings pose a challenge to learning invariant sequential patterns.

4 The Proposed *IDEA* Framework

4.1 Overview

IDEA aims to learn stable user representations across different environments, thereby improving generalization. The key problem *IDEA* addresses is how to generate multiple high-quality environments and capture the users' invariance features from variant environments. We implement *IDEA* with a max-min game. As shown in Figure 2, it comprises two components: ***Environment Simulation***, which focuses on constructing diverse environments, and ***Invariant Optimization***, which enforces learning stable preferences across various environments.

For environment simulation, we introduce K learnable subsequence extractors that identify and drop unstable interactions under distribution shifts by evaluating the semantic relevance of interactions. The optimization of the subsequence extractors is guided by the maximize violations of invariance objectives, ensuring that it gradually filters out interactions that are highly sensitive to distribution shifts. Besides, the Mixup operation further extends the environments to unseen domains, simulating potential test distributions and thus enhancing the model generalization.

For invariant optimization, *IDEA* involves optimizing two sets of parameters: parameters of K subsequence extractors $\{\theta_1^*, \ldots, \theta_K^*\}$ and those of the recommendation module θ_{rec}. For subsequence extractors, we aim to maximize the diversity of the generated K subsequences, ensuring they capture distinct environmental variations. For the recommendation module, we apply the environment invariance constraint to minimize differences between generated environments, thereby enforcing consistent model performance across varying conditions. Finally, *IDEA* alternately optimizes these two modules through adversarial training—enhancing both the diversity of generated environments and the learning of users' invariant preferences in an iterative manner.

4.2 Environment Simulation

4.2.1 Sequential Recommendation Model Pretraining. In order to extract meaningful sub-sequences, we first pre-train a base sequential recommendation model $f_{\tilde{\theta}_{rec}}$ to obtain item embeddings. Specifically, we implement sequential recommendation based

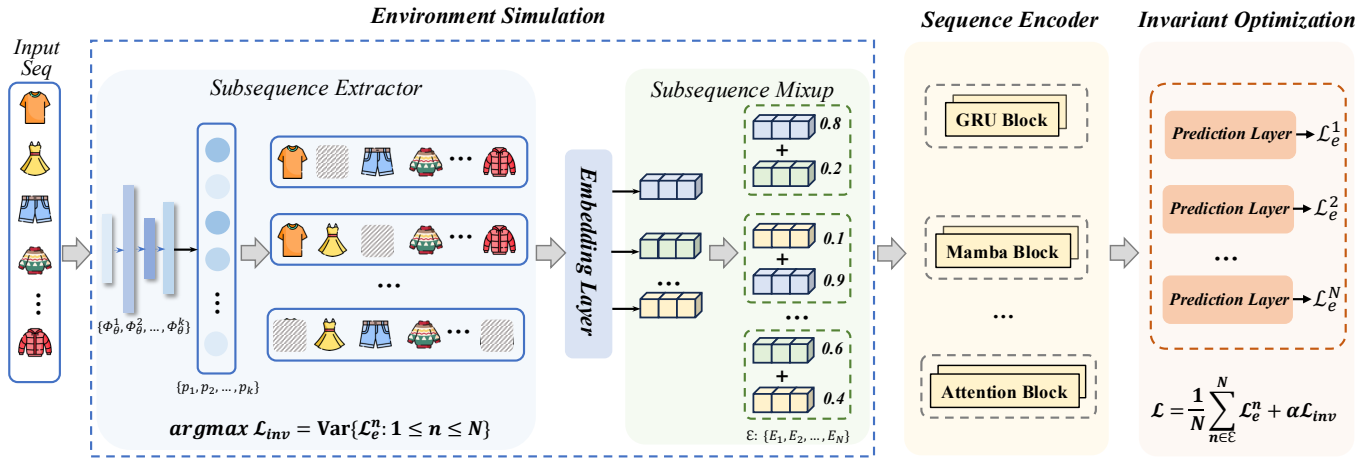


Figure 2: The overall framework of *IDEA*: subsequence extractions first generate learnable mask, followed by subsequence mixup generating diverse environments. Environments are then fed into the recommender which are optimized using environment invariance constraint (EIC) later.

on three representative backbones (GRU4Rec [19], SASRec [23], Mamba4Rec [25]). Let $\mathcal{I}$ denote the pre-trained item embedding matrix, we encode each input sequence as follows:

$$\mathbf{E}^u = \sigma\left(i_1^u \left\| i_2^u \right\| \ldots \| i_T^u\right), \tag{7}$$

where $\|$ denotes concat operation, i_t^u represent the embedding of item embedding user u interacted at timestamp t. σ denotes mean-pooling operation. We then feed the sequence embedding as input to activate the subsequence extractor. Next, we introduce how the extractor works.

4.2.2 Subsequence Extraction. Here, we design K independent subsequence extractors. Specifically, each user sequential representation $\mathbf{E}^u$ will be associated with a learnable vector $\mathbf{p}_k^u$, the representation of k^{th} extracted subsequence S_k^u is:

$$\mathbf{E}_k^u = \mathbf{E}^u \odot \mathbf{p}_k^u, \tag{8}$$

which indicates that item v_t^u in S^u remains if the corresponding element of p_k^u is 1 and is masked otherwise. The calculation of $\mathbf{p}_k^u$ takes users' collaborative singles ω_k into consideration as follows:

$$\mathbf{p}_k^u = \text{sigmoid}\left(\left(\log\delta - \log(1-\delta) + \omega_k\right)/\tau\right), \delta \sim \text{Uniform}(0,1), \tag{9}$$

where $\mathbf{p}_k^u$ is derived from a Bernoulli distribution with the weight ω_k, and we utilize the Gumbel-Softmax [22] reparameterization trick, allowing gradients to be backpropagated from the probability space to the parameters of the MLP network, enabling end-to-end optimization. δ is a random variable sampled from a uniform distribution and τ is the temperature hyperparameter, as $\tau \to 0$, $\mathbf{p}_k^u$ get close to binary. ω_k is parameterized via an MLP network Φ_θ^k:

$$\omega_k^u = \Phi_\theta^k\left(\sigma\left(\mathbf{E}^u\right)\right). \tag{10}$$

Hence, we can define K independent environment generators to generate K groups of subsequences named $\mathcal{D}_1$.

4.2.3 Subsequence Mixup. In practice, the variability of target domain shifts is inherently unpredictable, and the environments derived from the limited training domains are inadequate to encompass the full range of potential domains. Consequently, the model's generalization capability cannot be ensured. To enable robust performance in unpredictable test environments, we aim to extend the training distribution to unseen scenarios. Our key insight is that users' interests are diverse. Therefore, the environmental construction should adhere to two principles: (i) *Interest-driven heterogeneity*: different environments should be dominated by distinct subsets of user interests, preventing the model from over-relying on specific interest patterns. (ii) *Sufficient environmental diversity*: a broad spectrum of environments must be simulated to enhance the model's capacity for generalization. Here, we adopt the mixup technique [38, 51] to construct more diverse environments.

Mixup is a data augmentation technique that creates new samples by performing linear interpolation between original samples. Specifically, we randomly select two subsequences S_i and S_j from $\mathcal{D}_1$ and obtain their sequential representation $\mathbf{E}_i$ and $\mathbf{E}_j$ using Equation (7). We generate the mixed representation of $\mathbf{E}_i$ and $\mathbf{E}_j$:

$$\begin{aligned}\mathbf{E}'_{ij} &= \lambda\mathbf{E}_i + (1-\lambda)\mathbf{E}_j,\\ \lambda &\sim \text{Beta}(\alpha,\alpha),\end{aligned} \tag{11}$$

where $\mathbf{E}'_{ij}$ represents the mixed representations of S_i and S_j. λ is sampled from the Beta distribution [16] with parameter α, which is close to 1. By repeating the above mixup process N times, we obtain N distinct environments. Next, we present the optimization process of the *IDEA* framework.

4.3 Invariance-based Optimization

4.3.1 Environment Exploration. The purpose of environment simulation is to construct more diverse environments to serve the following invariant learning better. To optimize subsequence extractors for capturing distribution-stable interactions in training data, we design an environment exploration objective to optimize

subsequence extractors. Ideal environments should maximize violations of the invariance principle, thereby maximizing the challenge of learning stable features. In order to fully explore the item transition patterns that may be missed by the current model, we update the parameters of the subsequence extractors by defining the following optimization objective:

$$\theta_1^*, \ldots, \theta_K^* = \arg\max_{\theta_1,\ldots,\theta_K} \mathrm{Var}\left(\left\{\mathcal{L}_e^n : 1 \le n \le N\right\}\right), \tag{12}$$

where $\theta_1, \ldots, \theta_K$ are the parameters of K distinct subsequence extractors, $\mathcal{L}_e^n$ is the empirical risk in environment n, calculated by Equation (2) in practice. It encourages the subsequence extractors to explore more challenging environments by maximizing the variance of the empirical risk across N environments. In our implementation, this optimization alternates with the following overall optimization. Specifically, we define T iterations as a cycle.

4.3.2 Overall Optimization. After obtaining N challenging environments, we retrain the recommendation module in each environment with the ERM optimization objective, combined with the invariance-based regularizer to encourage the model to capture stable preferences.

The overall optimization objective is defined as follows:

$$\theta_{rec}^* = \arg\min_{\theta_{rec}} \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}_e^n + \beta \mathrm{Var}\left(\left\{\mathcal{L}_e^n : 1 \le n \le N\right\}\right), \tag{13}$$

where θ_{rec} is the parameter of recommendation module. The second term encourages the invariant risk minimization, as formulated in Equation (6). β is a hyperparameter.

The introduction of the invariance regularizer encourages the model to learn stable item transition relationships that are independent of distributional shifts, leading to stable item representations. In this way, the model gains stronger adaptability to varying environments. During the inference phase, by leveraging the learned stable representations, the model is able to maintain high prediction accuracy even in the presence of distributional drift, thereby enhancing its robustness and generalization.

4.4 Theoretical Analysis

In this section, we analyze from the perspective of domain generalization to prove the validity of *IDEA*. Our analysis demonstrates that implementing *IDEA* effectively reduces the upper bound of the error optimization function for generalization, thereby proving its effectiveness. Given $\mathbb{P}$ as the source domain distributions and $\mathbb{Q}$ as the target domain distribution, previous studies [35, 48] have proved the following theory:

Theorem 4.1. *Let $\mathcal{X}$ be a space and let $\mathcal{H}$ be a class of hypotheses corresponding to this space. Let $\mathbb{Q}$ and $\{\mathbb{P}_i\}_{i=1}^k$ be distributions over $\mathcal{X}$ and let $\{\varphi_i\}_{i=1}^k$ be a collection of non-negative coefficients with $\sum_i \varphi_i = 1$. Let the object O be a set of distributions such that for every $\mathbb{S} \in O$ the following holds.*

$$\sum_i \varphi_i d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{P}_i, \mathbb{S}\right) \le \max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{P}_i, \mathbb{P}_j\right), \tag{14}$$

where $d_{\mathcal{H}\Delta\mathcal{H}}$ measures distributional differences between $\mathbb{P}_i$ and $\mathbb{P}_j$.

Then, for any $h \in \mathcal{H}$, the following equation holds:

$$\varepsilon_{\mathbb{Q}}(h) \le \lambda_\varphi + \sum_i \varphi_i \varepsilon_{\mathbb{P}_i}(h) + \frac{1}{2}\min_{\mathbb{S}\in O} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q}) + \frac{1}{2}\max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{P}_i, \mathbb{P}_j\right), \tag{15}$$

where $\lambda_\varphi = \sum_i \varphi_i \lambda_i$ and each λ_i is the error of an IDEA *l joint hypothesis for $\mathbb{P}$ and $\mathbb{P}_i$.*

From the theory, the upper bound of the model's error in the unseen target domain $\mathbb{Q}$ can be expressed as Equation (15). A lower value of $\varepsilon_{\mathbb{Q}}(h)$ indicates better generalization performance of the model. Then, we analyze each term of Equation (15).

The *first term* is too small compared to the generation error to be omitted in practice. The *second term* is the convex combination of the source errors controlled by the ERM. For the *third term*, $\frac{1}{2}\min_{\mathbb{S}\in O} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}, \mathbb{Q})$ is the minimum pairwise H-divergence between $\mathbb{S}$ and $\mathbb{Q}$, we continuously increase the diversity of environment O to ensure that it contains sub-distribution $\mathbb{S}$ that is increasingly closer to the unknown test distribution. In *IDEA* , during the ***Environment Simulation*** process, we use K subsequence extractors to remove the potential noise interactions and reconstruct the training distribution. Besides, we apply subsequence mixup to mix the extracted subsequences pairwise to form N diverse environments, successfully expanding the training distribution to the unseen test distribution. Additionally, we further encourage distribution differences between different environments through Equation (12). The *final term* is the maximum differences among training environments, this term is optimized via Equation (13) during the ***Invariant Optimization*** phase, which reduces $\max_{i,j} d_{\mathcal{H}\Delta\mathcal{H}}\left(\mathbb{P}_i, \mathbb{P}_j\right)$ by penalizing the differences in empirical risks across environments. Through adversarial training, *IDEA* iteratively optimizes each term in Equation (15), thereby reducing the upper bound of the error in the target domain. As a result, *IDEA* achieves better robustness.

4.5 Model Discussion

4.5.1 Space Complexity. As illustrated in Algorithm 1, the parameters of *IDEA* are composed of two parts: sequential recommender parameters θ_{rec} and environment simulation parameters $\{\theta_1, \ldots, \theta_K\}$. Compared to backbone models, the extra parameters are those of the subsequence extractors, implemented as MLPs. Because θ_k are the shared for all sequences, the additional parameters of *IDEA* is affordable.

4.5.2 Time Complexity. The overall time complexity stems from two key components: *(i) Environment Simulation.* The complexity of $O(K|\mathcal{S}|d + N)$ arises from two stages: *Subsequence Extraction: K* subsequence extractors independently process each item's interaction sequence of length $|\mathcal{S}|$. For each position in the sequence, a probability is computed to identify critical subsequences, incurring $O(K|\mathcal{S}|d)$ complexity, where d denotes the dimensionality of sequence representations. *Subsequence Mixup: N* synthetic environments are created via mixup operations on the extracted subsequences, contributing $O(N)$ complexity. Parameters are updated every T iterations, effectively mitigating computational costs. Experiments show optimal performance with $K = 3$ and $N = 4$, confirming minimal overhead from environment generation. *(ii) Invariant Optimization.* The complexity of $O(N|\mathcal{S}|d)$ originates from optimizing invariant representations across all environments.

Algorithm 1: The Training Stage of the Proposed *IDEA*

Data: User set U, Item set V, Interaction sequence set $\mathcal{S}$
Result: Optimal robust sequential recommender $f^*_{\theta_{rec}}(\cdot)$

Initialize all parameters;
while not converged **do**
  Embed items $\{s_u\}$ to get the sequence initial representations via Eq.(7);
  *// **Environment Simulation***
  Calculate $\mathbf{p}_u^k$ and extract the subsequence via Eq.(8-10);
  Mixup the extracted subsequence to generate diverse environments via Eq.(11);
  *// **Invariance-based Optimization***
  for each environment n **do**
    Model sequential representation with backbone model;
    Compute ERM loss of each environment $\mathcal{L}_e^n$ via Eq.(2);
  end for
  Optimizing θ_{rec} with the overall loss $\mathcal{L}$ via Eq.(13);
  Alternatively optimize $\{\theta_1, \ldots, \theta_K\}$ via Eq.(12) every T batch;
end while
return the optimal robust sequential recommender $f^*_{\theta_{rec}}(\cdot)$

Specifically, for each of the N environments, the method enforces invariance over all $|\mathcal{S}|$ interaction sequences with d-dimensional representations. Considering the performance-complexity trade-off, the incurred costs are justifiable, as validated by empirical efficiency and robustness gains.

Table 1: Statistics of the experiment datasets.

Dataset	Users	Items	Interactions	Avg. Length
Beauty	22,364	12,102	198,502	8.87
Sports	35,599	18,358	296,337	8.32
MovieLens-1M	6,041	3,417	999,611	165.47

5 Experiments

In this section, we conduct experiments to demonstrate the effectiveness and generalization of our proposed *IDEA* and to reveal the reasons. We focus on the following important research questions:

- **RQ1:** How does *IDEA* compare with SOTA robust sequential recommendation frameworks
- **RQ2:** How effectively does *IDEA* mitigate performance degradation in distribution shifts?
- **RQ3:** How do some important designs and hyperparameters affect the model?
- **RQ4:** How do different components in our framework contribute to the final performance?

5.1 Experimental Settings

5.1.1 Datasets. To ensure a comprehensive evaluation, we utilize three publicly available recommendation datasets:

- **Beauty and Sports** [31]: These are subdatasets of the Amazon datasets, which include product reviews and purchase records categorized by top-level products. "Beauty" covers beauty product purchases, while "Sports" focuses on sports and outdoor items.
- **MovieLens-1M** [17]: It is a widely used benchmark dataset for recommendation systems, containing 1 million movie ratings along with user demographics and movie genres.

For all datasets, we utilize 5-core filtering [44], and sort interactions chronologically. The last item of each sequence is used for testing, the second-to-last item for validation, and all remaining items for training. The detailed statistics for each dataset after preprocessing are summarized in Table 1.

5.1.2 Backbones. We apply it to three representative vanilla sequential recommendation models:

- **GRU4Rec** [19]: It is a representative RNN-based sequential recommendation model which uses GRUs to control the RNN units to summarize all previous interactions through the hidden state.
- **SASRec** [23]: It is a transformer-based sequential recommendation model which uses the self-attention to model the relationship between any two items in a long sequence.
- **Mamba4Rec** [25]: It improves inference efficiency in transformer-based models by adopting selective SSMs [15].

5.1.3 Baselines. We compare our approach with the following sequential recommendation baselines, since these methods are all model agnostic, we implemented them on above backbone models.

- **CL4SRec** [46]: It leverages a contrastive learning framework to generate self-supervised signals. It employs three data augmentation methods to enrich training distribution.
- **STOSA** [12]: It uses stochastic Gaussian distribution to model the uncertain embedding of the item embeddings, which takes into account the randomness of item conversion.
- **RSR** [52]: It utilizes the idea of distributionally robust optimization (DRO) to optimize the worst-case distribution of training data. By reweighting the samples, it simulates different training distributions and thus overcomes potential distribution shifts.
- **DROS** [50]: It generalizes training data to a nominal distribution in a dynamic environment and seeks robust recommendations within a robust radius of the nominal distribution.

5.1.4 Implementation details. We implement all methods with Python 3.8 and PyTorch 2.4.1. We use Adam optimizer, the learning rate is tuned as 0.001 and the batch size is set as 1024. We empirically set the temperature coefficient $\tau = 0.1$, and the embedding size $d = 64$ for all datasets. To maintain a consistent comparison, we adopt the full-ranking strategy across all experiments. For each user, items included in the dataset are deemed positive, while others are considered negative. Ultimately, we present the average outcomes for all users within the testing set. We evaluate the model using widely used metrics, such as *Recall* and *NDCG*, by comparing the top-10 and top-20 recommended items with the ground truth For hyperparameters, the environment number N as well as the weight of the invariant constraint term β, we carefully search the best parameters for each dataset and report detailed comparisons. For all baselines, we refer to original parameters for fair comparisons.

5.2 Overall Performance (RQ1)

To verify the effectiveness of our proposed *IDEA*, we compare *IDEA* with competing methods. The overall comparison results are presented in Table 2, yielding the subsequent observations:

Table 2: Overall comparison between *IDEA* and baselines. "Vanilla" represent the base sequential recommendation model. We have implemented both the baseline and our method on different base models. "R" and "N" are the abbreviations for Recall and NDCG respectively.

Backbones	Methods	Beauty				Sports				ML-1M			
		R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
GRU4Rec	Vanilla	0.0591	0.0851	0.0326	0.0394	0.0303	0.0469	0.0153	0.0195	0.2366	0.3422	0.1241	0.1509
	CL4SRec	0.0598	0.0878	0.0331	0.0399	0.0322	0.0515	0.0166	0.0214	0.2430	0.3432	0.1282	0.1534
	STOSA	0.0605	0.0899	0.0339	0.0407	0.0345	0.0528	0.0182	0.0228	0.2414	0.3425	0.1243	0.1517
	RSR	0.0593	0.0881	0.0337	0.0402	0.0353	0.0537	0.0183	0.0229	0.2490	0.3512	0.1346	0.1602
	DROS	0.0617	0.0887	0.0343	0.0411	0.0406	0.0583	0.0212	0.0268	0.2454	0.3510	0.1300	0.1565
	IDEA	**0.0660**	**0.0983**	**0.0365**	**0.0431**	**0.0475**	**0.0644**	**0.0305**	**0.0281**	**0.3051**	**0.3602**	**0.1432**	**0.1613**
SASRec	Vanilla	0.0696	0.1048	0.0340	0.0431	0.0408	0.0603	0.0216	0.0270	0.2720	0.3707	0.1562	0.1811
	CL4SRec	0.0806	0.1154	0.0399	0.0487	0.0479	0.0700	0.0226	0.0282	0.2881	0.3915	0.1626	0.1897
	STOSA	0.0834	0.1137	0.0373	0.0455	0.0462	0.0678	0.0223	0.0281	0.2876	0.3843	0.1664	0.1907
	RSR	0.0838	0.1144	0.0415	0.0499	0.0477	0.0684	0.0222	0.0278	0.2882	0.3919	0.1711	0.1950
	DROS	0.0844	0.1151	0.0416	0.0503	0.0480	0.0704	0.0227	0.0283	0.2959	0.3940	0.1698	0.1945
	IDEA	**0.0851**	**0.1190**	**0.0464**	**0.0545**	**0.0490**	**0.0735**	**0.0232**	**0.0290**	**0.3043**	**0.4001**	**0.1732**	**0.2038**
Mamba4Rec	Vanilla	0.0721	0.0978	0.0433	0.0506	0.0405	0.0605	0.0217	0.0256	0.3053	0.3856	0.1779	0.1951
	CL4SRec	0.0727	0.1033	0.0435	0.0513	0.0410	0.0617	0.0230	0.0288	0.3062	0.3886	0.1781	0.1986
	STOSA	0.0758	0.1066	0.0442	0.0514	0.0426	0.0621	0.0236	0.0276	0.3073	0.3885	0.1800	0.1959
	RSR	0.0739	0.1058	0.0436	0.0517	0.0415	0.0612	0.0237	0.0287	0.3072	0.3889	0.1813	0.1965
	DROS	0.0759	0.1071	0.0443	0.0521	0.0417	0.0613	0.0231	0.0280	0.3074	0.3891	0.1851	0.2002
	IDEA	**0.0801**	**0.1093**	**0.0462**	**0.0548**	**0.0427**	**0.0629**	**0.0245**	**0.0295**	**0.3082**	**0.3906**	**0.1879**	**0.2101**

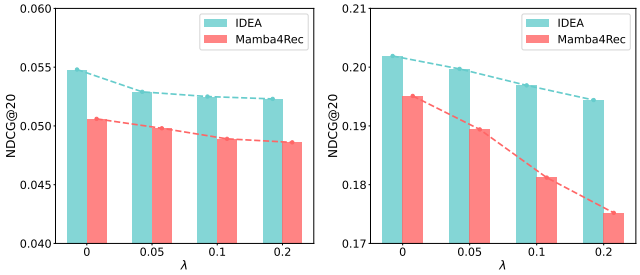
- Among the different base sequence recommendation models, we find that Mamba4Rec is basically superior to GRU4Rec and SASRec on both sparse and dense datasets. The reason might be that Mamba4Rec employs the selective state space model (SSMs), which introduces an input-dependent selection mechanism that effectively extracts key information and filters out noise based on the input data. Additionally, the incorporation of techniques such as residual connections and layer normalization further enhances the model's sequence modeling capability, especially on long sequences.
- All baseline methods outperform naive sequential recommendation models, emonstrating their effectiveness. Among them, DRO-based frameworks have shown greater improvements compared to other baselines, with DROS being the best-performing method. This performance improvement can be attributed to the considering the dynamics of data in the training process, rather than statically relying on previous interactions in the training data. CL-based methods have only achieved marginal improvements in some cases, suggesting their limitations of manually designed data augmentation methods without guidance. The STOSA method for modeling uncertainty provides overly conservative estimates of the extent to which the distribution of future test data will be shifted, and thereby has performance limitations.
- Encouragingly, *IDEA* consistently outperforms the baselines across different datasets, evaluation metrics and base models. Specifically, on the Amazon Beauty, *IDEA* improves the NDCG@20 of the base models GRU4Rec, SASRec, and Mamba4Rec by 9.39%, 26.45%, and 8.30%, respectively. Compared to the strongest baseline, DROS, the improvements are 4.87%, 8.35%, and 5.18%. These results demonstrate the generalization and effectiveness of *IDEA*. A closer examination reveals that while most current self-supervised learning techniques rely on random data augmentation to generate views, *IDEA* implements learnable environment simulators by adversarially maximizing the variance in invariant learning objectives, enabling the exploration of diverse and informative distributions. The invariance penalty endows our model with an enhanced ability to capture stable user preferences.

5.3 Robustness to Distribution Shifts (RQ2)

Motivation & Settings To further assess the robustness of our model to distributional changes, we simulate distributional shifts by constructing a semi-synthetic dataset following [52]. Specifically, we keep the testing set, and simulate distribution shifts by replacing real items in the training sequences with some fake items (which will not appear in the test set) with various proportions $\lambda \in \{0, 0.05, 0.1, 0.2\}$, artificially constructing unstable interactions under distribution shifts. $\lambda = 0$ means using the original training data. We focus on the improvement of our framework on base sequential recommenders (we choose the best performing model: Mamba4Rec), and omit the other baselines in the final results for simplicity.

Results As illustrated in Figure 3, we exhibit the recommendation performances under different fake injection ratio, where the horizontal axis represents the fake item injection ratio, the vertical axes denote NDCG@20 and relative improvements compared with Mamba4Rec. From the results presented in Figure 3, we can see:

- Increasing the value of λ means a larger distribution shift between the training and test sets. We can find that as the proportion of injected fake items increases, all models exhibit a decline in performance. It also demonstrates the challenge that distribution

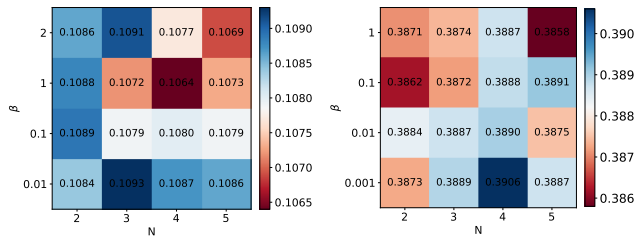


(a) Beauty (b) ML-1M

Figure 3: Performance comparison under different distribution shifts. For simplicity, we remove "@10" in the metrics.

shifting poses to model generalization. Fake items in the user sequences during training may mislead the model into learning incorrect sequence patterns, which then results in poor performance on the test set, where the distribution has shifted.

- For each noise ratio, our framework consistently outperforms competing models. Comparing the performance improvement of our framework for different values of λ, we find that the improvement is more significant on datasets with larger distributional skews. Notably, our NDCG@20 on the ML-1M dataset improves by 5.44% (when $\lambda = 0.05$) and 8.75% (when $\lambda = 0.1$) and 10.96% (when $\lambda = 0.2$), respectively, when compared to the backbone (MambaRec). These improvements demonstrate the effectiveness of our framework under different noise ratios.



(a) Beauty (b) ML-1M

Figure 4: Impact of the number of environments N and invariance coefficient β.

5.4 Hyper-Parameter Sensitivities (RQ3)

We next analyze the influence of two core parameters in *IDEA*: the environment number N and the invariance constraint weight β which controls the trade-off between out-of-distribution exploration and prediction accuracy. As shown in Figure4, we conduct careful grid search of (N, β) on three datasets. We observe that *IDEA* reaches the best performance when $N = 3, \beta = 0.01$ on Amazon Beauty, and $N = 4, \beta = 0.001$ on ML-1M respectively. The hyperparameter β has a direct impact on the model. If β is too high, the model generates distributions beyond the training range, introducing harmful noise. If too low, it prioritizes prediction over sufficient out-of-distribution exploration. Therefore, a tailored β is essential for balancing these factors and must be carefully tuned for optimal performance.

Table 3: Ablation study of *IDEA*.

Models	Beauty		Sports		ML-1M	
	R@10	N@10	R@10	N@10	R@10	N@10
IDEA w/o BOTH	0.0721	0.0978	0.0405	0.0605	0.3053	0.3856
IDEA w/o EE	0.0790	0.1091	0.0419	0.0621	0.2854	0.3861
IDEA w/o IL	0.0774	0.1076	0.0426	0.0627	0.2833	0.3859
IDEA	0.0801	0.1104	0.0427	0.0629	0.3082	0.3906

5.5 Ablation Study (RQ4)

To exploit the effectiveness of each component of our proposed *IDEA*, we use Mamba4Rec as backbone model and conduct ablation study on three datasets. As shown in Table 3, ***IDEA*-w/o EE** denotes the variant without environment exploration, performing invariant learning only on the original generated environments. ***IDEA*-w/o IL** refers to maintaining the environment simulation phase, but training under the ERM paradigm. ***IDEA*-w/o BOTH** means that we remove both modules at the same time, thereby degrading to backbone model. We can observe that each component of *IDEA* contributes to the final superior performance. First, removing environment exploration significantly degrades performance, underscoring its essential role in promoting invariant learning. Since the environment remains unknown to the model, the absence of this component prevents the model from effectively capturing cross-environment invariant item transition patterns, which could potentially lead to suboptimal performance. Second, performance gap between *IDEA* and w/o IL highlights the effectiveness of invariant learning in guiding the stable representation learning process and boosting the model's generalization.

6 Conclusion

In this paper, we introduce *IDEA*, a novel framework that aims to enhance generalization of sequential recommendation under the distribution shifts scenario from an invariant learning perspective. By modeling diverse environments and incorporating invariant learning, *IDEA* enables models to capture robust item representations that remain valid across different data distributions. The framework combines learnable environment generation strategy with an optimization process that focuses on discovering invariant prediction mechanisms to ensure adaptability and robustness. Theoretical analyses support the soundness of our approach, while extensive experiments demonstrate that it outperforms existing methods, achieving consistent improvements in both regular and out-of-distribution settings. Our work highlights the potential of using invariant learning to address real-world challenges in sequential recommendation, providing a scalable and effective solution to improve robustness and generalization in dynamic environments.

Acknowledgements

This work was supported in part by grants from the National Key Research and Development Program of China (Grant No. 2021ZD0111802), the National Natural Science Foundation of China (Grant No. U23B2031, 721881011, 62402159), and the Fundamental Research Funds for the Central Universities (Grant No. JZ2023HGQA0471, JZ2024HGTA0187).

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893* (2019).

[2] Haoyue Bai, Le Wu, Min Hou, Miaomiao Cai, Zhuangzhuang He, Yuyang Zhou, Richang Hong, and Meng Wang. 2024. Multimodality invariant learning for multimedia-based new item recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 677–686.

[3] Andreas Buja, Werner Stuetzle, and Yi Shen. 2005. Loss functions for binary class probability estimation and classification: Structure and applications. *Working draft, November* 3 (2005), 13.

[4] Miaomiao Cai, Lei Chen, Yifan Wang, Haoyue Bai, Peijie Sun, Le Wu, Min Zhang, and Meng Wang. 2024. Popularity-Aware Alignment and Contrast for Mitigating Popularity Bias. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 187–198.

[5] Miaomiao Cai, Min Hou, Lei Chen, Le Wu, Haoyue Bai, Yong Li, and Meng Wang. 2024. Mitigating Recommendation Biases via Group-Alignment and Global-Uniformity in Representation Learning. *ACM Transactions on Intelligent Systems and Technology* (2024).

[6] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 378–387.

[7] Lei Chen, Le Wu, Kun Zhang, Richang Hong, Defu Lian, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Improving recommendation fairness via data augmentation. In *Proceedings of the ACM Web Conference 2023*. 1012–1020.

[8] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.

[9] Mingyue Cheng, Qi Liu, Wenyu Zhang, Zhiding Liu, Hongke Zhao, and Enhong Chen. 2024. A general tail item representation enhancement framework for sequential recommendation. *Frontiers of Computer Science* 18, 6 (2024), 186333.

[10] Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. 2021. Environment inference for invariant learning. In *ICML*. PMLR, 2189–2200.

[11] Ziwei Fan, Zhiwei Liu, Shen Wang, Lei Zheng, and Philip S Yu. 2021. Modeling sequences as distributions with uncertainty for sequential recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 3019–3023.

[12] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. 2022. Sequential recommendation via stochastic self-attention. In *Proceedings of the ACM web conference 2022*. 2036–2047.

[13] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–42.

[14] Chongming Gao, Shiqi Wang, Shijun Li, Jiawei Chen, Xiangnan He, Wenqiang Lei, Biao Li, Yuan Zhang, and Peng Jiang. 2023. CIRS: Bursting filter bubbles by counterfactual interactive recommender system. *ACM Transactions on Information Systems* 42, 1 (2023), 1–27.

[15] Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752* (2023).

[16] Arjun K Gupta and Saralees Nadarajah. 2004. *Handbook of beta distribution and its applications*. CRC press.

[17] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[18] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.

[19] B Hidasi. 2015. Session-based Recommendations with Recurrent Neural Networks. *arXiv preprint arXiv:1511.06939* (2015).

[20] Yaoshiang Ho and Samuel Wookey. 2019. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE access* 8 (2019), 4806–4813.

[21] Shanshan Huang, Haoxuan Li, Qingsong Li, Chunyuan Zheng, and Li Liu. 2023. Pareto invariant representation learning for multimedia recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6410–6419.

[22] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[23] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[24] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*. PMLR, 5815–5826.

[25] Chengkai Liu, Jianghao Lin, Jianling Wang, Hanzhou Liu, and James Caverlee. 2024. Mamba4rec: Towards efficient sequential recommendation with selective state space models. *arXiv preprint arXiv:2403.03900* (2024).

[26] Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. 2021. Heterogeneous risk minimization. In *International Conference on Machine Learning*. PMLR, 6804–6814.

[27] Jinwei Luo, Mingkai He, Weike Pan, and Zhong Ming. 2023. BGNN: Behavior-aware graph neural network for heterogeneous session-based recommendation. *Frontiers of Computer Science* 17, 5 (2023), 175336.

[28] Haokai Ma, Ruobing Xie, Lei Meng, Xin Chen, Xu Zhang, Leyu Lin, and Zhanhui Kang. 2024. Plug-In Diffusion Model for Sequential Recommendation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, February 20-27, 2024, Vancouver, Canada*. 8886–8894. https://doi.org/10.1609/AAAI.V38I8.28736

[29] Haokai Ma, Ruobing Xie, Lei Meng, Yimeng Yang, Xingwu Sun, and Zhanhui Kang. 2024. SeeDRec: Sememe-based Diffusion for Sequential Recommendation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*. 2270–2278.

[30] Wenyu Mao, Jiancan Wu, Haoyang Liu, Yongduo Sui, and Xiang Wang. 2024. Invariant graph learning meets information bottleneck for out-of-distribution generalization. *arXiv preprint arXiv:2408.01697* (2024).

[31] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[32] Hamed Rahimian and Sanjay Mehrotra. 2019. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659* (2019).

[33] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[34] Pengyang Shao, Le Wu, Kun Zhang, Defu Lian, Richang Hong, Yong Li, and Meng Wang. 2024. Average user-side counterfactual fairness for collaborative filtering. *ACM Transactions on Information Systems* 42, 5 (2024), 1–26.

[35] Anthony Sicilia, Xingchen Zhao, and Seong Jae Hwang. 2023. Domain adversarial neural networks for domain generalization: When it works and how to improve. *Machine Learning* 112, 7 (2023), 2685–2721.

[36] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[37] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[38] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*. PMLR, 6438–6447.

[39] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830* (2019).

[40] Shuyao Wang, Yongduo Sui, Chao Wang, and Hui Xiong. 2024. Unleashing the Power of Knowledge Graph for Recommendation via Invariant Learning. In *Proceedings of the ACM on Web Conference 2024*. 3745–3755.

[41] Zimu Wang, Yue He, Jiashuo Liu, Wenchao Zou, Philip S Yu, and Peng Cui. 2022. Invariant preference learning for general debiasing in recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1969–1978.

[42] Hongyi Wen, Xinyang Yi, Tiansheng Yao, Jiaxi Tang, Lichan Hong, and Ed H Chi. 2022. Distributionally-robust recommendations for improving worst-case user experience. In *Proceedings of the ACM Web Conference 2022*. 3606–3610.

[43] Chenwang Wu, Xiting Wang, Defu Lian, Xing Xie, and Enhong Chen. 2023. A causality inspired framework for model interpretation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2731–2741.

[44] Le Wu, Xiangzhi Chen, Fei Liu, Junsong Xie, Chenao Xia, Zhengtao Tan, Mi Tian, Jinglong Li, Kun Zhang, Defu Lian, et al. 2025. EduStudio: towards a unified library for student cognitive modeling. *Frontiers of Computer Science* 19, 8 (2025), 198342.

[45] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 346–353.

[46] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.

[47] En Xu, Zhiwen Yu, Nuo Li, Helei Cui, Lina Yao, and Bin Guo. 2023. Quantifying predictability of sequential recommendation via logical constraints. *Frontiers of Computer Science* 17, 5 (2023), 175612.

[48] Hefei Xu, Min Hou, Le Wu, Fei Liu, Yonghui Yang, Haoyue Bai, Richang Hong, and Meng Wang. [n. d.]. Fair Personalized Learner Modeling Without Sensitive Attributes. In *THE WEB CONFERENCE 2025*.

[49] Yonghui Yang, Le Wu, Yuxin Liao, Zhuangzhuang He, Pengyang Shao, Richang Hong, and Meng Wang. 2025. Invariance Matters: Empowering Social Recommendation via Graph Invariant Learning. *arXiv preprint arXiv:2504.10432* (2025).

[50] Zhengyi Yang, Xiangnan He, Jizhi Zhang, Jiancan Wu, Xin Xin, Jiawei Chen, and Xiang Wang. 2023. A generic learning framework for sequential recommendation with distribution shifts. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 331–340.

[51] Hongyi Zhang. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017).

[52] Rui Zhou, Xian Wu, Zhaopeng Qiu, Yefeng Zheng, and Xu Chen. 2023. Distributionally Robust Sequential Recommnedation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 279–288.