Test-Time Adaptation on Recommender System with Data-Centric Graph Transformation

Yating Liu[1] , **Xin Zheng**[2] , **Yi Li**[1] , **Yanqing Guo**[1*]
[1]Dalian University of Technology, China
[2]Griffith University, Australia
yatingliu@mail.dlut.edu.cn, xin.zheng@griffith.edu.au, liyi@dlut.edu.cn, guoyq@dlut.edu.cn

Abstract

Distribution shifts in recommender systems between training and testing in user-item interactions lead to inaccurate recommendations. Despite the promising performance of test-time adaptation technology in various domains, it still faces challenges in recommender systems due to the *impracticality* of fine-tuning models and the *infeasibility* of obtaining test-time labels. To address these challenges, we first propose a **T**est-**T**ime **A**daptation framework for **G**raph-based **Rec**ommender system, named **TTA-GREC**, to dynamically adapt user-item graphs at test time in a data-centric way, handling distribution shifts effectively. Specifically, our TTA-GREC targets KG-enhanced GNN-based recommender systems with three core components: (1) Pseudo-label guided UI graph transformation for adaptive improvement; (2) Rational score guided KG graph revision for semantic enhancement; and (3) Sampling-based self-supervised adaptation for contrastive learning. Experiments demonstrate TTA-GREC's superiority at test time and provide new data-centric insights on test-time adaptation for better recommender system inference.

1 Introduction

Recommender systems powered by knowledge graphs (KGs) and modeled by graph neural networks (GNNs) represent a branch of graph-based recommender systems, *i.e.*, KGNNs. These systems [Guo *et al.*, 2020; Choudhary *et al.*, 2021; Yang *et al.*, 2023] play crucial roles in capturing complex relationships and semantic information between users and items, enabling highly accurate, personalized, and context-aware recommendations. Typically, KGNNs learn from both user-item (UI) interaction graphs and knowledge graphs (KGs), which enrich user-item relationships through KGs and capture high-order contextual information through graph convolutions, leading to high-quality user-item representations.

However, a major challenge for KGNN-based recommender systems is distribution shift between training and test data [Wu *et al.*, 2024]. It shows in user-item relationship changes, item popularity fluctuations, and contextual factors. First, the dynamic nature of user preferences leads to discrepancies between user behaviors during training and test phases [Shen *et al.*, 2023; Shafiloo *et al.*, 2024]. Second, fluctuations in item popularity may cause users to interact with cold or newly introduced items during testing, which are poorly represented in the training data, thereby reducing recommendation accuracy [Zhang *et al.*, 2023; He *et al.*, 2024]. Finally, contextual variations, such as seasonality and marketing campaigns [Adomavicius *et al.*, 2021; Meng *et al.*, 2022], further exacerbate the differences between interactions in the training and test stages. These distribution shifts result in a significant decrease in model performance during testing.

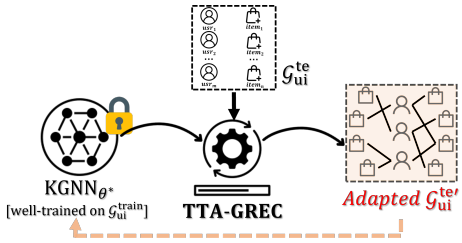

Figure 1: Test-time adaptation on recommender system.

Inspired by test-time adaptation (TTA) methods in computer vision [He *et al.*, 2021; Azimi *et al.*, 2022; Roy *et al.*, 2023], TTA has emerged as a promising strategy for dynamically adjusting data or fine-tuning model parameters at test time to address distribution shifts, improving generalization. However, existing TTA methods fail in recommender systems due to key challenges: C1: Impracticality of fine-tuning models: Model updates during testing are resource-intensive and introduce delays, so it is unsuitable for real-time recommendations [Hou *et al.*, 2023; Xi *et al.*, 2024]. C2: Infeasibility of obtaining test-time labels: Sparse data and absent supervision hinder semantic learning and adaptation [Lee *et al.*, 2018].

To address these challenges, we first propose a novel **T**est-**T**ime **A**daptation framework for **G**raph-based **Rec**ommender system, named **TTA-GREC**, to dynamically adapt user-item graphs at test time in a data-centric manner, enhancing the capability of recommender system to handle the distribu-

* Corresponding author.

tion shift issue. As shown in Figure 1, our proposed TTA-GREC dynamically adapts the test-time graphs in a data-centric manner, without fine-tuning models.

Specifically, our proposed TTA-GREC consists of three key modules: (a) **Pseudo-label guided UI graph transformation** for adaptive improvement: It refines the user-item graph by generating pseudo-labels, pruning edges, and adaptively sampling key interactions to enhance semantic information; (b) **Rational score guided KG graph revision** for semantic enhancement: Rational score is computed by combining entity and relation embeddings. It induces a series of self-supervised tasks to achieve semantic enhancement and thus robust adaptation. (c) **Sampling-based self-supervised adaptation** for contrastive learning: It enhances semantic representations through a set of self-supervised tasks, enabling robust adaptation through contrastive learning. Extensive experiments on multiple public datasets demonstrate that TTA-GREC significantly outperforms existing methods on key metrics such as Recall and NDCG (*e.g.*, 4.46% Recall and 1.86× NDCG improvement in Last-FM).

In summary, the contributions of our proposed method are listed as follows:

- **First TTA Framework on Recommendation.** To the best of our knowledge, we are the first to propose a **T**est-**T**ime **A**daptation framework for **G**raph-based **Rec**ommender system, *i.e.*, **TTA-GREC**, to dynamically adapt user-item graphs at test time in a data-centric manner for better model generalization and adaptability.
- **Dual UI and KG Graph Data-centric Transformation.** Our proposed TTA-GREC contains (a) Pseudo-label guided UI transformation, (b) Rational score guided KG graph revision, and (c) Sampling-based self-supervised adaptation.
- **Superior Test-Time Performance.** Extensive experiments conducted on multiple public datasets demonstrate that our TTA-GREC significantly enhances the test-time inference ability of KGNN-based recommendation systems.

2 Related Work

Test-Time Adaptation. Test-time adaptation (TTA) seeks to perform real-time refinements of test data or adjust the model parameters during the model deployment phase [Jha *et al.*, 2021; Ashraf *et al.*, 2022; Zheng *et al.*, 2024; Huang *et al.*, 2025]. This is done to enhance the model's generalization ability in conditions where there might be distribution shifts between the training and testing datasets. The existing TTA methods can mainly be divided into two categories: (1) Data-centric adaptation: These strategies expand or perturb the data, helping the model generalize well to the unseen data [Zheng *et al.*, 2023]. For instance, DropEdge counters overfitting by uniformly dropping some edges [Rong *et al.*, 2019]. Interaction masking or feature perturbation methods help in fusing the graphs [Mishra *et al.*, 2020; Shanmugam *et al.*, 2021]. (2) Model-centric adaptation: This approach involves modifying the model parameters during testing. This can be done by fine-tuning with self-supervised learning [Wang *et al.*, 2020], or employing contrastive learning [Chen *et al.*, 2020; Bu *et al.*, 2024]. Such approaches remain inadequate for complex recommenders, such as KGNN-based systems that handle both UI and KG elements.

3 Test-Time Adaptation on Graph-based Recommendation System (TTA-GREC)

Preliminary. For a user-item (UI) interaction graph, we have $\mathcal{G}_{\text{ui}} = \{(u, v, y_{uv})\}$, where $u \in \mathcal{U}$ represents a user in the user set $\mathcal{U}$ and $v \in \mathcal{E}$ represents an item belonging to the entity set $\mathcal{E}$. $y_{uv} = 1$ indicates that user u interacted with item v, while $y_{uv} = 0$ indicates no interaction. For a knowledge graph (KG), we have $\mathcal{G}_{\text{kg}} = \{(h, r, t)\}$, which is represented as a triple set with a head entity $h \in \mathcal{E}$, a tail entity $t \in \mathcal{E}$, and a relation $r \in \mathcal{R}$, where $\mathcal{E}$ and $\mathcal{R}$ denote the entity set and relation set in the knowledge graph, respectively.

Problem Definition. The objective of our TTA-GREC is to optimize the test UI graph $\mathcal{G}_{\text{ui}}^{\text{te}}$ dynamically at test time by constructing a test-time adaptation function $\mathcal{F}(u, v|\mathbf{\Phi}, \mathcal{G}_{\text{ui}}, \mathcal{G}_{\text{kg}}, \text{KGNN}_{\boldsymbol{\theta}^*})$. Here $\mathcal{F}(\cdot)$ is a function with learnable parameters $\mathbf{\Phi}$ for test-time UI graph transformation. Additionally, $\text{KGNN}_{\boldsymbol{\theta}^*}$ refers to the well-trained KGNN model optimized on the training set with parameters $\boldsymbol{\theta}^*$, where its learning objective is to predict the interaction likelihood between a user u and an unseen item v. In our test-time adaptation process, $\text{KGNN}_{\boldsymbol{\theta}^*}$ remains **frozen** without any parameter fine-tuning, emphasizing our data-centric approach.

KGNN Training. In the training phase of KGNN-based recommender models, the primary objective is to effectively capture the relationships among users, items, and KG entities, allowing the learning of expressive representations of user preferences and item attributes. Given a KG $\mathcal{G}_{\text{kg}}$ and a training UI graph $\mathcal{G}_{\text{ui}}^{\text{tr}}$, the optimization objective can be written as:

$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{rec}}(\text{KGNN}_{\boldsymbol{\theta}}|\mathcal{G}_{\text{kg}}, \mathcal{G}_{\text{ui}}^{\text{tr}}), \tag{1}$$

where $\mathcal{L}_{\text{rec}}(\cdot)$ is the recommendation loss (*e.g.*, binary cross-entropy loss or matrix factorization loss), and $\text{KGNN}_{\boldsymbol{\theta}}$ captures user and item features through the following function:

$$\mathbf{e}'_u, \mathbf{e}'_h = \text{KGNN}_{\boldsymbol{\theta}}\left(\mathbf{e}_u, \mathbf{e}_r, \mathbf{e}_h\right), \tag{2}$$

where $\{\mathbf{e}_u, \mathbf{e}_h, \mathbf{e}_r\} \in \mathbb{R}^d$ represent embeddings of user u, entity h from KG, and relation r from UI and KG, respectively, and $\mathbf{e}'_u, \mathbf{e}'_h \in \mathbb{R}^{d_1}$ denote the updated user and entity embeddings from KGNN by propagating and aggregating neighbor information through GNN. Then, these embeddings are used to predict the interaction probability between user u and item v, providing interaction prediction results for the recommendation system. After training, the KGNN model learns the optimal parameters $\boldsymbol{\theta}^*$, resulting in a well-trained recommender model $\text{KGNN}_{\boldsymbol{\theta}^*}$.

Test-Time KGNN Inference. For a real-world test user-item list with no existing connections, we still use graph-based representations for consistent expression, defined as $\mathcal{G}_{\text{ui}}^{\text{te}} = (u^{\text{te}}, v^{\text{te}})$. We assume that there may be a distribution shift between the training set $\mathcal{G}_{\text{ui}}^{\text{tr}}$ and the test set $\mathcal{G}_{\text{ui}}^{\text{te}}$.

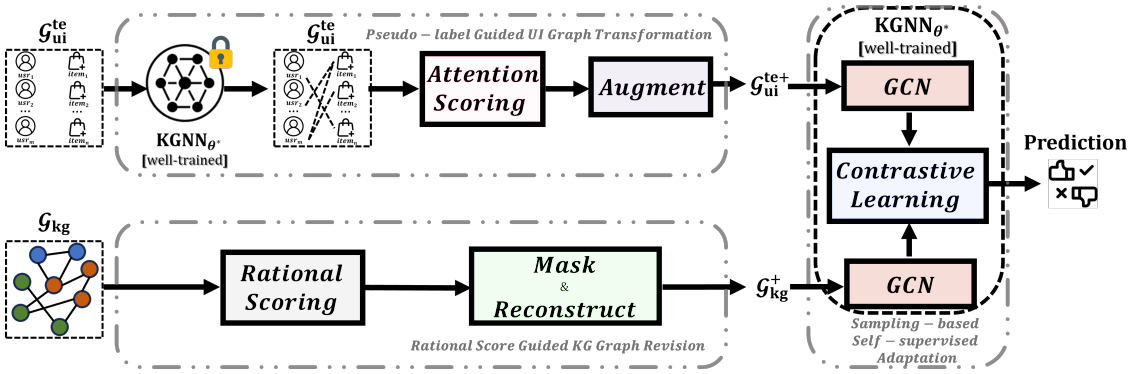

Figure 2: Overview of the TTA-GREC framework. The pseudo-label guided UI transformation module optimizes test graph structures to align training and testing distributions. The rational score guided KG revision module leverages self-supervised feature reconstruction to handle interaction data. Finally, the contrastive learning framework ensures robust embeddings by aligning representations under noisy and variable conditions.

This shift primarily manifests as differences in the joint distribution of users and items, *i.e.*, $P_{\text{tr}}(u, v) \neq P_{\text{te}}(u, v)$. Typically, the trained model $\text{KGNN}_{\boldsymbol{\theta}^*}$ is used directly to infer the test graph, such as $\mathbf{e}_u, \mathbf{e}_v = \text{KGNN}_{\boldsymbol{\theta}^*}(\mathcal{G}_{\text{ui}}^{\text{te}}, \mathcal{G}_{\text{kg}})$. However, due to potential distribution shifts at test time, the optimal parameters $\boldsymbol{\theta}^*$ learned from the training graph may not always be suitable for inference on the test graph. This could negatively impact the accuracy of predicting item interactions in the test graph, thereby harming the generalizability of the KGNN model.

3.1 Methodology

In this work, we propose a data-centric solution by learning optimal test-time UI and KG transformation distribution, named **TTA-GREC**, which aims to improve performance through test-time graph data transformation, thereby achieving better generalization for GNNs. As shown in Figure 2, our TTA-GREC is composed of three main modules: (1) **Pseudo-label guided UI graph transformation** for adaptive improvement: This module dynamically adjusts the test-phase user-item interaction graph by generating pseudo-labels, pruning noisy edges, and performing adaptive edge sampling. It enhances the semantic structure of the interaction graph by focusing on critical relationships, thus mitigating distribution shifts caused by unseen interactions during testing; (2) **Rational score guided KG graph revision** for semantic enhancement: Rationality scores are computed by combining entity and relation embeddings, guiding self-supervised tasks such as edge masking and refactoring to enhance semantics and refine the user-item graph. These tasks enable robust adaptation to noisy test data by learning meaningful representations without supervised labeling; (3) **Sampling-based self-supervised adaptation** for contrastive learning: By constructing positive and negative sample pairs, this module leverages contrastive learning to align user and item representations. It reduces noise interference and ensures the semantic consistency of embeddings across test-time data variations, improving the robustness and generalization of test-time recommendation. More detailed modular designs of our proposed TTA-GREC are presented below.

3.2 Modular Design

Pseudo-label Guided UI Graph Transformation. We have three components in this module: (a) Simulate potential interaction; (b) Attention scoring and edge sampling; and (c) Graph augmentation strategy.

(a) Simulate potential interaction. Given a set of M test users $\mathcal{U}^{\text{te}} = \{u_1^{\text{te}}, u_2^{\text{te}}, \cdots, u_M^{\text{te}}\}$ and a set of all N items $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$, there are no pre-labeled interaction relationships y_{uv} between these test users and items. To construct a new test graph, use the trained $\text{KGNN}_{\boldsymbol{\theta}^*}$ model to predict the p items that each user $u \in \mathcal{U}^{\text{te}}$ is most likely to interact with, generating a test item set $\mathcal{V}^{\text{te}} = \{v_1^{\text{te}}, v_2^{\text{te}}, \cdots, v_p^{\text{te}}\}$. These predicted interaction relationships serve as pseudo-labels y_{uv}^{te}, describing the potential interactions between users and items. Finally, we combine the test user set, the test item set, and the pseudo-labels into a new test graph $\mathcal{G}_{\text{ui}}^{\text{te}} = (\mathcal{U}^{\text{te}}, \mathcal{V}^{\text{te}}, y_{uv}^{\text{te}})$.

(b) Attention scoring and edge sampling. The attention score $\alpha_{(u,v)}$ of user-item interactions is calculated based on user and item embeddings and edge features in the interaction graph. Given a test graph $\mathcal{G}_{\text{ui}}^{\text{te}} = (V, E_{\text{ui}})$, where V is all the nodes of the original graph and E_{ui} is the set of edges. For each user-item interaction edge E_{ui}, obtain the user embedding $\mathbf{e}_u = f_u(\mathcal{G}_{\text{ui}}^{\text{te}}, u)$ and the item embedding representation $\mathbf{e}_v = f_u(\mathcal{G}_{\text{ui}}^{\text{te}}, v)$. Based on the embeddings of users and items, as well as the edges E_{ui} of the interaction graph, calculate the attention score $\alpha_{(u,v)}$ to measure the degree of user attention to each item:

$$\alpha_{(u,v)} = \text{Softmax}\left(\mathbf{e}_u^\top \mathbf{e}_v / \sqrt{d}\right). \tag{3}$$

During the inference (testing) stage, these attention scores can be aggregated or averaged to identify which interaction pairs are most significant for user preferences. The attention score can be further normalized:

$$\alpha_{(u,v)} = \frac{\exp\left(\mathbf{e}_u^\top \mathbf{e}_v / \sqrt{d}\right)}{\sum_{j \in \mathcal{E}} \exp\left(\mathbf{e}_u^\top \mathbf{e}_j / \sqrt{d}\right)}. \tag{4}$$

To increase the diversity of sampling, a double logarithmic transformation noise perturbation ϵ is used, the Gumbel dis-

tribution is introduced to enhance exploration, and finally the perturbed sampling probability distribution $P(e)$ is obtained:

$$P(e) = \frac{\exp(\alpha_{(u,v)} + \epsilon)}{\sum_{(u,v)' \in E_{\text{ui}}} \exp(\alpha_{(u,v)'} + \epsilon)}, \tag{5}$$

$$\epsilon = -\log(-\log(U)), \quad U \sim \text{Uniform}(0,1). \tag{6}$$

Sampling is performed based on this probability distribution $P(e)$ to obtain the edge set E_{sample}, and then a new user-item interaction enhanced graph $\mathcal{G}_{\text{ui}}^{+} = (V, E_{\text{sample}})$ is constructed. By retaining edges with high attention and discarding edges with low attention, the model can adaptively streamline the graph structure, making the user-item interaction graph closer to the real preference relationships.

(c) Graph augmentation strategy. The enhanced user-item interaction graph $\mathcal{G}_{\text{ui}}^{+}$ is input into the GCN of the $\text{KGNN}_{\boldsymbol{\theta}^*}$ model to update node embeddings, and by optimizing the similarity between user embedding vectors and positive sample item embedding vectors while increasing the distance between user embedding vectors and negative sample item embedding vectors, thereby improving recommendation performance.

The calculation of the user embedding $\mathbf{e}_u$ and the item embedding $\mathbf{e}_v$ is updated through the graph neural network layers in the optimized interaction graph $\mathcal{G}_{\text{ui}}^{+}$. The embedding update formula is:

$$\mathbf{e}_u^{(l+1)} = \sigma\left(\sum_{v \in \mathcal{N}(u)} \frac{\mathbf{W}^{(l)} \cdot \mathbf{e}_v^{(l)}}{\sqrt{|\mathcal{N}(u)| \cdot |\mathcal{N}(v)|}}\right), \tag{7}$$

$$\mathbf{e}_v^{(l+1)} = \sigma\left(\sum_{v \in \mathcal{N}(v)} \frac{\mathbf{W}^{(l)} \cdot \mathbf{e}_u^{(l)}}{\sqrt{|\mathcal{N}(v)| \cdot |\mathcal{N}(u)|}}\right), \tag{8}$$

where $\mathcal{N}(u)$ is the neighbor set of user u; $\mathbf{W}^{(l)}$ is the weight matrix of the l-th layer; σ is the activation function.

Rational Score Guided KG Graph Revision. We have two components in this module: (a) Rational scoring function; and (b) Adaptive masking and reconstruction.

(a) Rational scoring function. The importance of the semantics and relationships contained in different triples (h, r, t) in the knowledge graph may vary significantly. Different user preferences for similar products may be influenced by various complex relationships. Hence, the design goal of the rational weighting function is to enhance the role of reliable relationships while suppressing the negative impact of potential noise and low-correlation relationships on model performance. Inspired by the Heterogeneous Graph Transformer (HGT) [Hu *et al.*, 2020], we propose the rational weighting score $y(h, r, t)$:

$$y(h,r,t) = \text{Softmax}_h\left(\frac{\mathbf{e}_u\mathbf{W} \cdot (\mathbf{e}_t\mathbf{W} \odot \mathbf{e}_r)}{\sqrt{d}}\right) \cdot \deg(h). \tag{9}$$

Here, $\mathbf{e}_u$ and $\mathbf{e}_t$ are the embeddings of the head node and the tail node, respectively, $\mathbf{W}$ is a linear projection matrix $\mathbb{R}^{d \times d}$, where d is the hidden dimension, $\odot$ is the element-wise dot product used to fuse the relationship embedding $\mathbf{e}_r$, Softmax_h indicates the normalization of all edges of the head node h, and $\deg(h)$ is the degree of the head node h.

(b) Adaptive masking and reconstruction. Given a knowledge graph $\mathcal{G}_{\text{kg}} = (\mathcal{E}, E_{\text{kg}})$, its edge set is defined as $E_{\text{kg}} = \{(h, r, t) | h \in \mathcal{E}, t \in \mathcal{E}, r \in \mathcal{R}\}$. Edges with high attention scores are usually more important in semantic expression. By masking these high-importance edges, the model can extract potential structures and relationships from secondary edges when key semantic information is missing. Add noise ϵ to the rational score $y(h, r, t)$ for perturbation and select the top k edges with the highest scores for masking:

$$\text{Top}_k = \text{argsort}(y(h,r,t) + \epsilon)[:k]. \tag{10}$$

The noise $\epsilon = -\log(-\log(U))$, where $U \sim \text{Uniform}(0,1)$.

The set of high-attention edges E_{top} is obtained:

$$E_{\text{top}} = \{(h_i, r_i, t_i) | i \in \text{Top}_k\}. \tag{11}$$

Masking only high-attention edges may cause the model to focus too much on prominent features. To avoid the model focusing too much on prominent features and ignoring potential long-tail features, a random mask Random_k covering the entire graph is further introduced:

$$E_{\text{rand}} = \{(h_i, r_i, t_i) | i \in \text{Random}(\{1, 2, \ldots, |E|\})\}. \tag{12}$$

Combining the high-attention edge mask and the random mask, the final mask set is obtained:

$$E_{\text{mask}} = E_{\text{top}} \cup E_{\text{rand}}. \tag{13}$$

We convert the original knowledge graph $\mathcal{G}_{\text{kg}}$ into a new graph structure $\mathcal{G}_{\text{kg}}^{+} = (\mathcal{E}, E_{\text{new}})$, where $E_{\text{new}} = E_{\text{kg}} \setminus E_{\text{mask}}$. On the new graph $\mathcal{G}_{\text{kg}}^{+}$, the head node embedding $\mathbf{e}_h = f_k(\mathcal{G}_{\text{kg}}^{+}, h)$ and the tail node embedding $\mathbf{e}_t = f_k(\mathcal{G}_{\text{kg}}^{+}, t)$ are calculated, and these embeddings are used to optimize the objective function.

Sampling-based Self-supervised Adaptation. We have two components in this module: (a) Adaptive sampling; and (b) Latent space projection.

(a) Adaptive sampling. We dynamically select representative UI and KG edges based on the attention scores of the edges to enhance the learning effect of the model, respectively. Adaptive sampling of KG edges filters out edges with higher rational weighting scores $y(h, r, t)$ in $\text{Top}_k(y(h, r, t))$ to retain important relationship information while discarding edges with low correlation. Given the knowledge graph $\mathcal{G}_{\text{kg}} = (\mathcal{E}, E_{\text{kg}})$, create a Boolean mask

$$M_{\text{kg}}[i] = \begin{cases} \text{True}, & h, r, t \in \text{Top}_k y(h,r,t) \\ \text{False}, & x \geq 0 \end{cases}. \tag{14}$$

The retained edge indices E'_{kg} and their types T'_{kg} are selected through the mask M_{kg}:

$$E'_{\text{kg}} = E_{\text{kg}} \cdot M_{\text{kg}}, \tag{15}$$

$$T'_{\text{kg}} = T_{\text{kg}} \cdot M_{\text{kg}}. \tag{16}$$

Adaptive sampling of UI edges uses the average attention $\alpha_{(u,v)}$ of items as the basis of the sampling probability $P(e)$

to ensure that edges with high attention are more likely to be retained. The sampled edges according to probability:

$$E'_{\text{ui}} = \text{sample}(E_{\text{ui}}, P(e), k). \tag{17}$$

To maintain the average of the overall weight, adjust the weight of the sampled edges:

$$\mathbf{W}'_{\text{ui}} = \mathbf{W}_{\text{ui}}/(1-\phi), \tag{18}$$

where ϕ represents the edge drop rate, the number of retained edges is $k = \lfloor (1-\phi) \times |E| \rfloor$, $\mathbf{W}_{\text{ui}}$ is the original edge weight.

(b) Latent space projection. In our proposed TTA-GREC, a two-layer multi-layer perceptron (MLP) is used to perform nonlinear transformation on the embedding vectors, thereby enhancing the expressive ability of embedding representation and the effect of contrastive learning. First, the GCN of the $\text{KGNN}_{\boldsymbol{\theta}^*}$ model is used to encode the user-item interaction graph and the knowledge graph, respectively, to obtain enhanced item embedding vectors:

$$\mathbf{X}_v^{\text{kg}} = \text{GCN}_{\text{kg}}(\mathbf{e}_h, E'_{\text{kg}}, T'_{\text{kg}}), \tag{19}$$

$$\mathbf{X}_v^{\text{ui}} = \text{GCN}_{\text{ui}}(\mathbf{e}_u, \mathbf{e}_v, E'_{\text{ui}}, \mathbf{W}'_{\text{ui}}). \tag{20}$$

To align the embedding vectors in the user-item interaction graph and the knowledge graph, this paper uses a two-layer multi-layer perceptron (MLP) to map the embedding vectors into the latent space. The specific process is as follows:

$$\mathbf{z}^* = \text{ReLU}(\mathbf{W}_2 \cdot \text{ReLU}(\mathbf{W}_1 \cdot \mathbf{X}_v^* + b_1) + b_2). \tag{21}$$

Through the nonlinear transformation of the above two-layer MLP, the embedding vectors are mapped into the latent space to form more expressive and discriminative representations.

3.3 Optimization Objective

Mask Reconstruction Loss $\mathcal{L}_{mae}$. The rationale for this process is that by filtering out critical and randomly important edges through the masked edge set, the new graph structure can more effectively capture the core semantic information in the knowledge graph while reducing the interference of noise and irrelevant relationships, thereby improving the recommendation performance and generalization ability of the model. After masking, the goal of the model is to recover the masked information as much as possible. The masked edges are reconstructed using a dot product decoder, and the reconstruction optimization objective of the masked edges is defined as follows:

$$\mathcal{L}_{\text{mae}} = -\frac{1}{|E_{\text{mask}}|} \sum_{E_i \in E_{\text{mask}}} \log \sigma\left(\mathbf{r}'\right), \tag{22}$$

where $\mathbf{r}' = \langle \mathbf{e_h}, \mathbf{e_t} \odot \mathbf{r} \rangle$, and $\sigma(\cdot)$ represents the sigmoid function.

Recommendation Loss $\mathcal{L}_{rec}$. In the user-item interaction graph dynamic optimization module, the recommendation loss is used to optimize the relationship between user embeddings and item embeddings of positive and negative samples. The optimization function is defined as:

$$\mathcal{L}_{\text{rec}} = -\frac{1}{|\mathcal{D}|} \sum_{(u, v_{\text{pos}}, v_{\text{neg}}) \in \mathcal{D}} \log \sigma\left({\mathbf{e}_u}^\top \mathbf{e}_{v_{\text{pos}}} - {\mathbf{e}_u}^\top \mathbf{e}_{v_{\text{neg}}}\right), \tag{23}$$

where $\mathcal{D}$ is a set of triples containing users, positive samples, and negative samples. Through the graph augmentation strategy, the structure of UI graph is more in line with real user preferences.

Contrastive Learning Loss $\mathcal{L}_{cl}$. The objective of the optimization function is to maximize the similarity between positive samples and minimize the similarity between positive and negative samples through contrastive learning. To construct negative sample pairs, a random permutation method is used to generate negative sample indices j, thus forming negative sample pairs of $\left(\mathbf{z}_i^{\text{ui}}, \mathbf{z}_j^{\text{kg}}\right)$ and $\left(\mathbf{z}_j^{\text{ui}}, \mathbf{z}_i^{\text{kg}}\right)$. We adopt InfoNCE-based contrastive loss function as follows:

$$\mathcal{L}_{\text{cl}} = \frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp\left(s_{ii}/\tau\right)}{\sum_{j \neq i} \left(\exp\left(s_{ij}/\tau\right) + \exp\left(s_{ji}/\tau\right) \right)}. \tag{24}$$

Here, $\mathbf{s}_{ij} = \cos(\mathbf{z}_i^{\text{ui}}, \mathbf{z}_j^{\text{kg}})$, $\cos(\cdot)$ is the cosine similarity and τ is the temperature parameter.

The total loss function of the model can be expressed as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{mae}} + \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{cl}}. \tag{25}$$

By minimizing $\mathcal{L}_{\text{total}}$, the model can jointly improve recommendation performance and generalization ability in three aspects: knowledge graph enhancement, user-item interaction optimization, and contrastive learning.

4 Experiment

In this section, we answer several research questions related to TTA-GREC through experiments: **Q1:** How does the proposed TTA-GREC perform on pre-trained KGNNs for recommendation tasks under different test-time graph distribution shifts? **Q2:** How do different submodules and learning strategies in TTA-GREC contribute to its performance? **Q3:** How sensitive is TTA-GREC to hyperparameter settings? **Q4:** How does TTA-GREC perform in terms of runtime efficiency and visualization?

4.1 Experimental Settings

Datasets. We utilize three different datasets: Last-FM, MIND, and Alibaba-iFashion, which respectively represent different domains of recommendation systems. Last-FM [Wang *et al.*, 2019; Zhao *et al.*, 2019]: It is a dataset of user-music interaction logs with rich metadata. MIND [Tian *et al.*, 2021]: It is a news recommendation dataset with complex user-item interactions and semantic content. Alibaba-iFashion [Wang *et al.*, 2021]: It is a dataset focused on fashion product recommendations, featuring dynamic user preferences and detailed item attributes. We follow the procedures and partitions in previous works [Wang *et al.*, 2019; Tian *et al.*, 2021; Wang *et al.*, 2021; Yang *et al.*, 2023]. More statistical information of datasets is listed in the Appendix.

Test-time Evaluation Protocol. For each KGNN model, we follow a standard training pipeline and train it on the training set until it achieves the best performance on the validation set in terms of recommendation. These 'well-trained' GNN models remain fixed during the entire test-time adaptation

Dataset	Method	KGIN				KGCL			
		Recall	NDCG	Precision	Hit Ratio	Recall	NDCG	Precision	Hit Ratio
Last-FM	Baseline-TT	0.0873	0.0766	0.0357	0.3595	0.0971	0.0888	0.0405	0.3760
	Dropedge-TT	0.0707	0.1511	0.0578	0.3634	0.1670	0.1850	0.0773	0.5099
	Featmask-TT	0.0713	0.1205	0.0569	0.3837	0.2857	0.3008	0.1411	0.7468
	TTA-GREC (Ours)	**0.0911**	**0.2191**	**0.0969**	**0.5319**	**0.2877**	**0.3054**	**0.1433**	**0.7517**
MIND	Baseline-TT	0.0340	0.0212	0.0107	0.1795	0.0338	0.0209	0.0090	0.1573
	Dropedge-TT	0.0301	**0.0808**	0.0207	**0.3704**	0.1228	0.1065	0.0548	0.5295
	Featmask-TT	0.0281	0.0586	0.0141	0.2666	**0.2211**	**0.1791**	**0.0849**	**0.7471**
	TTA-GREC (Ours)	**0.0348**	0.0564	**0.0309**	0.3497	0.2161	0.1757	0.0839	0.7369
Alibaba-iFashion	Baseline-TT	0.1172	0.0732	0.0196	0.3238	0.1270	0.0801	0.0213	0.3464
	Dropedge-TT	0.0894	0.1733	0.0273	0.5192	0.2387	**0.1501**	0.0377	0.5637
	Featmask-TT	0.1072	0.2141	0.0317	0.5869	0.2255	0.1394	0.0356	0.5399
	TTA-GREC (Ours)	**0.1295**	**0.2526**	**0.0409**	**0.7035**	**0.2387**	0.1491	**0.0377**	**0.5640**

Table 1: Comparison of performance metrics across datasets and methods. The best and second-best performances are highlighted in **bold** and underlined, respectively.

process, ensuring that no model parameters are updated during testing. To ensure a fair comparison, we maintain consistent settings in all baselines and TTA-GREC. All evaluations are conducted under a full-ranking setup, and average performance across multiple runs is reported to ensure reliability. We evaluate performance using Recall@N and NDCG@N, with $N = 20$, to assess the model's capability in generating top-N recommendations effectively.

Baseline Methods. To evaluate the effectiveness of the proposed TTA-GREC, we compare it with the following three groups of baseline methods: (a) Knowledge graph-enhanced recommenders: we use a Baseline-TT setting where the model is traditionally trained, and the fixed model parameters are directly applied during the test phase without modification to the test data; (b) Graph structure based self-supervised learning: DropEdge [Rong *et al.*, 2019] is a widely used graph data augmentation method that randomly removes edges during the training process. We apply DropEdge-TT to modify the UI graph during the test phase. The resulting embeddings are passed to the contrastive learning module for predictions; (c) Graph feature based self-supervised learning: NodeFeatureMask [Mishra *et al.*, 2020] is a feature-level enhancement that randomly masks node features. We design a Featmask-TT approach where feature masking is applied to UI embedding during the test phase and subsequently used for contrastive learning.

4.2 Experimental Results

Performance of Test-time Recommendation. To evaluate the effectiveness of the proposed test-time adaptation framework TTA-GREC, we compare its performance with baseline methods across multiple datasets, as shown in Table 1. Across all datasets, the proposed TTA-GREC outperforms all baseline methods on key metrics for both KGIN and KGCL models. On Last-FM, TTA-GREC achieves a Recall of 0.0911 and 0.2877 for KGIN and KGCL, respectively, representing a relative improvement of 4.36% and 0.70% over the best-performing Featmask-TT. On MIND, while Featmask-TT slightly surpasses TTA-GREC in Recall for KGCL, TTA-GREC still achieves competitive results across NDCG, Precision, and Hit Ratio. For example, TTA-GREC produces an NDCG of 0.0564 for KGIN, which outperforms Baseline-TT and Dropedge-TT. On Alibaba-iFashion, TTA-GREC demonstrates superior performance, particularly in metrics that highlight overall recommendation quality. For KGIN, it achieves a Hit Ratio of 0.7035, surpassing Featmask-TT. For KGCL, TTA-GREC yields consistent improvements in Precision and NDCG.

Dataset	Model	Recall	NDCG
Last-FM	**TTA-GREC (Ours)**	**0.0911**	**0.2191**
	w/o UI transformation	0.0814	0.1520
	w/o KG revision	0.0707	0.1511
	w/o CL	0.0890	0.1761
MIND	**TTA-GREC (Ours)**	**0.0348**	0.0564
	w/o UI transformation	0.0139	0.0187
	w/o KG revision	0.0301	**0.0808**
	w/o CL	0.0265	0.0470
Alibaba-iFashion	**TTA-GREC (Ours)**	**0.1295**	**0.2526**
	w/o UI transformation	0.1175	0.1597
	w/o KG revision	0.0894	0.1733
	w/o CL	0.1137	0.1788

Table 2: Ablation Study on the contribution of submodules in TTA-GREC.

In summary, the self-supervised edge masking and reconstruction strategies employed in our framework mitigate the effects of data sparsity and improve semantic embedding quality. This explains the significant improvements in NDCG for datasets with rich KG information. By constructing positive and negative sample pairs, TTA-GREC introduces additional supervision signals, which reduce noise interference

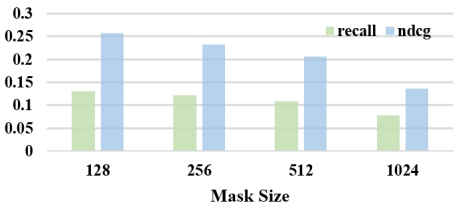

Figure 3: Hyper-parameter mask size analysis on Alibaba-iFashion dataset.

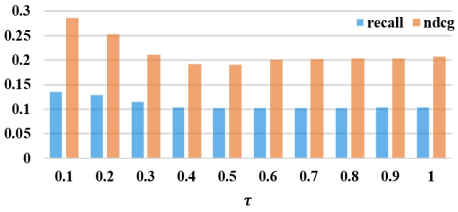

Figure 4: Hyper-parameter τ analysis on Alibaba-iFashion dataset.

and improve the discriminability of embeddings. This adaptation is particularly effective in dynamic scenarios, where user preferences frequently change.

Ablation Study of TTA-GREC. To evaluate the contribution of each submodule in TTA-GREC, we conduct ablation studies by sequentially removing: (I) **w/o UI transformation**: Removes UI graph transformation with only the original test UI list. (II) **w/o KG revision**: Removes KG transformation by only origanal KG embedding. (III) **w/o CL**: Removes sampling-based contrastive learning by the Euclidean distance of the embedding. We report the results of the ablation study in Table 2 and draw the following observations: First, removing the UI transformation module leads to a significant decline in Recall and NDCG. This indicates that this module is crucial to refining user-item interactions, eliminating noise, and optimizing key interactions. Second, removing the KG revision module has the most significant impact on the NDCG of all datasets, for example, it drops by more than 30% on Alibaba-iFashion. This highlights the critical role of this module in improving knowledge graph embeddings, enriching semantic information, and alleviating data sparsity. Lastly, removing the CL module has a relatively small impact on Recall and NDCG, but there is a significant decline on all datasets, which verifies its value in further enhancing the robustness and discriminability of the model through contrastive learning.

Hyper-parameter Sensitivity Analysis. The results in Figures 3 and 4 highlight the impact of mask size and temperature parameter on Recall and NDCG. Figure 3 shows the effect of different mask sizes on Recall and NDCG. The main observations are as follows: the best performance is achieved with a mask size of 128. This indicates that smaller mask sizes are effective in retaining key semantic information while introducing enough noise for robust representation learning. Both metrics continue to decline as the mask size increases to 256 and larger. This suggests that too large a mask may result in excessive information loss, reducing the model's ability to generate high-quality embeddings. Figure 4 shows the effect of different values of τ on Recall and NDCG. We observe that both Recall and NDCG reach their highest values when $\tau = 0.1$. This indicates that smaller temperatures can effectively balance the positive and negative sample distributions, improving contrastive learning performance. When τ exceeds 0.2, both metrics show a significant decrease. This means that a too high value of τ reduces the effectiveness of negative sampling.

Model	Dataset	Train Time (s)	Test Time (s)
KGIN	Last-FM	421.82	421.82
	MIND	851.84	851.84
	Alibaba-iFashion	354.67	354.67
DropEdge-TT	Last-FM	296.13	296.13
	MIND	2035.77	2035.77
	Alibaba-iFashion	1209.93	1209.93
Featmask-TT	Last-FM	164.21	164.21
	MIND	1130.28	1130.28
	Alibaba-iFashion	733.45	733.45
TTA-GREC (Ours)	Last-FM	488.17	488.17
	MIND	712.63	712.63
	Alibaba-iFashion	895.56	895.56

Table 3: Runtime efficiency comparison (Evaluated on NVIDIA-RTX 4090 GPU).

Running Time Comparison. Table 3 shows the training and testing times for each model. Details on KGCL are provided in the Appendix. For each method, we run the experiments until the best hyperparameters are obtained, and the report training and testing times are averaged over multiple runs. TTA-GREC achieves competitive runtime efficiency across all datasets. While TTA-GREC exhibits a moderate increase in runtime compared to lightweight models like KGIN, it significantly outperforms more computationally expensive models such as DropEdge. TTA-GREC's runtime efficiency is evident in its ability to provide test-time adaptation with minimal computational overhead. Its testing time on larger datasets, such as Alibaba-iFashion, demonstrates its scalability.

5 Conclusion

We first propose a Test-Time Adaptation framework for Graph-based Recommender system, named TTA-GREC, to address the key issue of distribution shift between training data and test data. The framework adopts a data-centric approach. TTA-GREC achieves test-time adaption through three core components: (1) Pseudo-label guided UI graph transformation for adaptive improvement, (2) Rational score guided KG graph revision for semantic enhancement, and (3) Sampling-based self-supervised adaptation for contrastive learning. Experiments conducted on public datasets demonstrate the effectiveness of the method. In the future, we will focus on more efficient TTA strategies to enhance the real-time use of online recommender systems.

Acknowledgments

This work is supported by the Major Program of the National Social Science Foundation of China under Grant No.19ZDA127.

References

[Adomavicius *et al.*, 2021] Gediminas Adomavicius, Konstantin Bauman, Alexander Tuzhilin, and Moshe Unger. Context-aware recommender systems: From foundations to recent developments. In *Recommender systems handbook*, pages 211–250. Springer, 2021.

[Ashraf *et al.*, 2022] Hassan Ashraf, Asim Waris, Muhammad Fazeel Ghafoor, Syed Omer Gilani, and Imran Khan Niazi. Melanoma segmentation using deep learning with test-time augmentations and conditional random fields. *Scientific Reports*, 12(1):3948, 2022.

[Azimi *et al.*, 2022] Fatemeh Azimi, Sebastian Palacio, Federico Raue, Jörn Hees, Luca Bertinetto, and Andreas Dengel. Self-supervised test-time adaptation on video data. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3439–3448, 2022.

[Bu *et al.*, 2024] Weixin Bu, Xiaofeng Cao, Yizhen Zheng, and Shirui Pan. Improving augmentation consistency for graph contrastive learning. *Pattern Recognition*, 148:110182, 2024.

[Chen *et al.*, 2020] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[Choudhary *et al.*, 2021] Shivani Choudhary, Tarun Luthra, Ashima Mittal, and Rajat Singh. A survey of knowledge graph embedding and their applications. *arXiv preprint arXiv:2107.07842*, 2021.

[Guo *et al.*, 2020] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering*, 34(8):3549–3568, 2020.

[He *et al.*, 2021] Yufan He, Aaron Carass, Lianrui Zuo, Blake E Dewey, and Jerry L Prince. Autoencoder based self-supervised test-time adaptation for medical image analysis. *Medical image analysis*, 72:102136, 2021.

[He *et al.*, 2024] Ming He, Han Zhang, Zihao Zhang, and Chang Liu. Invariant representation learning to popularity distribution shift for recommendation. *World Wide Web*, 27(2):10, 2024.

[Hou *et al.*, 2023] Yan-e Hou, Wenbo Gu, WeiChuan Dong, and Lanxue Dang. A deep reinforcement learning real-time recommendation model based on long and short-term preference. *International Journal of Computational Intelligence Systems*, 16(1):4, 2023.

[Hu *et al.*, 2020] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710, 2020.

[Huang *et al.*, 2025] Yujin Huang, Zhi Zhang, Qingchuan Zhao, Xingliang Yuan, and Chunyang Chen. Themis: Towards practical intellectual property protection for post-deployment on-device deep learning models. In *Proceedings of the 34th USENIX Security Symposium (USENIX Security 25)*, 2025.

[Jha *et al.*, 2021] Debesh Jha, Pia H Smedsrud, Dag Johansen, Thomas De Lange, Håvard D Johansen, Pål Halvorsen, and Michael A Riegler. A comprehensive study on colorectal polyp segmentation with resunet++, conditional random field and test-time augmentation. *IEEE journal of biomedical and health informatics*, 25(6):2029–2040, 2021.

[Lee *et al.*, 2018] Youngnam Lee, Sang-Wook Kim, Sunju Park, and Xing Xie. How to impute missing ratings? claims, solution, and its application to collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, pages 783–792, 2018.

[Meng *et al.*, 2022] Xiangwu Meng, Yulu Du, Yujie Zhang, and Xiaofeng Han. A survey of context-aware recommender systems: from an evaluation perspective. *IEEE Transactions on Knowledge and Data Engineering*, 35(7):6575–6594, 2022.

[Mishra *et al.*, 2020] Pushkar Mishra, Aleksandra Piktus, Gerard Goossen, and Fabrizio Silvestri. Node masking: Making graph neural networks generalize and scale better. *arXiv preprint arXiv:2001.07524*, 2020.

[Rong *et al.*, 2019] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

[Roy *et al.*, 2023] Subhadeep Roy, Shankhanil Mitra, Soma Biswas, and Rajiv Soundararajan. Test time adaptation for blind image quality assessment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16742–16751, 2023.

[Shafiloo *et al.*, 2024] Reza Shafiloo, Marjan Kaedi, and Ali Pourmiri. Considering user dynamic preferences for mitigating negative effects of long-tail in recommender systems. *Information Sciences*, 669:120558, 2024.

[Shanmugam *et al.*, 2021] Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1214–1223, 2021.

[Shen *et al.*, 2023] Chenglei Shen, Xiao Zhang, Wei Wei, and Jun Xu. Hyperbandit: Contextual bandit with hypernewtork for time-varying user preferences in streaming recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2239–2248, 2023.

[Tian *et al.*, 2021] Yu Tian, Yuhao Yang, Xudong Ren, Pengfei Wang, Fangzhao Wu, Qian Wang, and Chenliang

Li. Joint knowledge pruning and recurrent graph convolution for news recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 51–60, 2021.

[Wang *et al.*, 2019] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019.

[Wang *et al.*, 2020] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[Wang *et al.*, 2021] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the web conference 2021*, pages 878–887, 2021.

[Wu *et al.*, 2024] Man Wu, Xin Zheng, Qin Zhang, Xiao Shen, Xiong Luo, Xingquan Zhu, and Shirui Pan. Graph learning under distribution shifts: A comprehensive survey on domain adaptation, out-of-distribution, and continual learning. *arXiv preprint arXiv:2402.16374*, 2024.

[Xi *et al.*, 2024] Yunjia Xi, Weiwen Liu, Jianghao Lin, Muyan Weng, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Yong Yu, et al. Efficient and deployable knowledge infusion for open-world recommendations via large language models. *arXiv preprint arXiv:2408.10520*, 2024.

[Yang *et al.*, 2023] Yuhao Yang, Chao Huang, Lianghao Xia, and Chunzhen Huang. Knowledge graph self-supervised rationalization for recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3046–3056, 2023.

[Zhang *et al.*, 2023] An Zhang, Jingnan Zheng, Xiang Wang, Yancheng Yuan, and Tat-Seng Chua. Invariant collaborative filtering to popularity distribution shift. In *Proceedings of the ACM Web Conference 2023*, pages 1240–1251, 2023.

[Zhao *et al.*, 2019] Wayne Xin Zhao, Gaole He, Kunlin Yang, Hongjian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. Kb4rec: A data set for linking knowledge bases with recommender systems. *Data Intelligence*, 1(2):121–136, 2019.

[Zheng *et al.*, 2023] Xin Zheng, Yixin Liu, Zhifeng Bao, Meng Fang, Xia Hu, Alan Wee-Chung Liew, and Shirui Pan. Towards data-centric graph machine learning: Review and outlook. *arXiv preprint arXiv:2309.10979*, 2023.

[Zheng *et al.*, 2024] Xin Zheng, Dongjin Song, Qingsong Wen, Bo Du, and Shirui Pan. Online gnn evaluation under test-time graph distribution shifts. In *The Twelfth International Conference on Learning Representations*, 2024.