

Minimal-Perturbation Counterfactuals through Guided Denoising Diffusion for Recommender Systems Explanation

Amir Reza Mohammadi

Department of Computer Science, University of Innsbruck
Innsbruck, Austria
amirreza.m95@gmail.com

Michael M. Müller

Department of Computer Science, University of Innsbruck
Innsbruck, Austria
michael.m.mueller@uibk.ac.at

Andreas Peintner

Department of Computer Science, University of Innsbruck
Innsbruck, Austria
andreas.peintner@uibk.ac.at

Eva Zangerle

Department of Computer Science, University of Innsbruck
Innsbruck, Austria
eva.zangerle@uibk.ac.at

Abstract

Counterfactual explanations for recommender systems aim to identify minimal changes to a user's interaction history that would alter the model's recommendation, thereby providing actionable and user-centric explanations. However, existing CE methods often rely on local perturbation or mask-based strategies that require substantial profile modifications and exhibit unstable behavior under fine-grained ranking changes. In this work, we propose DICE REC, a diffusion-based, model-agnostic framework for generating interaction-level counterfactual explanations. DICE REC leverages denoising diffusion probabilistic models to learn the distribution of user interaction histories and integrates counterfactual objectives directly into the reverse diffusion process via guided denoising. By shaping the denoising trajectory at each step, DICE REC generates compact, semantically coherent counterfactuals that induce the desired ranking changes without relying on explicit sparsity regularization or heuristic perturbation budgets. Beyond the method itself, we introduce a fine-grained and constrained evaluation protocol that reveals important differences in explanation quality that are obscured by conventional coarse-grained, percentage-based perturbation schemes. Through extensive experiments, we show that DICE REC consistently outperforms state-of-the-art baselines, particularly in minimal-perturbation scenarios that reflect realistic counterfactual conditions. Our results highlight the advantages of diffusion-based counterfactual generation and underscore the necessity of fine-grained evaluation for faithful explanation assessment in recommender systems.

CCS Concepts

• Information systems → Recommender systems.

Keywords

Recommender Systems, Counterfactual Explanation, Denoising Diffusion Probabilistic Models, Evaluation

ACM Reference Format:

Amir Reza Mohammadi, Andreas Peintner, Michael M. Müller, and Eva Zangerle. 2026. Minimal-Perturbation Counterfactuals through Guided Denoising Diffusion for Recommender Systems Explanation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26)*, July 20–24, 2026, Melbourne, VIC, Australia. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3805712.3809584>

1 Introduction

Recommender systems (RS) have become ubiquitous in modern digital platforms, shaping user decisions across e-commerce, entertainment, and social media domains [3, 9, 31]. While these systems achieve strong predictive performance, their increasing reliance on complex, non-linear models has intensified concerns regarding transparency, accountability, and user trust [35, 36, 38]. As a result, users frequently question why particular items are recommended, making intelligible explanations a critical requirement for user satisfaction, regulatory compliance, and responsible deployment of RS [32, 33, 47].

Counterfactual explanations (CEs) have emerged as a principled approach to addressing this challenge by answering the question: “What needs to change for the recommendation to be different?” [4, 14]. In recommender systems, CEs typically identify a minimal set of modifications to a user's interaction history that would alter the model's output, thereby offering actionable and user-centric explanations [42]. Figure 1 illustrates such a counterfactual explanation in a music recommendation scenario.

Despite their conceptual appeal, existing CE methods for RS suffer from several fundamental limitations. First, many state-of-the-art approaches require relatively large perturbations to the user profile—at least modifying 10% of historical interactions—to induce a counterfactual outcome. This directly contradicts the core principle of counterfactual reasoning, which emphasizes *minimal and plausible changes* [42]. Second, prior works rely on local perturbation-based optimization, which only explores small changes around the original input. As a result, they often fail to adequately capture the complex, highly non-linear decision behavior of modern recommender models. Third, compactness is typically enforced via explicit sparsity regularization, leading to fragile trade-offs between explanation validity, sparsity, and realism. These limitations are particularly pronounced in deep learning-based RS, where user-item interactions follow complex, high-dimensional



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.

SIGIR '26, Melbourne, VIC, Australia

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2599-9/2026/07

<https://doi.org/10.1145/3805712.3809584>

distributions that are poorly approximated by local or heuristic perturbations. As a result, existing CE methods often generate explanations that are either implausible, require modifying a large number of user–item interactions to trigger a change in the recommendation, or are weakly aligned with the true data-generating process.

Diffusion models are particularly well-suited for CE in recommender systems because they generate changes through a sequence of infinitesimal, distribution-aligned denoising steps, naturally aligning counterfactual generation with the requirement of minimal and plausible interaction-level perturbations.

Motivated by this observation, we propose DICEC: **Diffusion-based Counterfactual Explanations for Recommender systems**, a novel, model-agnostic framework for generating compact counterfactual explanations. Model-agnostic in the sense that it does not depend on a specific architecture, but requires gradient access. DICEC learns the distribution of user interaction histories and integrates counterfactual objectives directly into the reverse diffusion process by guiding the noise prediction at each denoising step. This design enables the generation of valid counterfactuals that induce the desired ranking changes without relying on ad-hoc perturbation budgets or explicit sparsity regularization.

We evaluate DICEC on three widely used benchmark datasets—MovieLens-1M, Pinterest, and Yahoo! Music—across three representative recommender architectures: matrix factorization (MF), neural collaborative filtering (NCF), and variational autoencoders (VAE). Experimental results demonstrate that DICEC consistently outperforms established baselines and state-of-the-art counterfactual methods across multiple evaluation metrics, particularly in minimal-perturbation scenarios.

We summarize our main contributions as follows: (i) We introduce DICEC, a diffusion-based, model-agnostic framework for generating interaction-level CEs in RS; (ii) We propose a guided denoising mechanism that incorporates counterfactual objectives throughout the entire diffusion trajectory, enabling compact and plausible explanations; (iii) While prior work [6] introduces fixed granularity evaluation, we extend beyond and present a refined evaluation protocol that provides a constrained perturbation and realistic assessment of CE quality beyond coarse-grained perturbation schemes; (iv) Extensive experiments show that DICEC produces more compact explanations while significantly outperforming state-of-the-art baselines. To support reproducibility, we publicly release our code and experimental pipeline¹.

2 Related Work

Explainable AI (XAI) for recommender systems has received growing attention due to its importance for transparency, user trust, and regulatory compliance [10, 20]. Prior work on explainability in recommendation can be broadly organized into three themes: general explanation approaches, counterfactual explanation methods, and diffusion-based generative models:

Explanation approaches for recommender systems. A large body of work has studied explanation techniques that aim to justify recommendations by revealing influential model components, training data, or semantic relationships. Model-specific approaches

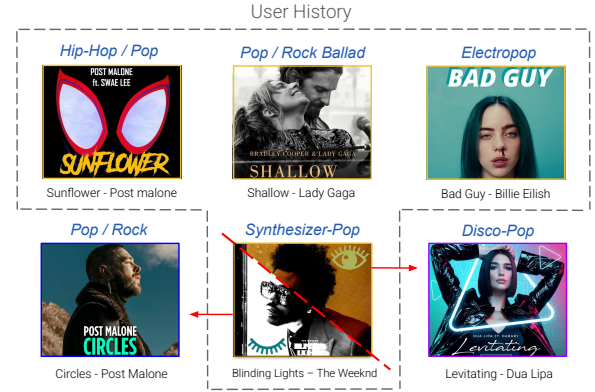


Figure 1: Illustrative example of a counterfactual explanation (CE) in a music recommender system. Given the user’s interaction history, the system recommends *Levitating*. The CE identifies *Blinding Lights* as a critical interaction whose removal would change the recommendation to *Circles*.

target particular recommendation architectures and exploit their internal structures. Early examples include explainable matrix factorization models that align latent factors with interpretable components [1, 2, 23]. Recent methods, such as ProtoMF [30] and AI2V++ [15], leverage prototypes or attention mechanisms to highlight influential interactions.

Post-hoc methods aim to explain recommendations without modifying the model. LIME [36] constructs local surrogate models, SHAP [28] attributes importance via Shapley values [17]. While SHAP and LIME are feature attribution methods rather than explainers, we include them as baselines for interaction-level importance estimation. Influence-based approaches such as FIA [12] and ACCENT [40] estimate the contribution of training interactions to predictions, extending influence functions to neural recommenders. Similarity-based heuristics such as Jaccard [5] and Cosine [39] similarity are also common, model-agnostic and computationally efficient. However, they provide correlational rather than causal explanations and do not account for ranking sensitivity.

Another line of work generates explanations using external item features, reviews, or graph structures. Aspect-based models such as the Explicit Factor Model (EFM) [48] align latent factors with explicit attributes to produce natural-language explanations. Path-based explainable recommenders leverage semantic paths in knowledge graphs to justify recommendations [11, 24]. However, recent studies have questioned the reliability of attention weights used in such models, motivating counterfactual path reasoning approaches that replace attention with causal interventions [25]. Despite their expressiveness, aspect- and path-based methods depend on rich side information and are not applicable in interaction-only settings.

Counterfactual explanation. CEs describe how a recommendation would change under hypothetical modifications to user or item data [42]. They are particularly appealing due to their causal interpretation and actionability. Early counterfactual approaches rely on heuristic perturbations or search strategies to identify influential features or interactions. Reinforcement learning-based methods, such as CERec [44] and CFairER [43], reduce the search space by

¹<https://github.com/amirreza-m95/DiffExplainer4RS>

learning policies over item attributes or graph paths, often targeting fairness or attribute-level explanations. Conversational recommenders further incorporate counterfactual reasoning to explain or improve recommendations through interactive feedback [46]. Algorithmic recourse frameworks, such as RecRec [41], focus on identifying actionable changes that alter item rankings, primarily from the perspective of content providers.

While effective, most counterfactual methods suffer from at least one of the following limitations: reliance on side information, expensive search procedures, model-specific assumptions, or evaluation restricted to single-item outcomes. The LXR framework [8] addresses scalability by introducing a self-supervised, model-agnostic approach that learns explanation masks without explicit perturbation. By embedding the recommender into the training objective, LXR establishes a strong baseline for post-hoc CE. However, LXR performs best under large perturbation budgets and does not explicitly optimize for minimal interaction-level changes or stability across the top- k recommendation list.

Diffusion-based explanations and generative models. Diffusion models are powerful generative tools in RS. Existing work primarily employs diffusion for recommendation modeling [49], fairness-aware recommendation [21], or counterfactual data augmentation to mitigate bias and improve robustness [16]. Recent surveys [27, 45] confirm that most diffusion-based RS focus on accuracy, robustness, or content generation, with explainability largely unexplored.

In the broader XAI literature, diffusion models have been used to generate CEs for classification tasks. SCD [29] introduces a plug-and-play framework that guides the reverse diffusion process to generate plausible and diverse counterfactuals for arbitrary black-box classifiers. Related work adjusts latent representations in disentangled diffusion spaces to improve the validity and plausibility of counterfactual examples [37]. However, these approaches are designed for label-based prediction tasks and do not extend to ranking-based recommendation, sparse implicit feedback, or interaction-level counterfactual reasoning.

Advancing beyond LXR. While LXR represents a significant advancement in generating accurate counterfactual explanations, it performs effectively under substantial user history modifications ($\geq 10\%$) but exhibits limited capability in *minimum-perturbation* scenarios, i.e., when the objective is to identify the smallest possible change to alter the model's recommendation. Diffusion models have been widely adopted for recommendation performance, robustness, and fairness, while diffusion-based counterfactual explainers such as SCD target classification rather than recommendation. In contrast, our proposed approach, *DICEREC*, is the first diffusion-based, model-agnostic approach for counterfactual explanation in RS that generates plausible, minimal interaction-level perturbations and directly evaluates explanation faithfulness through changes in the top- k ranking.

3 Problem Statement and Notations

Let $\mathcal{U}$ and $\mathcal{V}$ be the number of users and items, respectively. We denote a user data vector by $\mathbf{x}$ and for a specific user u , we denote their personal data vector by $\mathbf{x}_u$. In this work, we focus on explaining RS operating on implicit user-item interactions, in a

Algorithm 1: DICEREC

Input: User history $\mathbf{x}_u$, target item i_t , RS with encoder $\text{Enc}(\cdot)$ and decoder $\text{Dec}(\cdot)$, diffusion noise-predictor $\epsilon_\psi(\cdot, t)$, guidance strength λ , timesteps T

Output: Importance scores S over items in $\mathbf{x}_u$

Phase 1: Interaction Distribution Modeling
 Compute latent user embeddings $\mathbf{z} = \text{Enc}(\mathbf{x})$ for all users ;
 Set variance schedule $\{\beta_t\}_{t=1}^T$; define $\alpha_t = 1 - \beta_t$,
 $\bar{\alpha}_t = \prod_{j=1}^t \alpha_j$;
for $\text{epoch} = 1$ **to** epochs **do**
 for $\text{mini-batch } \mathbf{z}_0$ **of** latent embeddings **do**
 Sample $t \sim \text{Uniform}(\{1, \dots, T\})$, $\epsilon \sim \mathcal{N}(0, I)$;
 $\mathbf{z}_t \leftarrow \sqrt{\bar{\alpha}_t} \mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$; // forward diffusion
 $\hat{\epsilon} \leftarrow \epsilon_\psi(\mathbf{z}_t, t)$; // predict noise
 $\mathcal{L}_{\text{ddpm}} \leftarrow \|\epsilon - \hat{\epsilon}\|_2^2$;
 Update ψ by minimizing $\mathcal{L}_{\text{ddpm}}$;
 end
end
Phase 2: Guided Counterfactual Denoising (sampling)
 Initialize $\mathbf{z}_T \sim \mathcal{N}(0, I)$ and set $\mathbf{z}_t \leftarrow \mathbf{z}_T$;
for $t = T$ **to** 1 **do**
 $\hat{\epsilon} \leftarrow \epsilon_\psi(\mathbf{z}_t, t)$;
 $L_{\text{guid}} \leftarrow \text{Dec}(\mathbf{z}_t)[i_t]$; // target-score objective
 (demote i_t)
 $\mathbf{g}_t \leftarrow \nabla_{\mathbf{z}_t} L_{\text{guid}}$;
 $\hat{\epsilon}_{\text{guided}} \leftarrow \hat{\epsilon} - \lambda \mathbf{g}_t$; // guided noise
 Sample $\mathbf{w} \sim \mathcal{N}(0, I)$;
 $\mathbf{z}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{z}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \hat{\epsilon}_{\text{guided}} \right) + \sqrt{\beta_t} \mathbf{w}$;
end
 Return $S \leftarrow \text{Dec}(\mathbf{z}_0)$ restricted to items in $\mathbf{x}_u$;

model-agnostic manner and without relying on side features. Accordingly, $\mathbf{x}_u \in \{0, 1\}^{|\mathcal{V}|}$ is a binary vector that indicates the historical items consumed by u . Let $f_\theta : \{0, 1\}^{|\mathcal{V}|} \rightarrow [0, 1]^{|\mathcal{V}|}$ be a recommender, parameterized by θ , that receives user data $\mathbf{x}$, and outputs a probability distribution over the items. To transform the discrete data into continuous space, we use an Autoencoder (AE) [22] that learns the latent representation of users through an encoder $E_\theta : \{0, 1\}^{|\mathcal{V}|} \rightarrow \mathbb{R}^d$ and a decoder $D_\theta : \mathbb{R}^d \rightarrow [0, 1]^{|\mathcal{V}|}$, where d is the latent dimension. In our setting, the autoencoder and the recommender are pretrained and kept *fixed* throughout the explanation process. DICEREC performs all diffusion-based modeling and counterfactual generation entirely in the latent user embedding space $\mathbb{R}^d$, where user representations are more compact and continuous. The encoder maps user data $\mathbf{x}$ to a latent embedding $\mathbf{z} = E_\theta(\mathbf{x}) \in \mathbb{R}^d$, while the decoder reconstructs item preferences from the latent representation (see Figure 2). Given a pretrained recommender f_θ and a user data vector $\mathbf{x}$, our goal is to identify which elements of $\mathbf{x}$ are most critical for maintaining the Top-1 ranking of a specific item. In other words, we aim to generate a plausible and efficient *explanation ranking* that attributes item-level importance within the user's interaction history to a target recommendation.

An explainer $\mathcal{D}_\phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a function, parameterized by ϕ , that operates in the latent space of the recommender system. The explanation process involves three key steps: (1) encoding the user data to the latent space, (2) generating a counterfactual embedding through guided diffusion sampling, and (3) decoding the counterfactual embedding to obtain item importance scores. The explainer receives a user's latent embedding $z = E_\theta(\mathbf{x})$ and generates a counterfactual embedding $z_{cf} = \mathcal{D}_\phi(z)$. Specifically, let $\mathbf{y} \in \{0, 1\}^{|V|}$ be a one-hot vector representing the recommended item (target item) to be explained. The explainer's output is an explanation mask $\mathbf{m} = \text{decode}(\mathcal{D}_\phi(E_\theta(\mathbf{x}), \mathbf{y}))$ that produces an ordered sensitivity ranking over the user's historical interactions $\mathbf{x}$ with respect to the recommendation of the target item $\mathbf{y}$. Namely, $\mathbf{m}[i]$ represents the importance of the data in $\mathbf{x}[i]$ to the recommendation of $\mathbf{y}$.

4 DICEREC

This section presents DICEREC, our diffusion-based framework for generating counterfactual explanations in recommender systems under minimal and plausible perturbations. The central design principle of DICEREC is to generate counterfactual user representations through a sequence of incremental, distribution-aligned transformations, rather than through abrupt local perturbations or post-hoc masking strategies. By operating along the learned data manifold of user interaction histories, DICEREC aligns the counterfactual generation process with the fundamental requirement of counterfactual reasoning: identifying the smallest possible change that induces a meaningful shift in recommendation outcomes.

In contrast to existing CE methods that apply constraints or sparsity regularization after generating candidate explanations, DICEREC integrates counterfactual objectives directly into the generative process itself (see Figure 2). Specifically, we leverage denoising diffusion probabilistic models to learn the distribution of user interaction embeddings and introduce a guided denoising mechanism that steers the reverse diffusion trajectory toward counterfactual outcomes at each timestep. This process-level integration ensures that counterfactual guidance influences every stage of generation, yielding explanations that are compact.

The DICEREC framework consists of two main components: (1) *interaction distribution modeling*, which learns a diffusion-based generative model over user interaction embeddings; (2) *guided counterfactual denoising*, which incorporates recommendation-level objectives into the reverse diffusion process to induce ranking changes and *counterfactual explanation extraction* decodes the resulting counterfactual embedding into item-level importance scores. An overview of the complete procedure is provided in Algorithm 1. We describe each component in detail below.

4.1 Interaction Distribution Modeling

DICEREC requires a generative prior over *plausible* user states so that counterfactual changes can be produced as small, distribution-aligned updates rather than arbitrary perturbations of the interaction vector. We therefore model the distribution of users in the *latent interaction space* of a pretrained recommender autoencoder. Let $\mathbf{x}_u \in \{0, 1\}^{|V|}$ denote the implicit interaction history of user u , and let $\mathbf{z}_u = \text{Enc}(\mathbf{x}_u) \in \mathbb{R}^d$ be the corresponding latent embedding produced by the frozen encoder of the recommender. We learn a

diffusion model over $\{\mathbf{z}_u\}$, i.e., a density $p_\psi(\mathbf{z})$ that approximates the empirical distribution of latent user embeddings induced by observed interaction histories. This learned prior is the key mechanism that keeps our later counterfactual generation (Section 4.3) on-manifold, enabling *minimal* interaction-level explanations.

Forward diffusion in latent user space. Given a clean latent embedding $\mathbf{z}_0 \sim q(\mathbf{z})$ (with $\mathbf{z}_0$ instantiated as $\mathbf{z}_u$ for some user), we define a forward noising process that gradually corrupts $\mathbf{z}_0$ with Gaussian noise over T steps:

$$q(\mathbf{z}_t | \mathbf{z}_{t-1}) = \mathcal{N}(\mathbf{z}_t; \sqrt{\alpha_t} \mathbf{z}_{t-1}, (1 - \alpha_t)\mathbf{I}), \quad t = 1, \dots, T, \quad (1)$$

where $\alpha_t = 1 - \beta_t$ and $\{\beta_t\}_{t=1}^T$ is a variance schedule. By standard DDPM algebra, we can sample $\mathbf{z}_t$ directly from $\mathbf{z}_0$:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t} \mathbf{z}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad \bar{\alpha}_t = \prod_{j=1}^t \alpha_j. \quad (2)$$

For sufficiently large T (so that $\bar{\alpha}_T \approx 0$), the terminal distribution approaches a standard normal, $q(\mathbf{z}_T) \approx \mathcal{N}(0, \mathbf{I})$.

Reverse denoising as a latent-user generator. The reverse process learns to denoise a noisy latent user state $\mathbf{z}_t$ into $\mathbf{z}_{t-1}$:

$$p_\psi(\mathbf{z}_{t-1} | \mathbf{z}_t) = \mathcal{N}(\mathbf{z}_{t-1}; \mu_\psi(\mathbf{z}_t, t), \sigma_t^2 \mathbf{I}), \quad (3)$$

where σ_t^2 can be fixed from the schedule, and μ_ψ is parameterized via a noise-prediction network $\epsilon_\psi(\mathbf{z}_t, t)$. Concretely, using the reparameterization in Equation (2), $\epsilon_\psi(\mathbf{z}_t, t)$ predicts the injected noise component, which determines μ_ψ and yields the standard DDPM sampling update used in Algorithm 1.

Training objective over user embeddings. We train the diffusion model on latent embeddings of *observed users*: each training sample is $\mathbf{z}_0 = \mathbf{z}_u = \text{Enc}(\mathbf{x}_u)$ from the frozen recommender. At each iteration, we sample $t \sim \text{Uniform}(\{1, \dots, T\})$ and noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$, construct $\mathbf{z}_t$ via Equation (2), and minimize the noise-matching loss:

$$\mathcal{L}_{\text{DDPM}}(\psi) = \mathbb{E}_{u, t, \epsilon} \left[\|\epsilon - \epsilon_\psi(\mathbf{z}_t, t)\|_2^2 \right]. \quad (4)$$

This objective learns a score/denoising field that captures how real user embeddings concentrate on the latent manifold induced by interaction histories. In Section 4.2, we leverage this learned denoising trajectory as a *distribution-preserving backbone* and inject counterfactual guidance during sampling to demote a target recommendation while remaining close to realistic user states.

4.2 Guided Counterfactual Denoising

The objective of guided counterfactual denoising is to steer the diffusion-based generation process toward representations that induce a desired change in recommendation behavior, while remaining constrained to the learned distribution of user interaction histories. Unlike post-hoc counterfactual methods that first generate candidate explanations and subsequently enforce constraints or ranking objectives, DICEREC integrates counterfactual guidance directly into the reverse diffusion process. This design ensures that counterfactual objectives influence every stage of generation, resulting in incremental changes to user representations.

At inference time, counterfactual samples are generated by iteratively applying the reverse diffusion process. Starting from Gaussian

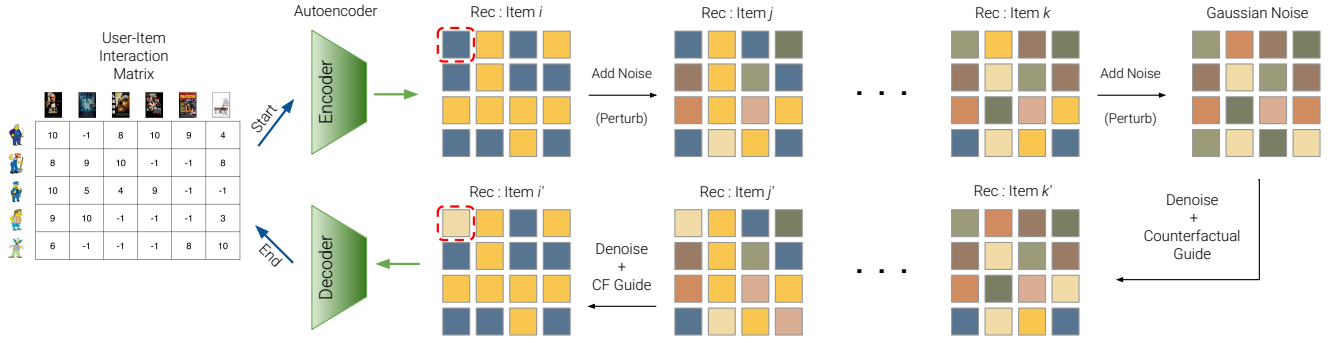


Figure 2: Illustration of the Guided Counterfactual Denoising phase of DICE REC. The red dashed contour highlights the localized modification in the embedding space that produces the counterfactual example.

noise, the reverse transition at timestep k is given by:

$$x_{k-1} = \frac{1}{\sqrt{\alpha_k}} \left(x_k - \frac{\beta_k}{\sqrt{1-\alpha_k}} \epsilon_\theta(x_k, k) \right) + \sigma_\theta(x_k, k) z, \quad (5)$$

where $z \sim \mathcal{N}(0, I)$. Repeating this procedure from $k = K$ to 1 yields a generated sample x_0 .

The core innovation of DICE REC lies in modifying the noise prediction at each timestep to incorporate counterfactual guidance. Concretely, we augment the predicted noise with a gradient-based guidance term that encourages changes in recommendation outcomes. The guided noise prediction is defined as:

$$\epsilon_{\text{guided}} = \epsilon_\theta(x_t, t) - \lambda \nabla_{x_t} \mathcal{L}_{\text{guidance}}, \quad (6)$$

where λ controls the strength of the guidance and balances fidelity to the data distribution against the magnitude of the induced counterfactual effect. The guidance loss $\mathcal{L}_{\text{guidance}}$ is designed to directly target recommendation behavior. In particular, given a target item i_t to be explained, we define:

$$\mathcal{L}_{\text{guidance}} = \text{score}(i_t, \text{decode}(x_t)), \quad (7)$$

where $\text{decode}(x_t)$ maps the current latent representation back to item preference scores via the recommender decoder, and $\text{score}(i_t, \cdot)$ extracts the predicted preference score for the target item. Minimizing this objective encourages a reduction in the target item's score, making it less likely to appear at the top of the recommendation ranking. Importantly, this formulation allows ranking changes to emerge naturally from small, accumulated perturbations rather than from explicit ranking constraints. The guided reverse diffusion step then becomes:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_{t-1}}} \epsilon_{\text{guided}} \right) + \sqrt{\beta_t} \epsilon, \quad (8)$$

where $\epsilon \sim \mathcal{N}(0, I)$, and α_t, β_t are diffusion schedule parameters.

By injecting guidance at every diffusion step, DICE REC ensures that counterfactual influence accumulates gradually throughout the sampling trajectory. This results in counterfactual embeddings that are both *plausible*, as they remain close to the learned data manifold, and *effective*, as they induce the desired ranking changes through minimal interaction-level modifications.

This guided denoising mechanism offers several advantages over existing CE approaches. First, it enables end-to-end differentiable counterfactual generation with fine-grained control over

the strength of intervention. Second, operating within the diffusion framework preserves semantic coherence by constraining updates to distribution-aligned directions. Finally, by varying the guidance strength or target objectives, DICE REC can generate multiple valid CEs, offering diverse yet faithful insights into recommendation behavior.

4.3 Counterfactual Explanation Generation

Given a trained recommender and a user interaction history $\mathbf{x}_u$, DICE REC generates counterfactual explanations by translating counterfactual embedding shifts into item-level importance scores. We first encode the user's interaction history into the latent space, obtaining a factual user representation $z_u = \text{Encoder}(\mathbf{x}_u)$. Using the guided diffusion process described in Section 4.2, we then generate a counterfactual latent representation z_u^{cf} that induces a change in the recommendation outcome while remaining close to the learned data distribution.

The counterfactual embedding z_u^{cf} is subsequently decoded through the recommender decoder to obtain item-level preference scores. These scores reflect the model's predicted affinities under the counterfactual user representation and implicitly capture how individual historical interactions contribute to sustaining or altering the original recommendation. We interpret the decoded scores corresponding to items in the user's interaction history as *importance scores*, where higher values indicate stronger influence on the original recommendation outcome. This formulation allows CEs to be expressed as an ordered ranking over the user's historical interactions, directly supporting perturbation-based evaluation protocols. By progressively removing interactions according to their inferred importance, we can assess whether the explanation correctly identifies minimal and influential changes that lead to the desired ranking shift. In this way, DICE REC produces explanations that are both faithful to the recommender's behavior and aligned with practical counterfactual evaluation criteria.

5 Experiments and Results

In this section, we describe the setup and results of extensive experiments, where we compare DICE REC to various state-of-the-art models. Given our goal of providing efficient and plausible explanations, we establish the following research questions:

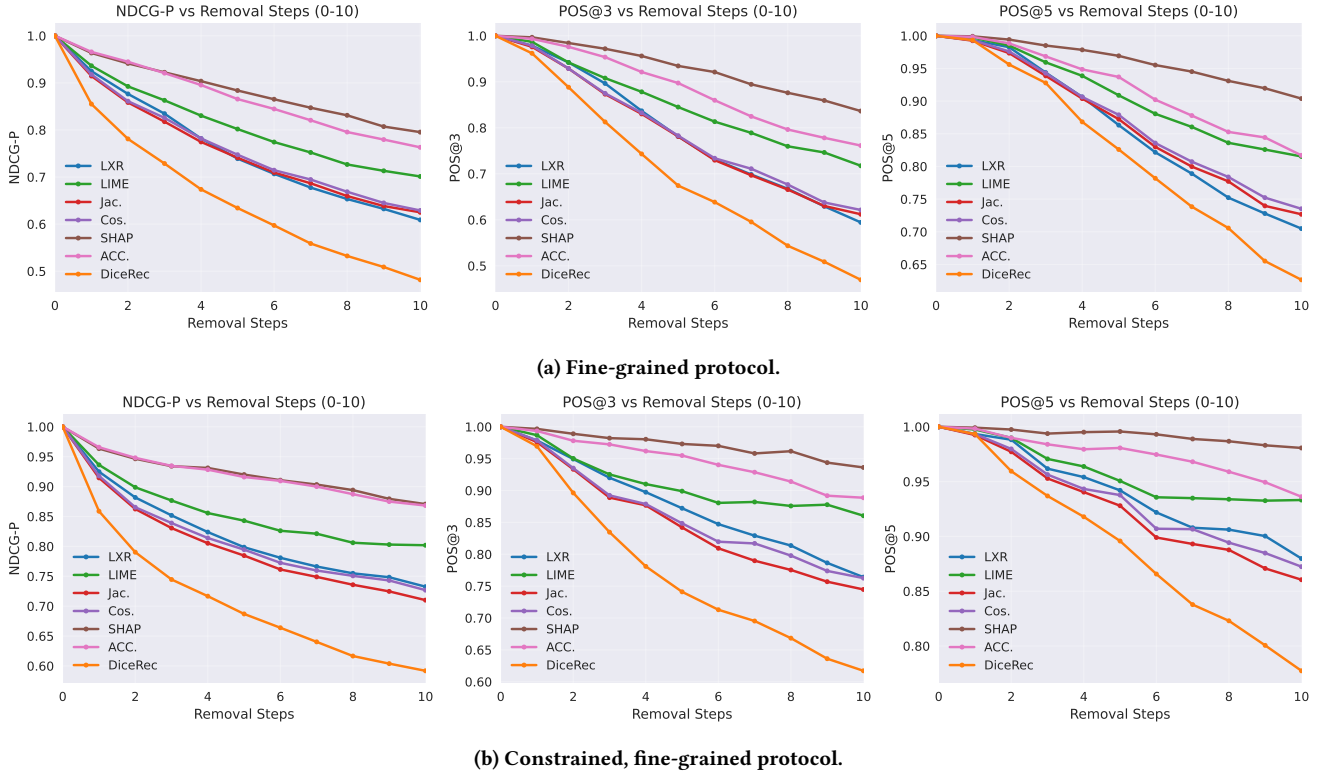


Figure 3: Comparative performance trends of methods on the *ML-1M* dataset based on *MF Recommender*: For all metrics, the less is the better. Among all methods, *DiceRec* consistently outperforms the others.

- RQ1** How does *DiceRec* perform compared to other state-of-the-art CE methods on different datasets?
- RQ2** How do different granularity levels (efficiency) effect in performance of *DiceRec* and other methods? How do different methods perform under limited perturbation (plausibility) condition?
- RQ3** To what extent do dataset-specific structural properties influence the effectiveness of counterfactual explanations under realistic perturbation constraints?
- RQ4** What is the performance of *DiceRec* in terms of computational efficiency compared to other models?

5.1 Experimental Setup

We evaluate *DiceRec* and baselines on three standard recommendation datasets that are not only widely used in recommender system research [9], but also well established benchmarks for assessing explanation approaches. They represent diverse characteristics in terms of user interaction patterns, item catalog sizes, and sparsity levels. The MovieLens-1M² (ML-1M) dataset contains 6,037 users and 3,381 items with an average of 165 interactions per user, representing a dense interaction scenario typical of established recommendation platforms. The Pinterest dataset [18] comprises 19,155 users and 9,362 items with an average of 27 interactions per user, reflecting a more moderate interaction density. The Yahoo!

²<https://grouplens.org/datasets/movielens/1m/>

Music dataset [13] includes 13,797 users and 4,604 items with an average of 12 interactions per user, representing a sparse interaction scenario common also in niche recommendation domains.

5.1.1 Baselines. Our evaluation spans established baselines and recent CE methods, including two similarity-based approaches, two traditional explainers, and two state-of-the-art methods:

Jaccard [5]/**Cosine** [39] Similarity generate explanations by comparing an item to those in the user’s history using Jaccard [5]/Cosine [39] similarity metrics based on co-interacted users.

LIME-RS [34] is a modified version of the LIME framework designed specifically for RS. It produces explanations by approximating the complex recommendation model with a locally interpretable linear model, capturing relationships within a neighborhood around the user’s data. We apply LIME-RS to the user’s historical interactions, carefully adjusting the number of samples based on performance on the validation set.

SHAP (SHapley Additive exPlanations) [50] quantifies the influence of individual features on a model’s predictions. By leveraging game theory, it assigns a value to each feature according to its contribution to the final output. Implementation that we used is tailored to recommendation tasks through Jaccard similarity clustering combined with k-means sampling where $K = 10$.

ACCENT (Action-based CEs for Neural Recommenders for Tangibility) [40] is a CE framework that extends influence functions

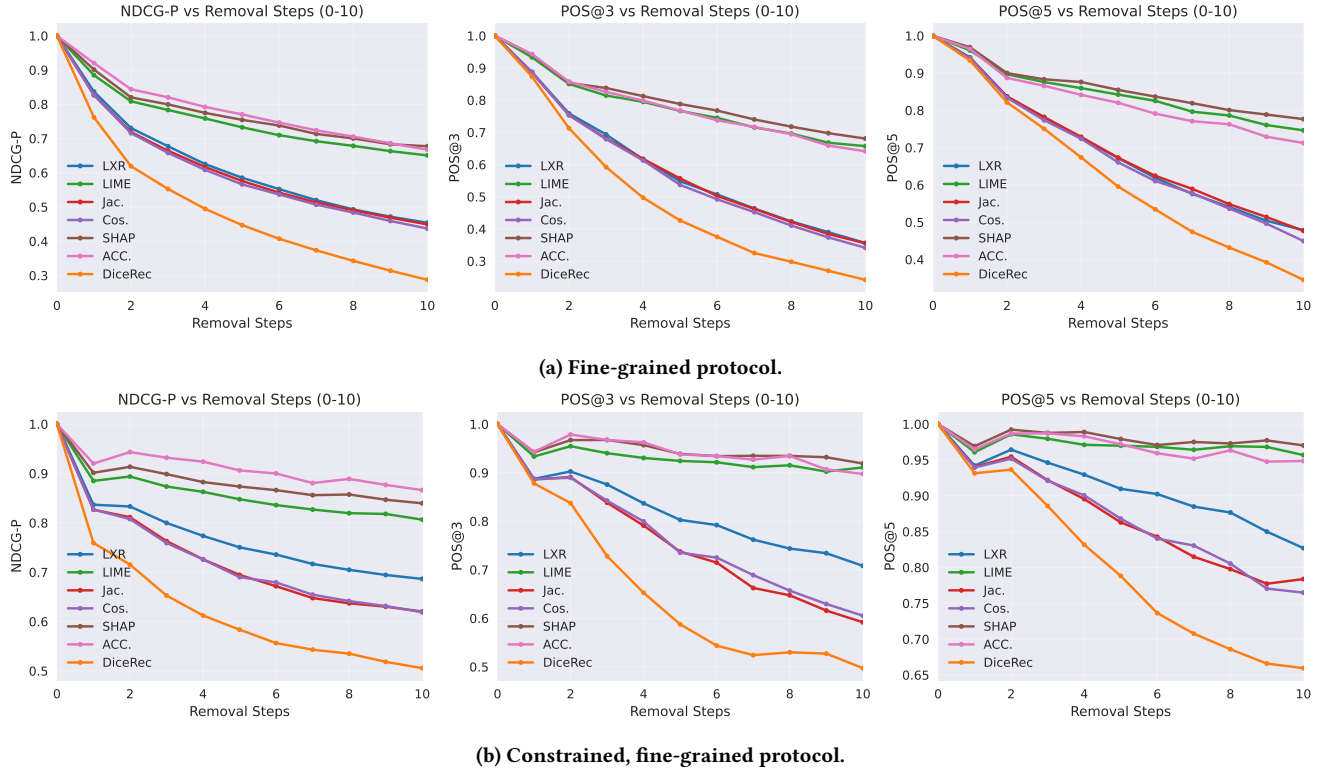


Figure 4: Comparative performance trends of various methods under two approaches of progressive user history removal on the Yahoo! Music dataset based on NCF Recommender: For all metrics, the less is the better. Among all methods, DICERec consistently outperforms the others.

to provide actionable, model-agnostic insights for neural recommenders. It extends techniques originally designed for latent factor models, making them applicable to a broader class of neural RS.

LXR [8], the state-of-the-art CE approach, employs self-supervised learning to generate explanation masks that highlight critical user data without perturbations using a gradient learning approach.

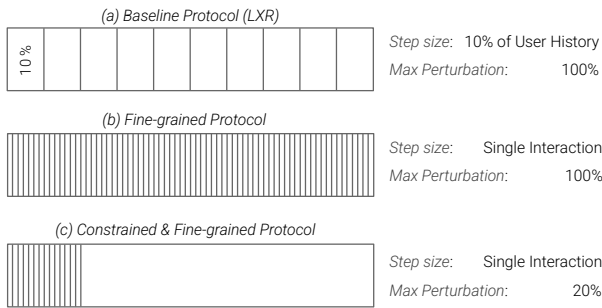


Figure 5: Illustration of the evaluation protocols. Each rectangle represents the full interaction history of a user.

Our study employs three recommenders to ensure robustness across different modeling paradigms. We use *Variational Autoencoders* (VAE), a generative model that learns probabilistic latent

representations of user-item interactions and reconstructs preferences via a decoder [26]. We adopt a VAE-based recommender with an architecture similar to [26]. We also consider *Neural Collaborative Filtering* (NCF), which extends matrix factorization by modeling user-item interactions through a multi-layer perceptron to capture non-linear preference patterns [19]. Finally, we include *Matrix Factorization* (MF) as a classical baseline that represents users and items in a shared latent space and estimates preference scores via the inner product of their embeddings [7].

5.1.2 Metrics. Following prior work [8, 32], we evaluate CE methods under implicit feedback settings, where each user is represented by a binary interaction vector $\mathbf{x}_u \in \{0, 1\}^V$. Recommendations are produced as ranked lists based on predicted utility scores $f_\theta(\mathbf{x}_u)[i]$ for each item i .

In our setting, CEs are generated with the explicit objective of altering the *Top-1* recommendation. Evaluation metrics therefore measure how effectively an explainer identifies minimal historical interactions whose removal pushes the originally recommended item down the ranking. The parameter K in the following metrics does *not* define the optimization target, but controls how far the original Top-1 item is displaced in the ranked list.

We adopt stepwise perturbation tests with three approaches that illustrated in Figure 5. In *positive perturbation* tests, interactions

are removed in descending order of importance. Let $\mathbf{x}_u^{\text{pos}}(m)$ denote the user history after m positive perturbation steps, and let $\text{rank}(\mathbf{x}_u^{\text{pos}}(m))$ denote the rank of the originally Top-1 item under the perturbed profile.

Positive Perturbations@K (POS@K) measures how quickly the original Top-1 recommendation falls beyond Top- K positions:

$$\text{POS@K} = \mathbb{I}[\text{rank}(\mathbf{x}_u^{\text{pos}}) \leq K]. \quad (9)$$

for each step m , with $\mathbb{I}$ being the indicator function [8].

NDCG Perturbations (NDCG-P) provides a rank-sensitive variant that captures how quickly the original Top-1 item moves down the ranked list during positive perturbations:

$$\text{NDCG-P}(m) = \frac{1}{\log_2(1 + \text{rank}(\mathbf{x}_u^{\text{pos}}(m)))}. \quad (10)$$

This metric assigns higher weight to early rank drops, offering a more fine-grained view of counterfactual effectiveness. For both metrics AUC values are calculated where lower value correspond to more faithful and efficient CEs. Results are computed per user and reported as averages over the test set [8].

5.2 Traditional Binned Evaluation (RQ1)

We begin by evaluating all methods under the widely adopted *binned evaluation* protocol [8], where user interactions are removed in coarse percentage increments (typically 10%) until the recommendation outcome changes (Figure 5(a)). This protocol remains the de facto standard for benchmarking CE methods and is therefore included to ensure comparability with prior work. The left block of Table 1 summarizes performance under this setting. Figure 5 reports step-wise metric values, while Table 1 reports AUC over the full curve. Therefore, values are not directly comparable. Across all datasets and metrics, DICERec achieves the lowest values. Notably, DICERec improves upon the strongest competing baseline by an average of 10% in POS@K and 22% in NDCG-P, indicating its overall effectiveness even under coarse evaluation. However, the binned protocol fundamentally limits the interpretability of results. Since all perturbations within a bin are aggregated, the protocol assigns identical scores to explanations that succeed after removing a single interaction and those that require removing up to 10% of the user history. As a result, it fails to discriminate between explanations that are *precise and minimal* and those that are *coarse but eventually effective*. This limitation is particularly problematic for CEs, whose core objective is to identify the smallest and most plausible change that alters the recommendation. Moreover, large, percentage-based removals substantially distort the user profile and can easily push the recommender into out-of-distribution regions. In such cases, a ranking change may reflect global profile corruption rather than the faithful identification of truly influential interactions. Consequently, while DICERec’s strong performance under this protocol establishes a reliable baseline, binned evaluation alone is insufficient to assess explanation faithfulness or practical plausibility. For this reason, we next turn to fine-grained and constrained evaluation protocols that explicitly expose how quickly and how efficiently different methods induce counterfactual ranking changes.

5.3 Fine-grained Performance Analysis (RQ2)

We now evaluate CE quality under protocols that more closely reflect the intended use of CE: inducing a change in the *Top-1* recommendation through minimal and plausible modifications to the user interaction history. Unlike binned evaluation, these protocols expose how precisely an explainer identifies the most influential interactions and how robust it remains under realistic perturbation budgets.

Fine-grained perturbation. In the fine-grained protocol, interactions are removed one at a time according to their explanation scores (Figure 5(b)). This setting enables a stepwise analysis of ranking sensitivity and directly measures the quality of the learned importance ordering. Methods that correctly prioritize the most critical interactions cause rapid degradation of ranking metrics in the early steps, whereas less precise methods require multiple removals before any substantial effect is observed. Figure 3(a) illustrates performance trends on ML-1M under this protocol. DICERec consistently exhibits the steepest early decline across POS@3, POS@5, and NDCG-P, indicating that it identifies highly influential interactions at the very top of the ranking. Importantly, the performance gap between DICERec and all baselines—including LXR—emerges within the first few removal steps and persists throughout the trajectory. This pattern demonstrates that DICERec’s advantage is not driven by cumulative perturbations, but by superior precision in ranking the most impactful interactions. In contrast, several baselines show comparatively flat curves in the early steps, followed by delayed degradation. A similar behavior is observed on the Yahoo! Music dataset (see Figure 4), although the extent and onset of this degradation vary due to dataset-specific characteristics, which we analyze in Section 5.4.

Constrained & fine-grained perturbation. While the fine-grained protocol isolates precision, it does not explicitly constrain the extent of profile modification. To assess plausibility under realistic conditions, we further restrict the maximum number of removable interactions to 20% of the user history (Figure 5(c)). This constrained setting reflects practical counterfactual scenarios in which explanations must succeed through limited changes. Under this protocol, performance differences become markedly more pronounced (Figure 3(b)). DICERec maintains strong and consistent degradation across all metrics, whereas the performance of most baselines deteriorates substantially. In particular, methods such as LIME-RS, SHAP, and ACCENT often fail to displace the original Top-1 item within the allowed budget, resulting in high AUC values despite reasonable performance under unconstrained settings. Notably, the margin of improvement achieved by DICERec over LXR and other baselines is larger in the constrained protocol than in the binned and unconstrained fine-grained settings (See Table 1). This widening gap provides direct evidence that DICERec’s explanations are not only precise, but also robust to realistic perturbation limits—an essential property for faithful counterfactual reasoning. Due to space constraints, we are unable to include all results in the paper; however, the complete set of experimental results is available in the anonymous GitHub repository referenced in the introduction.

Table 1: Counterfactual explanation (CE) performance under a VAE-based recommender. Best results are shown in bold, and second-best results are underlined; lower values indicate better performance. For comparability with the baseline protocol, results under the *Fine-grained* and *Constrained & Fine-grained* protocols are reported as averages across all removal steps. Improv. (%) denotes the relative reduction of DICERec compared with the second-best method. p -values are computed using paired t -tests against the second-best method.

Dataset	Method	Evaluation Protocol														
		Baseline Protocol					Fine-grained Protocol					Constrained & Fine-grained Protocol				
		POS@1	POS@3	POS@5	POS@10	NDCG-P	POS@1	POS@3	POS@5	POS@10	NDCG-P	POS@1	POS@3	POS@5	POS@10	NDCG-P
ML-1M	Jaccard	0.22	0.35	0.42	0.51	0.44	0.10	0.24	0.32	0.44	0.38	0.17	0.41	0.54	<u>0.67</u>	0.48
	Cosine	0.22	0.34	0.41	0.50	0.43	0.09	0.23	0.30	0.43	0.34	<u>0.15</u>	<u>0.40</u>	<u>0.52</u>	0.71	<u>0.47</u>
	LIME-RS	0.28	0.42	0.50	0.60	0.50	0.19	0.37	0.49	0.61	0.46	0.40	0.62	<u>0.79</u>	0.89	0.66
	SHAP	0.39	0.57	0.65	0.73	0.60	0.34	0.57	0.68	0.80	0.60	0.60	0.84	0.91	0.96	0.80
	ACCENT	0.31	0.42	0.48	0.56	0.50	0.21	0.36	0.45	0.50	0.45	0.37	0.57	0.70	0.84	0.62
	LXR	<u>0.19</u>	<u>0.29</u>	<u>0.35</u>	<u>0.43</u>	<u>0.39</u>	<u>0.08</u>	<u>0.16</u>	<u>0.22</u>	<u>0.33</u>	<u>0.30</u>	0.27	0.46	0.62	0.73	0.55
	DICERec	0.16	0.26	0.32	0.42	0.27	0.05	0.14	0.20	0.30	0.16	0.11	0.27	0.38	0.56	0.31
	Improv. (%)	18.75	11.53	9.37	2.38	44.44	60	14.28	10	10	87.50	36.36	48.14	36.84	19.64	51.61
	p -value	$8.7e^{-10}$	$1.1e^{-7}$	$1.5e^{-4}$	$5.3e^{-2}$	$2.9e^{-84}$	$1.0e^{-10}$	$1.8e^{-7}$	$8.0e^{-5}$	$6.5e^{-2}$	$2.1e^{-86}$	$7.6e^{-10}$	$9.1e^{-8}$	$6.4e^{-6}$	$1.7e^{-2}$	$4.7e^{-77}$
Pinterest	Jaccard	0.18	0.27	0.32	0.39	0.37	0.10	0.22	0.31	0.44	0.35	0.31	0.59	0.72	0.86	0.60
	Cosine	<u>0.18</u>	<u>0.27</u>	<u>0.32</u>	<u>0.39</u>	<u>0.37</u>	<u>0.10</u>	<u>0.22</u>	<u>0.31</u>	<u>0.44</u>	<u>0.35</u>	<u>0.31</u>	<u>0.59</u>	<u>0.72</u>	<u>0.86</u>	<u>0.60</u>
	LIME-RS	0.21	0.33	0.39	0.47	0.41	0.15	0.32	0.43	0.58	0.42	0.40	0.68	0.80	0.91	0.67
	SHAP	0.30	0.45	0.51	0.60	0.50	0.30	0.55	0.64	0.77	0.57	0.54	0.81	0.90	0.96	0.76
	ACCENT	0.27	0.37	0.41	0.48	0.45	0.23	0.39	0.47	0.59	0.47	0.27	0.37	0.41	0.48	0.45
	LXR	0.25	0.31	0.34	0.42	0.44	0.21	0.40	0.52	0.66	0.48	0.48	0.75	0.84	0.93	0.72
	DICERec	0.15	0.24	0.29	0.37	0.25	0.07	0.17	0.25	0.39	0.21	0.26	0.56	0.70	0.85	0.54
	Improv. (%)	20	12.50	10.34	5.40	48	42.85	29.41	24	12.82	66.66	19.23	5.35	2.85	1.17	11.11
	p -value	$1.2e^{-107}$	$6.9e^{-123}$	$1.3e^{-134}$	$1.0e^{-140}$	$0.0e^{+0}$	$7.0e^{-104}$	$4.2e^{-118}$	$3.7e^{-131}$	$2.1e^{-142}$	$0.0e^{+0}$	$2.6e^{-114}$	$1.1e^{-121}$	$1.4e^{-128}$	$1.9e^{-139}$	$0.0e^{+0}$
Yahoo! Music	Jaccard	0.21	0.34	0.41	0.52	0.44	<u>0.07</u>	0.20	0.29	0.44	0.34	<u>0.33</u>	<u>0.64</u>	0.79	<u>0.92</u>	0.63
	Cosine	<u>0.20</u>	0.33	0.40	0.50	0.46	<u>0.07</u>	<u>0.17</u>	<u>0.25</u>	0.40	<u>0.32</u>	<u>0.33</u>	0.65	0.80	<u>0.92</u>	0.64
	LIME-RS	0.33	0.50	0.57	0.66	0.55	0.27	0.51	0.61	0.72	0.54	0.60	0.91	0.96	0.99	0.81
	SHAP	0.35	0.52	0.60	0.68	0.57	0.34	0.55	0.66	0.77	0.59	0.68	0.93	0.97	0.99	0.85
	ACCENT	0.36	0.48	0.54	0.62	0.55	0.31	0.48	0.56	0.65	0.54	0.76	0.93	0.96	0.99	0.88
	LXR	0.21	<u>0.32</u>	<u>0.39</u>	<u>0.48</u>	<u>0.42</u>	0.08	0.19	0.26	<u>0.38</u>	0.33	0.42	0.74	0.87	0.96	0.70
	DICERec	0.17	0.29	0.35	0.46	0.30	0.05	0.13	0.19	0.32	0.19	0.23	0.52	0.68	0.88	0.53
	Improv. (%)	17.64	10.34	11.42	4.34	40	40	30.76	31.57	18.75	68.42	43.47	23.07	16.17	4.54	15.87
	p -value	$8.7e^{-10}$	$1.5e^{-4}$	$5.3e^{-4}$	$4.7e^{-2}$	$2.9e^{-84}$	$4.6e^{-16}$	$2.2e^{-13}$	$3.9e^{-6}$	$9.0e^{-3}$	$3.3e^{-158}$	$2.6e^{-17}$	$7.2e^{-9}$	$3.0e^{-6}$	$1.7e^{-2}$	$1.4e^{-160}$

5.4 Dataset-Specific Performance Patterns (RQ3)

We observe distinct performance patterns across datasets reflecting differences in interaction density, profile length, and recommendation sensitivity. These variations reveal how CE methods behave under different structural conditions, especially under fine-grained and constrained perturbation protocols.

On MovieLens-1M, which contains relatively dense user histories with rich interaction signals, DICERec exhibits the largest performance gains over baseline methods. In this setting, the fine-grained evaluation protocol reveals that DICERec consistently identifies a small number of highly influential interactions whose removal rapidly displaces the original Top-1 recommendation. This behavior suggests that richer interaction histories provide sufficient structure for diffusion-based modeling to capture subtle but impactful dependencies between user preferences and recommendation outcomes. In contrast, the Pinterest dataset is characterized by shorter user histories and higher sparsity. Under these conditions, the performance gap between methods narrows, particularly in unconstrained settings. Nevertheless, DICERec maintains a consistent advantage in constrained perturbation scenarios, indicating that its effectiveness does not rely solely on dense interaction patterns. Instead, its ability

to operate within limited perturbation budgets appears robust even when fewer interactions are available for explanation.

The Yahoo! Music dataset presents an intermediate case, with moderate interaction density and notable ranking sensitivity. Here, DICERec achieves steady improvements across all evaluation protocols while avoiding the sharp performance degradation observed in several baselines under constrained conditions. This highlights the importance of evaluation protocols that explicitly limit perturbation size, as differences between explanation methods become more pronounced when only realistic counterfactual changes are permitted. Overall, these dataset-specific trends reinforce the conclusion that explanation quality cannot be fully assessed without considering both interaction structure and evaluation granularity. Fine-grained, constrained protocols reveal behavioral differences that remain hidden under coarse evaluation schemes, particularly in scenarios where only minimal changes are meaningful for counterfactual reasoning.

5.5 Computational Efficiency Analysis (RQ4)

Table 2 reports the mean per-instance running time for training and inference across three datasets. DICERec introduces an additional offline training stage to learn a diffusion-based prior over latent

user representations; this cost is incurred once and amortized over all explanation queries. As shown in the table, DICERec’s training time is comparable to neural baselines such as LXR and ACCENT, and lower than several perturbation-based methods that require repeated surrogate fitting. DICERec incurs higher inference latency than lightweight similarity-based methods due to iterative reverse diffusion steps. However, its inference time remains consistently lower than ACCENT and LIME, and is on the same order of magnitude as LXR across datasets. This reflects a deliberate trade-off between computational cost and the ability to generate distribution-consistent and compact CEs. The inference cost of DICERec scales linearly with the number of diffusion steps, while decoding and score evaluation introduce a shared linear dependence on the item set size. In practice, a moderate number of diffusion steps suffices to produce valid counterfactuals, keeping runtime within practical bounds.

Table 2: Mean runtime per instance (ms). Training cost is amortized over all explanations.

Method		ML-1M		Pinterest		Yahoo! Music	
		Train	Inf.	Train	Inf.	Train	Inf.
Non-Neural	Jaccard	585	30	170	54	569	32
	Cosine	288	24	931	46	322	25
	SHAP	228	18	590	15	270	14
	LIME	129	2431	451	5615	124	3243
Neural	LXR	238	88	320	94	360	110
	ACCENT	280	4374	730	3593	268	2358
	DICERec	94	343	127	272	77	308

6 Limitations

While DiceRec demonstrates strong performance in generating minimal and plausible counterfactual explanations, several limitations should be acknowledged. First, our approach relies on a pre-trained autoencoder to learn a latent representation of user interaction histories. This introduces an implicit assumption that the learned latent space adequately captures the data distribution relevant to the target recommender. If the recommender’s behavior is not well aligned with this distribution, the generated counterfactuals may be less faithful. Second, although DiceRec is model-agnostic in the sense that it does not assume a specific recommender architecture, it requires access to gradient information with respect to the target item’s score. This restricts applicability to settings where such information is available (i.e., white-box or differentiable models). Third, the guidance mechanism operates primarily through the gradient of a single target item’s score. While this enables efficient counterfactual generation, it may not fully capture complex listwise interactions in ranking-based recommendation scenarios. Fourth, the diffusion-based generation process introduces additional computational overhead compared to simpler perturbation-based methods, particularly when using a large number of diffusion steps. Finally, our evaluation focuses on latent factor and neural collaborative filtering models. Extending the approach to structurally different recommender architectures, such as sequential or graph-based models, remains an important direction for future work.

7 Conclusion and Future Work

We proposed DICERec, a diffusion-based framework for generating counterfactual explanations under minimal and plausible perturbations. By injecting counterfactual guidance into diffusion, DICERec produces compact, coherent explanations that reliably flip Top-1 recommendations without relying on heuristic perturbation budgets or explicit sparsity regularization. Our experiments show that DICERec consistently outperforms state-of-the-art baselines, particularly under fine-grained and constrained evaluation protocols that reflect realistic counterfactual conditions. These findings underscore the importance of aligning explanation methods and evaluation strategies with minimal-perturbation objectives.

DICERec incurs additional computational cost due to the iterative diffusion-based generation at inference time. While this overhead is practical in our experiments, it may require optimization for large-scale or latency-sensitive settings. Our current guidance formulation targets a Top-1 recommendation; extending it to multiple target items or ranked lists is a natural next step. Although DICERec is post-hoc and model-agnostic, it assumes access to latent representations and gradient information, which may not be available in strictly black-box scenarios.

Future work further includes incorporating side information for richer explanations, exploring adaptive or multi-objective guidance strategies, and developing metrics that jointly assess recommendation quality and explanation faithfulness.

References

- [1] Behnoush Abdollahi and Olfa Nasraoui. 2016. Explainable Matrix Factorization for Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Companion Volume*. ACM, 5–6. doi:10.1145/2872518.2889405
- [2] Behnoush Abdollahi and Olfa Nasraoui. 2017. Using explainability for constrained matrix factorization. In *Proceedings of the eleventh ACM conference on recommender systems*. 79–83.
- [3] Dariusz Afchar, Alessandro B. Melchiorre, Markus Schedl, Romain Hennequin, Elena V. Epure, and Manuel Moussallam. 2022. Explainability in Music Recommender Systems. *CoRR* abs/2201.10528 (2022). arXiv:2201.10528 <https://arxiv.org/abs/2201.10528>
- [4] Saugat Aryal and Mark T. Keane. 2024. Even-Ifs from If-Onlys: Are the Best Semi-factual Explanations Found Using Counterfactuals as Guides?. In *Case-Based Reasoning Research and Development - ICCBR 2024, Merida, Mexico, July 1-4, 2024, Proceedings*, Vol. 14775. Springer, 33–49. doi:10.1007/978-3-031-63646-2_3
- [5] Sujoy Bag, Sri Krishna Kumar, and Manoj Kumar Tiwari. 2019. An efficient recommendation generation using relevant Jaccard similarity. *Inf. Sci.* 483 (2019), 53–64.
- [6] Mikhail Baklanov, Veronika Bogina, Yehonatan Elisha, Yahlly Schein, Liron I. Allerhand, Oren Barkan, and Noam Koenigstein. 2025. Refining Fidelity Metrics for Explainable Recommendations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, Nicola Ferro, Maria Maistro, Gabriella Pasi, Omar Alonso, Andrew Trotman, and Suzan Verberne (Eds.). ACM, 2967–2971. doi:10.1145/3726302.3730242
- [7] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. 2011. Matrix factorization techniques for context aware recommendation. In *Proceedings of the fifth ACM conference on Recommender systems*. 301–304.
- [8] Oren Barkan, Veronika Bogina, Liya Gurevitch, Yuval Asher, and Noam Koenigstein. 2024. A Counterfactual Framework for Learning and Evaluating Explanations for Recommender Systems. In *WWW '24*. ACM, 3723–3733.
- [9] Christine Bauer, Eva Zangerle, and Alan Said. 2024. Exploring the Landscape of Recommender Systems Evaluation: Practices and Perspectives. *Trans. Recomm. Syst.* 2, 1 (2024), 11:1–11:31. doi:10.1145/3629170
- [10] Mustafa Bilgic and Raymond J Mooney. 2005. Explaining recommendations: Satisfaction vs. promotion. In *Beyond personalization workshop, IUI*, Vol. 5. 153.
- [11] Tong Chen, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang. 2020. Try this instead: Personalized and interpretable substitute recommendation. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 891–900.

- [12] Weiyu Cheng, Yanyan Shen, Linpeng Huang, and Yanmin Zhu. 2019. Incorporating Interpretability into Latent Factor Models via Fast Influence Analysis. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) (KDD '19). Association for Computing Machinery, New York, NY, USA, 885–893. doi:10.1145/3292500.3330857
- [13] Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer. 2012. The Yahoo! Music Dataset and KDD-Cup '11. In *Proceedings of KDD Cup 2011 competition, San Diego, CA, USA, 2011 (JMLR Proceedings, Vol. 18)*. JMLR.org, 8–18. <http://proceedings.mlr.press/v18/dror12a.html>
- [14] Timo Freiesleben. 2022. The Intriguing Relation Between Counterfactual Explanations and Adversarial Examples. *Minds Mach.* 32, 1 (2022), 77–109. doi:10.1007/s11023-021-09580-9
- [15] Keren Gaiger, Oren Barkan, Shir Tsiptory-Samuel, and Noam Koenigstein. 2023. Not all memories created equal: Dynamic user representations for collaborative filtering. *Ieee Access* 11 (2023), 34746–34763.
- [16] Muskan Gupta, Priyanka Gupta, and Lovekesh Vig. 2024. Guided Diffusion-based Counterfactual Augmentation for Robust Session-based Recommendation. *CoRR abs/2410.21892* (2024). arXiv:2410.21892 doi:10.48550/ARXIV.2410.21892
- [17] Sergiu Hart and Andreu Mas-Colell. 1989. Potential, value, and consistency. *Econometrica: Journal of the Econometric Society* (1989), 589–614.
- [18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 173–182. doi:10.1145/3038912.3052569
- [19] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
- [20] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. 2000. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. 241–250.
- [21] Zhenhao Jiang and Jicong Fan. 2024. DiffRec: Generative Fair Recommender with Conditional Diffusion Model. In *IEEE International Conference on Data Mining, ICDM 2024, Abu Dhabi, United Arab Emirates, December 9-12, 2024*, Elena Baralis, Kun Zhang, Ernesto Damiani, Mérouane Debbah, Panos Kalnis, and Xindong Wu (Eds.). IEEE, 717–722. doi:10.1109/ICDM59182.2024.00082
- [22] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1312.6114>
- [23] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.
- [24] Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. *arXiv preprint arXiv:2105.11601* (2021).
- [25] Yicong Li, Xiangguo Sun, Hongxu Chen, Sixiao Zhang, Yu Yang, and Guandong Xu. 2024. Attention Is Not the Only Choice: Counterfactual Reasoning for Path-Based Explainable Recommendation. *IEEE Trans. Knowl. Data Eng.* 36, 9 (2024), 4458–4471. doi:10.1109/TKDE.2024.3373608
- [26] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 WWW 2018, Lyon, France, April 23-27, 2018*. ACM, 689–698. doi:10.1145/3178876.3186150
- [27] Jianghao Lin, Jiaqi Liu, Jiachen Zhu, Yunjia Xi, Chengkai Liu, Yangtian Zhang, Yong Yu, and Weinan Zhang. 2024. A Survey on Diffusion Models for Recommender Systems. *CoRR abs/2409.05033* (2024). arXiv:2409.05033 doi:10.48550/ARXIV.2409.05033
- [28] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 4765–4774. <https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html>
- [29] Nishtha Madaan and Srikanta Bedathur. 2024. Navigating the Structured What-If Spaces: Counterfactual Generation via Structured Diffusion. In *IEEE Conference on Secure and Trustworthy Machine Learning, SaTML 2024, Toronto, ON, Canada, April 9-11, 2024*. IEEE, 710–722. doi:10.1109/SATML59370.2024.00041
- [30] Alessandro B. Melchiorre, Navid Rekasaz, Christian Ganhör, and Markus Schedl. 2022. ProtoMF: Prototype-based Matrix Factorization for Effective and Explainable Recommendations. In *Proceedings of the 16th ACM Conference on Recommender Systems* (Seattle, WA, USA) (RecSys '22). Association for Computing Machinery, New York, NY, USA, 246–256. doi:10.1145/3523227.3546756
- [31] Amir Reza Mohammadi. 2023. Explainable Graph Neural Network Recommenders: Challenges and Opportunities. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song (Eds.). ACM, 1318–1324. doi:10.1145/3604915.3608875
- [32] Amir Reza Mohammadi, Andreas Peintner, Michael Müller, and Eva Zangerle. 2024. Are We Explaining the Same Recommenders? Incorporating Recommender Performance for Evaluating Explainers. In *RecSys '24*. ACM, 1113–1118.
- [33] Amir Reza Mohammadi, Andreas Peintner, Michael Müller, and Eva Zangerle. 2025. Beyond Top-1: Addressing Inconsistencies in Evaluating Counterfactual Explanations for Recommender Systems. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems, RecSys 2025, Prague, Czech Republic, September 22-26, 2025*, Mária Bielíková, Pavel Kordík, Markus Schedl, Marco de Gemmis, Sole Pera, Rodrigo Alves, Olivier Jeunen, and Vito Ostuni (Eds.). ACM, 515–520. doi:10.1145/3705328.3748028
- [34] Caio Nóbrega and Leandro Balby Marinho. 2019. Towards explaining recommendations through local surrogate models. In *ACM/SIGAPP '19*. ACM, 1671–1678.
- [35] Andreas Peintner, Amir Reza Mohammadi, and Eva Zangerle. 2025. Efficient Session-based Recommendation with Contrastive Graph-based Shortest Path Search. *Trans. Recomm. Syst.* 3, 4 (2025), 46:1–46:24. doi:10.1145/3701764
- [36] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. ACM, 1135–1144. doi:10.1145/2939672.2939778
- [37] Simon Schrodli, Karim Farid, Max Argus, and Thomas Brox. 2024. Latent Diffusion Counterfactual Explanations. In *Pattern Recognition - 46th DAGM German Conference, DAGM GCPR 2024, Munich, Germany, September 10-13, 2024, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 15297)*, Daniel Cremers, Zorah Löhner, Michael Moeller, Matthias Nießner, Björn Ommer, and Rudolph Triebel (Eds.). Springer, 295–311. doi:10.1007/978-3-031-85181-0_19
- [38] Javier Del Ser, Alejandro Barredo Arrieta, Natalia Díaz Rodríguez, Francisco Herrera, Anna Saranti, and Andreas Holzinger. 2024. On generating trustworthy counterfactual explanations. *Inf. Sci.* 655 (2024), 119898. doi:10.1016/j.ins.2023.119898
- [39] Ramni Harbir Singh, Sargam Maurya, Tanisha Tripathi, Tushar Narula, and Gaurav Srivastav. 2020. Movie recommendation system using cosine similarity and KNN. *International Journal of Engineering and Advanced Technology* 9, 5 (2020), 556–559.
- [40] Khanh Hiep Tran, Azin Ghazimatin, and Rishiraj Saha Roy. 2021. Counterfactual Explanations for Neural Recommenders. In *SIGIR '21*. ACM, 1627–1631.
- [41] Sahil Verma, Ashudeep Singh, Varich Boonsanong, John P. Dickerson, and Chirag Shah. 2023. RecRec: Algorithmic Recourse for Recommender Systems. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, Ingo Frommholz, Frank Hopfgartner, Mark Lee, Michael Oakes, Mounia Lalmas, Min Zhang, and Rodrygo L. T. Santos (Eds.). ACM, 4325–4329. doi:10.1145/3583780.3615181
- [42] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *CoRR abs/1711.00399* (2017). arXiv:1711.00399 <http://arxiv.org/abs/1711.00399>
- [43] Xiangmeng Wang, Qian Li, Dianer Yu, Qing Li, and Guandong Xu. 2024. Counterfactual Explanation for Fairness in Recommendation. *ACM Trans. Inf. Syst.* 42, 4 (2024), 106:1–106:30. doi:10.1145/3643670
- [44] Xiangmeng Wang, Qian Li, Dianer Yu, Qing Li, and Guandong Xu. 2024. Reinforced Path Reasoning for Counterfactual Explainable Recommendation. *IEEE Trans. Knowl. Data Eng.* 36, 7 (2024), 3443–3459. doi:10.1109/TKDE.2024.3354077
- [45] Ting-Ruen Wei and Yi Fang. 2025. Diffusion Models in Recommendation Systems: A Survey. *CoRR abs/2501.10548* (2025). arXiv:2501.10548 doi:10.48550/ARXIV.2501.10548
- [46] Dianer Yu, Qian Li, Xiangmeng Wang, Qing Li, and Guandong Xu. 2024. Counterfactual Explainable Conversational Recommendation. *IEEE Trans. Knowl. Data Eng.* 36, 6 (2024), 2388–2400. doi:10.1109/TKDE.2023.3322403
- [47] Yi Yu, Kazunari Sugiyama, and Adam Jatowt. 2025. Domain Counterfactual Data Augmentation for Explainable Recommendation. *ACM Trans. Inf. Syst.* 43, 3, Article 58 (Feb. 2025), 30 pages. doi:10.1145/3711856
- [48] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 83–92.
- [49] Jujia Zhao, Wenjie Wang, Yiyan Xu, Teng Sun, Fuli Feng, and Tat-Seng Chua. 2024. Denoising Diffusion Recommender Model. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccan, and Yi Zhang (Eds.). ACM, 1370–1379. doi:10.1145/3626772.3657825
- [50] Jinfeng Zhong and Elsa Negre. 2022. Shap-enhanced counterfactual explanations for recommendations. In *SAC '22*. ACM, 1365–1372.