

# Fidelity-Aware Recommendation Explanations via Stochastic Path Integration

Oren Barkan<sup>1\*</sup>, Yahly Schein<sup>2\*</sup>, Yehonatan Elisha<sup>2</sup>,  
Veronika Bogina<sup>2</sup>, Mikhail Baklanov<sup>2</sup>, Noam Koenigstein<sup>2†</sup>

<sup>1</sup>The Open University, Israel

<sup>2</sup>Tel Aviv University, Israel

## Abstract

Explanation fidelity, which measures how accurately an explanation reflects a model’s true reasoning, remains critically underexplored in recommender systems. We introduce SPINRec (Stochastic Path Integration for Neural Recommender Explanations), a model-agnostic approach that adapts path-integration techniques to the sparse and implicit nature of recommendation data. To overcome the limitations of prior methods, SPINRec employs stochastic baseline sampling: instead of integrating from a fixed or unrealistic baseline, it samples multiple plausible user profiles from the empirical data distribution and selects the most faithful attribution path. This design captures the influence of both observed and unobserved interactions, yielding more stable and personalized explanations. We conduct the most comprehensive fidelity evaluation to date across three models (MF, VAE, NCF), three datasets (ML1M, Yahoo! Music, Pinterest), and a suite of counterfactual metrics, including AUC-based perturbation curves and fixed-length diagnostics. SPINRec consistently outperforms all baselines, establishing a new benchmark for faithful explainability in recommendation.

**Code** — <https://github.com/DeltaLabTLV/SPINRec>

## Introduction

Recent advances in recommender systems over the past decade (He et al. 2017; Kang and McAuley 2018; He et al. 2020; Barkan et al. 2019; Barkan, Katz, and Koenigstein 2020; Barkan et al. 2021; Katz et al. 2022) have increasingly shaped personalized decisions across e-commerce, social media, and streaming platforms, making transparency and trust more essential than ever. (Fan et al. 2022). Explainability in these systems is critical not only for user satisfaction but also for accountability, compliance with regulations, and user control. However, while explainable recommendation research is rapidly expanding (Zhang, Chen et al. 2020; Varasteh et al. 2024), most existing work focuses on user-centric aspects such as persuasiveness, clarity, or satisfaction (Kunkel et al. 2019; Tintarev 2025). A critical yet underexplored dimension is *fidelity*, which measures how accurately explanations

reflect a recommender’s actual decision process. Without fidelity, explanations may appear plausible while failing to reveal the true reasoning behind recommendations (Koenigstein 2025).

We introduce SPINRec (Stochastic Path Integration for Neural Recommender Explanations), the first adaptation of path-integration (PI) (Sundararajan, Taly, and Yan 2017) to recommender systems. Unlike prior applications of PI in vision (Kapishnikov et al. 2021; Barkan et al. 2023a,b, 2025) or NLP (Sikdar, Bhattacharya, and Heese 2021; Enguehard 2023), recommender data is characterized by extreme sparsity and binary-valued interactions, where the absence of an interaction may be ambiguous. Standard PI methods, which integrate gradients from an all-zero baseline, fail in this setting due to weak or misleading attribution signals. Crucially, modern recommenders leverage both observed and unobserved interactions as informative signals. SPINRec addresses this by stochastically sampling plausible user baselines from the empirical data distribution and selecting the explanation that maximizes fidelity. This adaptation enables more stable and faithful explanations tailored to the structure of recommender systems.

To evaluate SPINRec, we conducted extensive fidelity evaluation spanning three model architectures (MF, VAE, NCF), multiple benchmark datasets (ML1M, Yahoo! Music, Pinterest), and a suite of counterfactual fidelity metrics (Barkan et al. 2024; Gurevitch et al. 2025; Baklanov et al. 2025). Our results establish SPINRec as the new state-of-the-art benchmark in recommender systems explainability, with ablation studies confirming the distinct contributions of both path-integration and our stochastic baseline sampling strategy.

## Contributions:

- Introduce SPINRec, the first adaptation of path-integration methods to recommender systems.
- Develop a novel stochastic baseline sampling strategy tailored to sparse, binary recommendation data.
- Conduct comprehensive fidelity-focused evaluation across multiple architectures and datasets.
- Establish SPINRec as the new state-of-the-art for fidelity-aware recommendation explanations.

As fidelity remains an underexplored yet critical dimension in explainable recommendation (Baklanov et al. 2025; Mohammedi et al. 2025; Koenigstein 2025), we expect this work to

\*Equal contribution.

†Corresponding author.

lay essential groundwork for future research on trustworthy, model-faithful explanations.

## Related Work

The rapid growth of recommender systems has driven increasing interest in Explainable AI (XAI) methods to ensure transparency, build trust, and enhance user engagement (Tintarev and Masthoff 2022; Zhang, Chen et al. 2020). While a broad range of explanations for recommenders exist, standardized benchmarks for explanation fidelity remain significantly underexplored (Baklanov et al. 2025; Mohammadi et al. 2025; Koenigstein 2025).

## Explanation Methods for Recommenders

Early works typically proposed model-specific explanations, such as those for matrix factorization (Abdollahi and Nasraoui 2016, 2017) or inherently interpretable recommender architectures (Barkan et al. 2020, 2023c; Melchiorre et al. 2022; Gaiger et al. 2023; Sugahara and Okamoto 2024). Aspect-based methods attribute recommendations to human-interpretable item features (e.g., price or color) (Vig, Sen, and Riedl 2009; Zhang et al. 2014; Wang et al. 2018; Li, Zhang, and Chen 2021). However, these methods rely heavily on structured feature availability and are difficult to generalize to implicit or sparse data scenarios.

Model-agnostic explanation methods offer broader applicability by explaining arbitrary recommenders independently of their internal mechanisms. Prominent examples include LIME-RS (Nóbrega and Marinho 2019), influence-based methods such as FIA and ACCENT (Cheng et al. 2019; Tran, Ghazimatin, and Saha Roy 2021), and Shapley-value-based methods (SHAP4Rec and DeepSHAP) (Zhong and Negre 2022; Lundberg and Lee 2017a). While model-agnostic methods enhance generality, their fidelity remains less scrutinized and insufficiently benchmarked.

## Fidelity Evaluation in Recommender Systems

Explanation fidelity, the degree to which an explanation reflects the true reasoning of a recommender model, is essential for transparency and accountability, yet remains underexplored in the recommendation domain. Unlike computer vision or NLP, where fidelity evaluation is well-established (Samek et al. 2016; Agarwal et al. 2022), recommender systems have historically focused on user-centric goals like persuasiveness (Tintarev and Masthoff 2015; Zhang, Chen et al. 2020) or satisfaction, often overlooking whether explanations faithfully reflect model logic (Koenigstein 2025). Recent works have introduced counterfactual frameworks that evaluate fidelity by perturbing user histories and observing changes in recommendation outcomes (Barkan et al. 2024; Gurevitch et al. 2025). However, early approaches conflate supportive and contradictory features, apply coarse fixed-percentage masking, and lack control over explanation conciseness. The refined metrics proposed by Baklanov et al. (2025) address these limitations by evaluating fixed-length explanations, separating feature roles, and enabling consistent, interpretable comparisons across users. Our work builds directly on this line of research, offering the first extensive

empirical evaluation that spans both the original (Barkan et al. 2024) and refined (Baklanov et al. 2025) fidelity metrics. By benchmarking a wide range of explanation methods across multiple datasets and recommender models, we establish SPINRec as a new state-of-the-art for fidelity-aware explanations in recommender systems.

## Path-Integration (PI) for Explainability

Path-integration (PI) techniques (Sundararajan, Taly, and Yan 2017) are widely adopted in computer vision and NLP (Kapishnikov et al. 2021; Xu, Venugopalan, and Sundararajan 2020; Sanyal and Ren 2021; Enguehard 2023) to overcome limitations of vanilla gradients such as saturation and instability. By integrating gradients along a path from a baseline to the input, PI yields more robust and interpretable explanations.

However, directly applying existing PI methods to recommender systems is suboptimal. User representations in this domain are high-dimensional, sparse, and binary, where the absence of interaction conveys ambiguous information. Naïve baselines, such as all-zero “cold user” vectors, fail to reflect realistic user behavior and may produce weak or misleading gradient signals. This issue parallels challenges in computer vision, where black-image baselines distort attribution in dark regions (Haug et al. 2021).

SPINRec introduces a stochastic baseline sampling strategy explicitly tailored to these challenges. Instead of a single unrealistic baseline, it samples multiple plausible user histories from the empirical data distribution, capturing both presence and absence information.

## The SPINRec Algorithm

We introduce SPINRec (Stochastic Path Integration for Neural Recommender Explanations), the first adaptation of path-integration methods to explain recommender systems.

### Setup and Notation

Let  $\mathcal{U}$  and  $\mathcal{V}$  denote the sets of users and items, respectively. Each user  $u \in \mathcal{U}$  is represented by a binary feature vector  $\mathbf{x}_u \in \{0, 1\}^{|\mathcal{V}|}$ , where each feature  $\mathbf{x}_u[i] = 1$  indicates that user  $u$  interacted with item  $i$ , and 0 otherwise.

We consider a recommender model  $f : \{0, 1\}^{|\mathcal{V}|} \rightarrow [0, 1]^{|\mathcal{V}|}$ , parameterized by  $\theta$ , which outputs predicted affinity scores over items given  $\mathbf{x}_u$ . The predicted affinity for item  $y \in \mathcal{V}$  is denoted  $f^y(\mathbf{x}_u)$ .

An explanation algorithm assigns a relevance score to each feature via an explanation map  $\mathbf{m} \in [0, 1]^{|\mathcal{V}|}$ , where  $\mathbf{m}[i]$  quantifies the contribution of feature  $i$  (i.e.,  $\mathbf{x}[i]$ ) to the prediction  $f^y(\mathbf{x}_u)$ . High  $\mathbf{m}[i]$  values indicate stronger influence on the recommendation.

### Path-Integration for Recommenders

Given a user data vector  $\mathbf{x}$  and a recommended item  $y$ , SPINRec attributes the predicted affinity score  $f^y_\theta(\mathbf{x})$  to individual features in the user data vector by integrating gradients along a path from a baseline vector  $\mathbf{z}$  to  $\mathbf{x}$ . We define a straight-line path  $\mathbf{r}(t) = t \cdot \mathbf{x} + (1 - t) \cdot \mathbf{z}$  for  $t \in [0, 1]$ , interpolating between the baseline and actual user representation.

The attribution is defined as the difference in predicted scores between the affinity of item to a user with personal data  $\mathbf{x}$  and a baseline  $\mathbf{z}$ . The difference can be decomposed via the chain rule:

$$\begin{aligned} f_\theta^y(\mathbf{x}) - f_\theta^y(\mathbf{z}) &= \int_0^1 \frac{d}{dt} f_\theta^y(\mathbf{r}(t)) dt = \int_0^1 \mathbf{r}'(t) \cdot \nabla f_\theta^y(\mathbf{r}(t)) dt \\ &= \sum_{i=1}^{|\mathcal{V}|} \int_0^1 \frac{dr_i}{dt} \cdot \frac{\partial f_\theta^y(\mathbf{r}(t))}{\partial r_i} dt, \end{aligned} \quad (1)$$

where  $r_i(t)$  is the  $i$ -th coordinate of  $\mathbf{r}(t)$ . Hence, an explanation map  $\mathbf{m}$  can be calculated by attributing the difference between the predicted scores at  $\mathbf{x}$  and  $\mathbf{z}$  according to:

$$\mathbf{m} = \int_0^1 \frac{\partial f_\theta^y(\mathbf{r}(t))}{\partial \mathbf{r}(t)} \circ \frac{d\mathbf{r}(t)}{dt} dt, \quad (2)$$

where  $\circ$  denotes element-wise multiplication.

While PI is well-studied in continuous domains, applying it effectively in sparse, binary recommender data requires careful baseline design.

### Challenges in Baseline Selection

The choice of the baseline vector  $\mathbf{z}$  significantly impacts the effectiveness of path-integration (PI) methods (Haug et al. 2021; Erion et al. 2021). While baseline sampling is common in other domains e.g., computer vision (Erion et al. 2021), recommender systems pose unique challenges due to the sparse and binary nature of user data:

- **Implicit Binary Signals:** Binary inputs limit the variability and restrict the range of values the model expects. feedback restricts the applicability of continuous-valued baseline methods (Sturmfels, Lundberg, and Lee 2020; Haug et al. 2021).
- **Data Sparsity:** Most user–item interactions are zeros, meaning that no meaningful path exists when integrating from an all-zero baseline.
- **Diverse User Behaviors:** A single baseline may not capture the variability across user preferences and interactions.

Crucially, a naïve baseline (i.e., “cold user”) produces suboptimal gradient signals, since unobserved items remain zero during interpolation and thus contribute no gradients. However, modern recommenders leverage both observed and unobserved interactions as informative signals. This insight motivates our use of non-zero, data-driven baselines: by sampling plausible user profiles, SPINRec captures both presence and absence effects, yielding significantly more faithful explanations, as confirmed by our ablation study.

### Stochastic Baseline Sampling

SPINRec introduces a stochastic sampling strategy tailored to recommender systems’ sparse and binary data. Instead of relying on a single baseline, it samples a set of  $\kappa$  plausible baselines  $\mathcal{B} = \{\mathbf{z}_1, \dots, \mathbf{z}_\kappa\}$  from the distribution of user histories, capturing the diversity and heterogeneity of user behavior. For each  $\mathbf{z}_i \in \mathcal{B}$ , we compute an explanation map  $\mathbf{m}_i$  using Eq. 2. Then, the final explanation  $\mathbf{m}^*$  is chosen to maximize a fidelity metric  $s(\cdot)$  as follows:

$$\mathbf{m}^* = \arg \max_{\mathbf{m} \in \mathcal{M}} s(\mathbf{m}), \quad (3)$$

where  $\mathcal{M} = \{\mathbf{m}_1, \dots, \mathbf{m}_\kappa\}$ . We note that beyond the maps in  $\mathcal{M}$ , one can further consider the mean map  $\bar{\mathbf{m}} = \frac{1}{\kappa} \sum_{i=1}^{\kappa} \mathbf{m}_i$ ; by doing so, SPINRec generalizes Expected Gradients (Erion et al. 2021).

Algorithm 1 summarizes the SPINRec process, from baseline sampling to final explanation map selection.

---

#### Algorithm 1: SPINRec : Stochastic Path-Integration

---

- 1: **Input:** User data  $\mathbf{x}$ , recommender  $f_\theta$ , target item  $y$ , number of baselines to sample  $\kappa$ , metric  $s$
  - 2: **Output:** Explanation map  $\mathbf{m}^*$
  - 3:  $\mathcal{M} \leftarrow \{\}$ ; Sample  $\kappa$  baselines from  $\mathcal{U}$  to form the baselines set  $\mathcal{B}$
  - 4: **for**  $\mathbf{z} \in \mathcal{B}$  **do**
  - 5:   Compute path  $\mathbf{r}(t)$  from  $\mathbf{z}$  to  $\mathbf{x}$
  - 6:   Compute  $\mathbf{m}$  via Eq. 2;  $\mathcal{M} \leftarrow \mathcal{M} \cup \{\mathbf{m}\}$
  - 7: **end for**
  - 8: **return**  $\mathbf{m}^* \leftarrow \underset{\mathbf{m} \in \mathcal{M}}{\operatorname{argmax}} s(\mathbf{m})$
- 

### Computational Complexity

For each of the  $\kappa$  sampled baselines, SPINRec integrates over  $J$  gradient steps, followed by  $N$  perturbation-based evaluations to compute  $s(\cdot)$ . Accordingly, SPINRec’s computational cost is dominated by two components: gradient-based integration (Eq. 2) and the counterfactual evaluation via  $s(\cdot)$  (Eq. 3).

For a model with  $Q$  parameters and  $|\mathcal{V}|$  items, the overall cost is:

$$O(\kappa Q(J + N|\mathcal{V}|)) \approx O(\kappa QN|\mathcal{V}|),$$

since typically  $J \ll N|\mathcal{V}|$ .

Compared to SHAP (Zhong and Negre 2022) with exponential cost in  $|\mathcal{V}|$ , or LIME (Ribeiro, Singh, and Guestrin 2016) with cubic sample complexity, SPINRec scales *linearly* with the number of user features and baseline samples. All steps are embarrassingly parallel and well-suited to GPU acceleration. We note that while LXR (Barkan et al. 2024) offers faster inference via a trained explainer, it requires pre-training and still falls short of SPINRec’s fidelity as we show in our evaluations.

### Counterfactual Fidelity Metrics

As discussed earlier, recent work has introduced counterfactual fidelity metrics tailored to recommender systems (Barkan et al. 2024; Baklanov et al. 2025). We build on this foundation by being the first to systematically evaluate both the original AUC-based metrics from Barkan et al. (2024) and the refined fixed-length variants proposed by Baklanov et al. (2025). To ensure a fair comparison, we adhere strictly to the evaluation protocols used in these works.

**Illustrative Example.** Figure 1 illustrates the principal behind counterfactual fidelity evaluation. Given a user’s interaction history, SPINRec identifies key features driving the recommendation of “*The Lion King*”. Masking these features results in a substantial rank drop, empirically validating their explanatory power.

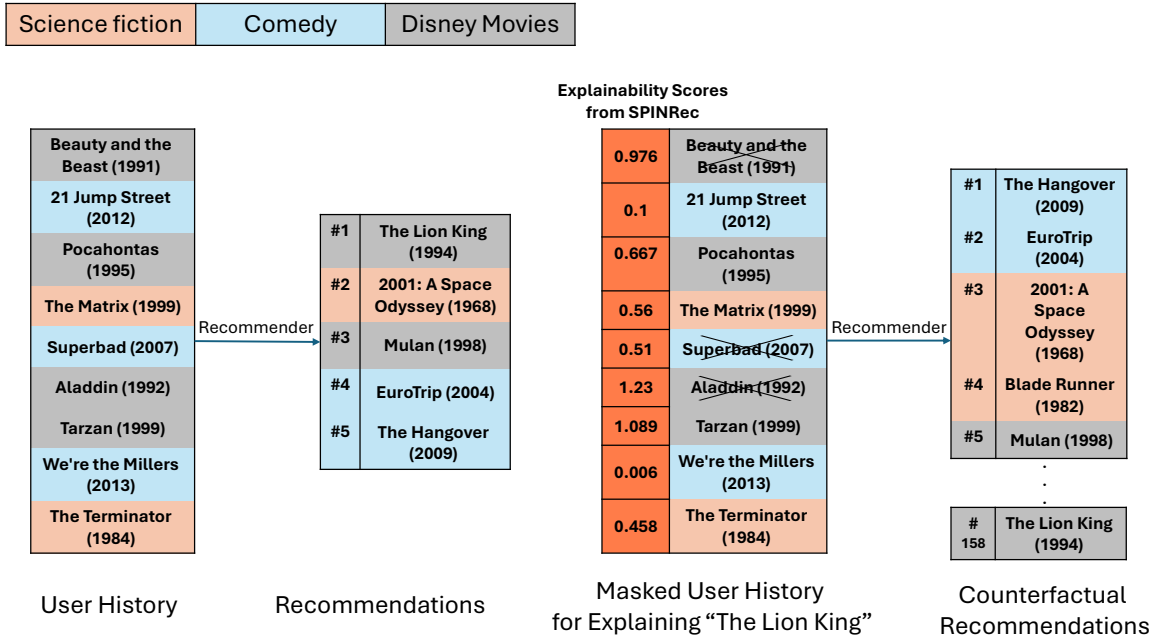


Figure 1: Illustration of Counterfactual Fidelity: SPINRec identifies items in the user’s history most responsible for recommending “*The Lion King*”. When these items are masked, the recommendation’s rank drastically drops, demonstrating explanation fidelity.

**Formal Definitions.** Let  $\mathbf{x}_u$  denote user  $u$ ’s historical interaction vector, and  $K_e$  the number of top explanatory features. Define a binary mask  $\mathbf{m}_{K_e}$  selecting the top  $K_e$  features, and form two perturbed user vectors:

**Retained Explanations Vector:**  $\mathbf{x}_u^{K_e} = \mathbf{x}_u \circ \mathbf{m}_{K_e}$

**Removed Explanations Vector:**  $\mathbf{x}_u^{\setminus K_e} = \mathbf{x}_u \circ (\mathbf{1} - \mathbf{m}_{K_e})$

The following metrics assess fidelity by measuring ranking or confidence changes for the target item  $y$  under these counterfactual modifications:

**POS@ $K_r, K_e$ :** Item  $y$  drops out of top- $K_r$  recommendations when top- $K_e$  features are removed (lower is better):

$$\text{POS}@K_r, K_e = \mathbb{1}[\text{rank}_{f_\theta}^y(\mathbf{x}_u^{\setminus K_e}) \leq K_r].$$

**DEL@ $K_e$ :** Confidence drop after removing top- $K_e$  features (lower is better):

$$\text{DEL}@K_e = \frac{f(\mathbf{x}_u^{\setminus K_e})_y}{f(\mathbf{x}_u)_y}.$$

**INS@ $K_e$ :** Confidence recovery from adding top- $K_e$  features (higher is better):

$$\text{INS}@K_e = \frac{f(\mathbf{x}_u^{K_e})_y}{f(\mathbf{x}_u)_y}.$$

**CDCG@ $K_e$ :** Rank degradation after removing explanatory features (lower is better):

$$\text{CDCG}@K_e = \frac{1}{\log_2(1 + \text{rank}_{f_\theta}^y(\mathbf{x}_u^{\setminus K_e}))}.$$

**AUC Computation.** To compute AUC variants, we follow the fixed-step perturbation strategy of Barkan et al. (2024), which averages the model’s scores as features in the user vector are progressively removed or added.

## Experimental Setup

Our setup builds on the protocol of Barkan et al. (2024), extending it with a third dataset (Pinterest), additional fidelity metrics from Baklanov et al. (2025), and a broader set of explanation baselines. Due to space constraints, the Pinterest results are provided in our public repository. Hyperparameters were tuned via grid search on a held-out validation set, with final values also available in the repository. All experiments were conducted on NVIDIA V100 GPUs using PyTorch 1.13 and CUDA 11.7.

## Recommendation Models

We evaluate SPINRec across three standard recommendation models:

**Matrix Factorization (MF)** (Koren, Bell, and Volinsky 2009): Despite its simplicity, MF remains competitive with modern recommenders (Rendle et al. 2022). We use a dynamic variant that derives user embeddings directly from interaction vectors.

**Variational Autoencoder (VAE)** (Liang et al. 2018; Shenbin et al. 2020): A generative latent variable model that reconstructs user-item vectors from compressed representations.

**Neural Collaborative Filtering (NCF)** (He et al. 2017): A hybrid architecture combining matrix factorization and multi-layer perceptrons to model nonlinear user-item interactions.

## Datasets

Experiments were conducted on three datasets: **ML1M** (Harper and Konstan 2015), **Yahoo! Music** (Dror et al. 2012), and **Pinterest** (He et al. 2017). All datasets were binarized to implicit feedback, with an 80/20 user-based train-test split. An additional 10% of users were withheld from training for hyperparameter tuning. All results are reported on the test set, where explanations target the top recommendation per user.

## Baselines and Methods

We compare SPINRec against a broad set of post-hoc, model-agnostic explanation baselines, spanning heuristic, perturbation-based, and learning-based methods:

**Cosine Similarity:** A non-counterfactual heuristic that ranks user-history items by cosine similarity to the recommended item (Singh et al. 2020).

**SHAP4Rec** (Zhong and Negre 2022): A perturbation-based method grounded in Shapley values (Lundberg and Lee 2017b), adapted for recommendation via Jaccard-based clustering and  $K = 10$  k-means sampling, as in (Barkan et al. 2024).

**DeepSHAP** (Lundberg and Lee 2017a): A fast SHAP approximation using DeepLIFT-style gradient propagation (Shrikumar, Greenside, and Kundaje 2017).

**LIME-RS** (Nóbrega and Marinho 2019): A LIME adaptation for recommender systems, fitting a local linear surrogate model around a perturbed user profile.

**LIRE** (Brunot et al. 2022): A robust LIME variant using importance sampling to improve faithfulness in sparse recommendation domains.

**FIA** (Cheng et al. 2019): An approach utilizing influence functions to estimate the effect of each user feature.

**ACCENT** (Tran, Ghazimatin, and Saha Roy 2021): A fidelity-aware explainer based on influence functions (Koh and Liang 2017), extending FIA to capture second-order model effects.

**LXR** (Barkan et al. 2024): A state-of-the-art fidelity-aware method that learns an auxiliary explainer network to optimize counterfactual metrics under perturbation.

**PI (Ablated SPINRec):** A vanilla path-integration baseline that omits the stochastic baseline sampling of SPINRec. This model serves to isolate the contribution of sampling, showing that while PI alone achieves strong fidelity, it remains suboptimal. The full SPINRec significantly outperforms ABLT across all settings, demonstrating the importance of adapting PI to sparse recommender data.

**SPINRec (Ours):** The proposed method combines path integration with fidelity-optimized stochastic baseline sampling to generate high-precision attribution maps for recommendation outcomes.

## Counterfactual Evaluation Results

We evaluate SPINRec across three recommender architectures (MF, VAE, NCF) and three benchmark datasets (ML1M, Yahoo! Music, Pinterest), using both AUC-style (Barkan et al.

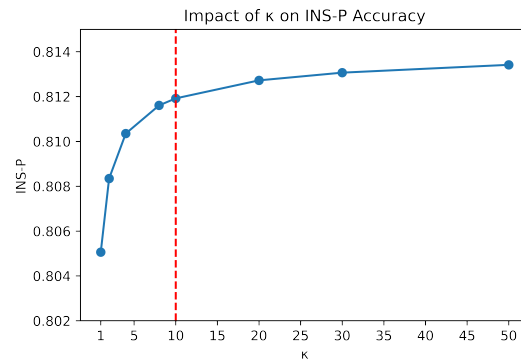


Figure 2: Fidelity (INS) vs. number of baseline samples ( $\kappa$ ) for NCF on ML1M. Gains plateau after  $\kappa=10$ .

2024) and fixed-length (Baklanov et al. 2025) counterfactual fidelity metrics. Results for Pinterest are reported in our public repository.

**AUC-Based Metrics.** Tables 1–2 report Area-Under-Curve (AUC) scores, summarizing fidelity degradation under step-wise perturbations. Across all models and datasets, SPINRec achieves the best results, significantly surpassing strong baselines such as LXR and FIA ( $p \leq 0.01$ , paired t-test).

**Fixed-Length Fidelity Metrics.** Tables 3–4 present fidelity at fixed explanation lengths  $K_e \in \{2, 3, 4\}$  and ranking cutoffs  $K_r \in \{5, 10, 20\}$ , simulating realistic user-facing scenarios where concise, high-impact explanations are crucial. As  $K_r$  increases and  $K_e$  decreases, the counterfactual test becomes more challenging: fewer explanatory items must shift the recommended item beyond a stricter cutoff. This difficulty is reflected in tighter performance margins, especially at  $K_e=2$  and 3, where multiple methods sometimes tie. We omit  $K_e=1$  here due to its instability and limited discriminative power, but include results for  $K_e=1$  and 5 in our public repository, where trends remain consistent. SPINRec again outperforms all baselines across all configurations, confirming its robustness across fidelity granularities.

## Ablation Study: Plain vs. Stochastic Path Integration

To isolate the contribution of stochastic baseline sampling, we compare SPINRec to its ablated variant (**PI**), which employs plain path integration without sampling. While PI performs competitively and often ranks near the top, SPINRec consistently achieves superior fidelity across most metrics and datasets. This improvement reflects a key insight: modern recommenders rely not only on observed interactions but also on their absence as informative signals. By sampling diverse, non-zero baselines, SPINRec allows missing items to contribute meaningful gradients. Gains over PI are most pronounced in advanced models (VAE, NCF), where unobserved interactions play a larger role.

**Impact of Sampling Count  $\kappa$ .** Figure 2 shows the effect of varying the number of sampled baselines  $\kappa$  on explanation fidelity. Performance plateaus at around  $\kappa=10$ , indicating a strong balance between fidelity and computational efficiency.

Rec	Method	POS@5 ↓	POS@10 ↓	POS@20 ↓	DEL ↓	INS ↑	CDCG ↓
MF	Cosine	0.646	0.703	0.744	0.776	0.911	0.589
	SHAP	0.812	0.857	0.883	0.851	0.858	0.734
	DeepSHAP	0.431	0.499	0.547	<u>0.564</u>	<u>0.938</u>	0.422
	LIME	0.644	0.725	0.778	0.735	0.928	0.576
	LIRE	0.588	0.651	0.694	0.678	0.926	0.512
	FIA	0.432	0.497	0.543	0.570	0.938	0.422
	ACCENT	0.707	0.760	0.797	0.729	0.910	0.652
	LXR	0.457	0.521	0.571	0.593	0.936	0.442
	PI	0.418	0.483	0.529	<b>0.555</b>	<b>0.939</b>	0.411
SPINRec	<b>0.410</b>	<b>0.478</b>	<b>0.527</b>	<b>0.555</b>	<b>0.939</b>	<b>0.405</b>	
VAE	Cosine	0.412	0.501	0.595	0.007	0.020	0.435
	SHAP	0.602	0.689	0.766	0.011	0.011	0.572
	DeepSHAP	0.340	0.410	0.490	0.007	0.025	0.396
	LIME	0.502	0.604	0.698	0.008	0.016	0.502
	LIRE	0.345	0.437	0.536	0.007	0.021	0.392
	FIA	0.234	0.312	0.411	<b>0.005</b>	<u>0.029</u>	0.320
	ACCENT	0.483	0.565	0.649	0.007	0.017	0.505
	LXR	0.348	0.430	0.518	<u>0.006</u>	0.022	0.394
	PI	0.236	0.319	0.416	<b>0.005</b>	<u>0.029</u>	0.322
SPINRec	<b>0.189</b>	<b>0.252</b>	<b>0.335</b>	<b>0.005</b>	<b>0.031</b>	<b>0.293</b>	
NCF	Cosine	0.263	0.315	0.360	0.485	0.769	0.312
	SHAP	0.501	0.559	0.602	0.620	0.667	0.504
	DeepSHAP	<u>0.210</u>	<u>0.247</u>	<u>0.282</u>	<u>0.387</u>	<u>0.805</u>	<u>0.272</u>
	LIME	0.291	0.345	0.391	0.484	0.774	0.334
	LIRE	0.301	0.350	0.397	0.474	0.776	0.338
	FIA	0.215	0.256	0.293	0.591	<u>0.805</u>	0.276
	ACCENT	0.306	0.348	0.387	0.462	0.774	0.347
	LXR	0.249	0.301	0.346	0.451	0.790	0.303
	PI	0.211	0.248	0.283	0.389	0.807	0.273
SPINRec	<b>0.185</b>	<b>0.223</b>	<b>0.261</b>	<b>0.382</b>	<b>0.810</b>	<b>0.258</b>	

Table 1: ML1M Dataset

Rec	Method	POS@5 ↓	POS@10 ↓	POS@20 ↓	DEL ↓	INS ↑	CDCG ↓
MF	Cosine	0.331	0.436	0.504	0.695	0.868	0.382
	SHAP	0.530	0.637	0.697	0.765	0.821	0.533
	DeepSHAP	0.258	0.348	0.401	<u>0.592</u>	<u>0.882</u>	0.324
	LIME	0.360	0.463	0.530	0.681	0.871	0.402
	LIRE	0.424	0.525	0.581	0.701	0.858	0.448
	FIA	0.263	0.352	0.408	0.601	<u>0.882</u>	0.328
	ACCENT	0.364	0.452	0.505	0.646	0.874	0.411
	LXR	0.282	0.371	0.428	0.620	0.879	0.344
	PI	0.256	0.344	0.398	<b>0.591</b>	<b>0.883</b>	<u>0.323</u>
SPINRec	<b>0.246</b>	<b>0.337</b>	<b>0.393</b>	<b>0.591</b>	<b>0.883</b>	<b>0.318</b>	
VAE	Cosine	0.402	0.503	0.605	0.014	0.042	0.432
	SHAP	0.605	0.690	0.766	0.021	0.031	0.576
	DeepSHAP	0.362	0.454	0.558	0.014	0.043	0.410
	LIME	0.576	0.664	0.744	0.021	0.032	0.554
	LIRE	0.453	0.559	0.665	0.017	0.038	0.463
	FIA	0.315	0.420	0.542	0.013	0.048	0.377
	ACCENT	0.541	0.622	0.704	0.019	0.032	0.553
	LXR	0.393	0.488	0.589	0.014	0.041	0.429
	PI	0.320	0.424	0.545	0.013	<u>0.048</u>	0.380
SPINRec	<b>0.280</b>	<b>0.376</b>	<b>0.489</b>	<b>0.012</b>	<b>0.049</b>	<b>0.358</b>	
NCF	Cosine	0.456	0.512	0.579	0.619	0.723	0.466
	SHAP	0.561	0.603	0.657	0.657	0.683	0.559
	DeepSHAP	0.253	0.285	0.339	0.526	<u>0.768</u>	0.315
	LIME	0.290	0.328	0.386	0.549	0.762	0.347
	LIRE	0.417	0.460	0.517	0.590	0.736	0.443
	FIA	0.254	0.288	0.342	0.527	<u>0.768</u>	0.317
	ACCENT	0.312	0.346	0.399	0.547	0.758	0.366
	LXR	0.267	0.300	0.353	0.536	0.764	0.327
	PI	0.250	0.284	0.338	0.527	<u>0.768</u>	0.313
SPINRec	<b>0.238</b>	<b>0.273</b>	<b>0.328</b>	<b>0.525</b>	<b>0.769</b>	<b>0.305</b>	

Table 2: Yahoo Dataset

Method	POS@5, $K_e$ ↓			POS@10, $K_e$ ↓			POS@20, $K_e$ ↓			DEL@ $K_e$ ↓			INS@ $K_e$ ↑			CDCG@ $K_e$ ↓		
	$K_e=2$	$K_e=3$	$K_e=4$	$K_e=2$	$K_e=3$	$K_e=4$	$K_e=2$	$K_e=3$	$K_e=4$	$K_e=2$	$K_e=3$	$K_e=4$	$K_e=2$	$K_e=3$	$K_e=4$	$K_e=2$	$K_e=3$	$K_e=4$
<b>MF</b>																		
Cosine	0.987	0.964	0.945	0.996	0.988	0.974	0.998	0.993	0.988	0.983	0.974	0.965	0.750	0.785	0.812	0.915	0.883	0.861
SHAP	1.000	0.999	0.994	1.000	0.999	0.998	1.000	1.000	1.000	0.997	0.995	0.992	0.648	0.662	0.676	0.969	0.956	0.945
DeepSHAP	<u>0.980</u>	<u>0.945</u>	<u>0.921</u>	<b>0.993</b>	0.974	0.957	<b>0.997</b>	<u>0.991</u>	<u>0.978</u>	<u>0.975</u>	<u>0.960</u>	<u>0.945</u>	0.795	0.842	0.877	0.886	0.836	0.791
LIME	0.987	0.964	0.948	<u>0.994</u>	0.988	0.976	<b>0.997</b>	0.993	0.988	0.980	0.968	0.958	0.751	0.788	0.818	0.916	0.884	0.850
LIRE	0.993	0.974	0.964	0.999	0.993	0.985	1.000	0.997	0.990	0.985	0.977	0.969	0.764	0.804	0.836	0.926	0.894	0.871
FIA	<u>0.980</u>	<u>0.945</u>	<u>0.916</u>	<b>0.993</b>	<u>0.974</u>	<u>0.953</u>	<b>0.997</b>	<b>0.988</b>	<b>0.974</b>	<b>0.974</b>	<b>0.959</b>	<b>0.944</b>	0.796	0.842	0.876	0.884	0.831	<u>0.788</u>
ACCENT	0.994	0.984	0.977	0.998	0.993	0.987	<u>0.998</u>	0.995	0.991	0.983	0.973	0.962	0.750	0.786	0.815	0.953	0.933	0.910
LXR	0.988	0.958	0.933	0.997	0.980	0.962	1.000	0.993	0.978	0.979	0.966	0.951	0.779	0.828	0.865	0.904	0.862	0.820
PI	<u>0.980</u>	<u>0.945</u>	<u>0.916</u>	<b>0.993</b>	0.974	<u>0.953</u>	<b>0.997</b>	<b>0.988</b>	<b>0.974</b>	<b>0.974</b>	<b>0.959</b>	<b>0.944</b>	0.798	<u>0.845</u>	<u>0.879</u>	0.884	0.832	0.789
SPINRec	<b>0.975</b>	<b>0.940</b>	<b>0.911</b>	<b>0.993</b>	<b>0.972</b>	<b>0.952</b>	<b>0.997</b>	<b>0.988</b>	<b>0.974</b>	<b>0.974</b>	<b>0.959</b>	<b>0.944</b>	<b>0.799</b>	<b>0.846</b>	<b>0.880</b>	<b>0.859</b>	<b>0.806</b>	<b>0.762</b>
<b>VAE</b>																		
Cosine	0.976	0.942	0.906	0.993	0.982	0.961	0.998	0.995	0.986	0.895	0.856	0.821	2.807	3.125	3.286	0.861	0.825	0.782
SHAP	0.994	0.984	0.978	0.999	0.997	0.989	0.999	0.998	0.997	0.986	0.983	0.973	0.617	0.656	0.696	0.941	0.922	0.903
DeepSHAP	0.952	0.903	0.840	0.986	0.964	0.929	<u>0.997</u>	0.989	0.970	0.876	0.832	0.795	2.162	2.536	2.770	0.828	0.772	0.723
LIME	0.985	0.959	0.938	0.996	0.991	0.973	0.999	0.998	0.993	0.936	0.911	0.886	1.015	1.108	1.216	0.893	0.863	0.830
LIRE	0.967	0.942	0.909	0.991	0.980	0.969	0.998	0.995	0.990	0.878	0.836	0.800	1.588	1.836	2.014	0.841	0.784	0.737
FIA	<u>0.921</u>	<u>0.844</u>	<u>0.752</u>	<u>0.978</u>	<u>0.937</u>	<u>0.891</u>	<b>0.996</b>	<u>0.983</u>	<u>0.960</u>	<u>0.810</u>	<u>0.746</u>	<u>0.694</u>	2.002	2.429	2.703	0.763	<u>0.680</u>	<u>0.616</u>
ACCENT	0.988	0.968	0.948	0.998	0.983	0.974	0.998	0.995	0.986	0.901	0.864	0.831	0.998	1.139	1.246	0.945	0.920	0.895
LXR	0.983	0.944	0.906	0.995	0.979	0.959	0.999	0.994	0.982	0.906	0.864	0.826	<u>2.823</u>	<u>3.369</u>	<u>3.873</u>	0.877	0.835	0.782
PI	0.922	0.848	0.770	0.981	0.937	0.902	0.997	0.984	0.963	0.814	0.752	0.701	1.987	2.400	2.676	0.761	0.689	0.626
SPINRec	<b>0.903</b>	<b>0.791</b>	<b>0.687</b>	<b>0.972</b>	<b>0.923</b>	<b>0.857</b>	<b>0.996</b>	<b>0.978</b>	<b>0.945</b>	<b>0.808</b>	<b>0.741</b>	<b>0.685</b>	<b>3.626</b>	<b>4.197</b>	<b>4.524</b>	<b>0.703</b>	<b>0.616</b>	<b>0.550</b>
<b>NCF</b>																		
Cosine	0.906	0.835	0.762	0.964	0.917	0.845	0.983	0.945	0.908	0.947	0.922	0.897	0.572	0.637	0.696	0.811	0.740	0.674
SHAP	0.979	0.955	0.930	0.998	0.997	0.983	1.000	0.993	0.983	0.985	0.977	0.968	0.480	0.507	0.533	0.910	0.879	0.845
DeepSHAP	0.908	0.820	0.728	0.986	0.964	0.929	0.997	0.938	0.887	0.936	0.905	0.875	0.600	<u>0.673</u>	<u>0.739</u>	0.788	0.711	0.634
LIME	0.939	0.884	0.823	0.971	0.934	0.887	0.984	0.959	0.923	0.948	0.924	0.901	0.560	0.615	0.666	0.836	0.772	0.717
LIRE	0.935	0.879	0.825	0.974	0.932	0.895	0.988	0.966	0.937	0.947	0.924	0.902	0.557	0.613	0.663	0.824	0.765	0.712
FIA	<u>0.887</u>	<u>0.792</u>	<u>0.699</u>	<u>0.945</u>	<u>0.887</u>	<u>0.806</u>	<u>0.971</u>	<u>0.930</u>	<u>0.882</u>	<b>0.929</b>	<b>0.896</b>	<b>0.865</b>	0.586	0.655	0.715	<u>0.775</u>	<u>0.685</u>	<u>0.615</u>
ACCENT	0.957	0.921	0.871	0.981	0.955	0.916	0.991	0.970	0.954	0.944	0.917	0.892	0.556	0.612	0.664	0.892	0.840	0.791
LXR	0.923	0.840	0.737	0.959	0.911	0.842	0.976	0.943	0.899	0.938	0.908	0.877	0.587	0.659	0.727	0.813	0.732	

Method	POS@5, $K_e \downarrow$			POS@10, $K_e \downarrow$			POS@20, $K_e \downarrow$			DEL@ $K_e \downarrow$			INS@ $K_e \uparrow$			CDCG@ $K_e \downarrow$		
	$K_e=2$	$K_e=3$	$K_e=4$	$K_e=2$	$K_e=3$	$K_e=4$	$K_e=2$	$K_e=3$	$K_e=4$	$K_e=2$	$K_e=3$	$K_e=4$	$K_e=2$	$K_e=3$	$K_e=4$	$K_e=2$	$K_e=3$	$K_e=4$
<b>MF</b>																		
Cosine	0.789	0.672	0.586	0.872	0.786	0.723	0.911	0.845	0.795	0.926	0.895	0.869	0.769	0.839	0.885	0.687	0.597	0.532
SHAP	0.879	0.811	0.754	0.927	0.883	0.843	0.947	0.915	0.879	0.949	0.925	0.905	0.655	0.709	0.751	0.811	0.748	0.703
DeepSHAP	0.745	0.619	0.501	0.839	0.749	0.662	0.889	0.817	0.744	0.912	0.875	0.843	0.826	0.897	0.939	0.643	0.549	0.476
LIME	0.803	0.704	0.622	0.882	0.804	0.747	0.912	0.854	0.810	0.923	0.891	0.864	0.772	0.840	0.886	0.714	0.630	0.567
LIRE	0.862	0.776	0.711	0.918	0.864	0.818	0.939	0.903	0.868	0.942	0.916	0.892	0.746	0.814	0.864	0.762	0.694	0.635
FIA	0.743	0.610	0.492	0.837	0.744	0.652	0.887	0.812	0.738	<b>0.911</b>	<b>0.874</b>	<b>0.841</b>	0.825	0.895	0.936	0.638	0.545	0.475
ACCENT	0.833	0.753	0.676	0.888	0.826	0.775	0.918	0.867	0.824	0.920	0.888	0.860	0.788	0.857	0.902	0.778	0.704	0.641
LXR	0.778	0.652	0.541	0.859	0.769	0.687	0.900	0.832	0.763	0.917	0.881	0.849	0.813	0.886	0.930	0.678	0.581	0.506
PI	0.743	0.609	0.492	0.837	0.744	0.652	<b>0.879</b>	0.813	0.738	<b>0.911</b>	<b>0.874</b>	<b>0.841</b>	0.825	0.894	0.934	0.638	0.546	0.476
SPINRec	<b>0.722</b>	<b>0.579</b>	<b>0.458</b>	<b>0.829</b>	<b>0.731</b>	<b>0.633</b>	0.884	<b>0.806</b>	<b>0.729</b>	<b>0.911</b>	<b>0.874</b>	<b>0.841</b>	<b>0.829</b>	<b>0.899</b>	<b>0.940</b>	<b>0.608</b>	<b>0.512</b>	<b>0.442</b>
<b>VAE</b>																		
Cosine	0.834	0.741	0.671	0.908	0.844	0.782	0.943	0.897	0.854	0.739	0.657	0.597	2.026	2.093	2.103	0.716	0.637	0.577
SHAP	0.900	0.848	0.804	0.938	0.895	0.857	0.957	0.925	0.894	0.886	0.836	0.793	0.711	0.785	0.849	0.825	0.770	0.725
DeepSHAP	0.813	0.702	0.614	0.895	0.820	0.752	0.939	0.887	0.838	0.737	0.653	0.590	1.785	1.885	1.927	0.687	0.600	0.538
LIME	0.897	0.836	0.791	0.934	0.897	0.854	0.991	0.926	0.896	0.885	0.831	0.788	0.921	1.015	1.077	0.809	0.754	0.710
LIRE	0.875	0.803	0.738	0.930	0.881	0.832	0.956	0.922	0.886	0.805	0.731	0.672	1.319	1.487	1.566	0.750	0.676	0.617
FIA	0.759	0.642	0.560	0.878	0.792	0.718	0.934	0.883	0.833	0.671	0.584	0.520	1.972	2.138	2.232	0.624	0.536	0.479
ACCENT	0.887	0.827	0.775	0.925	0.878	0.835	0.952	0.912	0.860	0.851	0.792	0.744	0.910	0.975	1.016	0.844	0.788	0.739
LXR	0.878	0.810	0.760	0.921	0.867	0.830	0.978	0.905	0.877	0.867	0.806	0.760	0.882	0.965	1.000	0.805	0.744	0.697
PI	0.769	0.649	0.571	0.883	0.799	0.723	0.936	0.888	0.835	0.675	0.588	0.523	1.979	2.159	2.249	0.631	0.543	0.484
SPINRec	<b>0.740</b>	<b>0.603</b>	<b>0.498</b>	<b>0.870</b>	<b>0.767</b>	<b>0.686</b>	<b>0.930</b>	<b>0.871</b>	<b>0.813</b>	<b>0.665</b>	<b>0.573</b>	<b>0.505</b>	<b>2.348</b>	<b>2.539</b>	<b>2.610</b>	<b>0.590</b>	<b>0.500</b>	<b>0.440</b>
<b>NCF</b>																		
Cosine	0.796	0.710	0.639	0.843	0.762	0.694	0.879	0.814	0.758	0.911	0.878	0.851	0.738	0.786	0.819	0.734	0.657	0.599
SHAP	0.871	0.810	0.755	0.889	0.830	0.779	0.910	0.863	0.819	0.947	0.923	0.901	0.650	0.688	0.719	0.828	0.772	0.726
DeepSHAP	0.705	0.583	0.489	0.767	0.650	0.551	0.817	0.713	0.636	0.877	0.833	0.796	0.793	0.853	0.892	0.655	0.559	0.481
LIME	0.718	0.607	0.514	0.784	0.670	0.585	0.833	0.733	0.658	0.882	0.840	0.805	0.782	0.839	0.877	0.673	0.580	0.512
LIRE	0.826	0.747	0.669	0.866	0.789	0.721	0.899	0.843	0.782	0.919	0.887	0.857	0.728	0.780	0.820	0.768	0.696	0.635
FIA	0.700	0.579	0.484	0.765	0.647	0.551	0.817	0.710	0.634	<b>0.876</b>	0.833	0.796	0.793	0.853	0.892	0.652	0.556	0.481
ACCENT	0.766	0.666	0.579	0.806	0.711	0.628	0.849	0.763	0.695	0.887	0.846	0.813	0.771	0.827	0.866	0.730	0.640	0.576
LXR	0.735	0.628	0.526	0.786	0.675	0.590	0.833	0.738	0.662	0.889	0.845	0.808	0.769	0.832	0.874	0.689	0.594	0.519
PI	0.704	0.585	0.489	0.768	0.653	0.554	0.819	0.715	0.636	0.877	0.833	0.796	0.792	0.852	0.891	0.657	0.562	0.486
SPINRec	<b>0.686</b>	<b>0.558</b>	<b>0.455</b>	<b>0.757</b>	<b>0.631</b>	<b>0.535</b>	<b>0.809</b>	<b>0.703</b>	<b>0.625</b>	<b>0.876</b>	<b>0.832</b>	<b>0.795</b>	<b>0.796</b>	<b>0.856</b>	<b>0.895</b>	<b>0.621</b>	<b>0.522</b>	<b>0.448</b>

Table 4: Fidelity Metrics at Different  $K_r$ ,  $K_e$  Values for MF, VAE, and NCF on Yahoo!

**Summary and Insights.** Across all metrics, datasets, and recommender architectures, SPINRec consistently outperforms existing explanation methods, establishing a new fidelity benchmark for recommendation. Path integration (PI) itself proves to be an inherently strong approach for fidelity, even without stochastic sampling. Incorporating stochastic baseline sampling further enhances fidelity by leveraging both observed and unobserved interactions, an effect particularly pronounced in more expressive models such as VAE and NCF.

## Conclusion

We introduced SPINRec, the first model-agnostic explanation method to apply path integration (PI) to recommender systems. By combining PI with a stochastic baseline sampling strategy tailored to sparse, binary user-item data, SPINRec reproduces stable, high-fidelity explanations that more faithfully capture model reasoning. A comprehensive evaluation across multiple models, datasets, and fidelity metrics demonstrates that SPINRec consistently outperforms existing approaches, establishing a new benchmark for fidelity-aware explainability in recommendation. We hope this work encourages further exploration of fidelity-focused explainability in recommendation.

## Acknowledgments

This work was supported by the Ministry of Innovation, Science & Technology, Israel.

## References

- Abdollahi, B.; and Nasraoui, O. 2016. Explainable matrix factorization for collaborative filtering. In *Proceedings of the 25th International Conference Companion on World Wide Web*, 5–6.
- Abdollahi, B.; and Nasraoui, O. 2017. Using explainability for constrained matrix factorization. In *Proceedings of the eleventh ACM conference on recommender systems*, 79–83.
- Agarwal, C.; Krishna, S.; Saxena, E.; Pawelczyk, M.; Johnson, N.; Puri, I.; Zitnik, M.; and Lakkaraju, H. 2022. Openxai: Towards a transparent evaluation of model explanations. *Advances in Neural Information Processing Systems*, 35: 15784–15799.
- Baklanov, M.; Bogina, V.; Elisha, Y.; Schein, Y.; Allerhand, L.; Barkan, O.; and Koenigstein, N. 2025. Refining Fidelity Metrics for Explainable Recommendations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2967–2971.
- Barkan, O.; Bogina, V.; Gurevitch, L.; Asher, Y.; and Koenigstein, N. 2024. A Counterfactual Framework for Learning and Evaluating Explanations for Recommender Systems. In *Proceedings of the ACM on Web Conference 2024*, 3723–3733.
- Barkan, O.; Elisha, Y.; Asher, Y.; Eshel, A.; and Koenigstein, N. 2023a. Visual Explanations via Iterated Integrated Attribu-

- tions. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2073–2084.
- Barkan, O.; Elisha, Y.; Weill, J.; Asher, Y.; Eshel, A.; and Koenigstein, N. 2023b. Deep integrated explanations. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 57–67.
- Barkan, O.; Elisha, Y.; Weill, J.; and Koenigstein, N. 2025. BEE: Metric-Adapted Explanations via Baseline Exploration-Exploitation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 1835–1843.
- Barkan, O.; Fuchs, Y.; Caciularu, A.; and Koenigstein, N. 2020. Explainable recommendations via attentive multi-persona collaborative filtering. In *Proceedings of the 14th ACM Conference on Recommender Systems*, 468–473.
- Barkan, O.; Hirsch, R.; Katz, O.; Caciularu, A.; Weill, J.; and Koenigstein, N. 2021. Cold item integration in deep hybrid recommenders via tunable stochastic gates. In *2021 IEEE International Conference on Data Mining (ICDM)*, 994–999. IEEE.
- Barkan, O.; Katz, O.; and Koenigstein, N. 2020. Neural attentive multiview machines. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3357–3361. IEEE.
- Barkan, O.; Koenigstein, N.; Yogev, E.; and Katz, O. 2019. CB2CF: a neural multiview content-to-collaborative filtering model for completely cold item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, 228–236.
- Barkan, O.; Shaked, T.; Fuchs, Y.; and Koenigstein, N. 2023c. Modeling users’ heterogeneous taste with diversified attentive user profiles. *User Modeling and User-Adapted Interaction*, 1–31.
- Brunot, L.; Canovas, N.; Chanson, A.; Labroche, N.; and Verdeaux, W. 2022. Preference-based and local post-hoc explanations for recommender systems. *Information Systems*, 108: 102021.
- Cheng, W.; Shen, Y.; Huang, L.; and Zhu, Y. 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*, 885–893.
- Dror, G.; Koenigstein, N.; Koren, Y.; and Weimer, M. 2012. The yahoo! music dataset and kdd-cup’11. In *Proceedings of KDD Cup 2011*, 3–18. PMLR.
- Enguehard, J. 2023. Sequential integrated gradients: a simple but effective method for explaining language models. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 7555–7565. Toronto, Canada: Association for Computational Linguistics.
- Erion, G.; Janizek, J. D.; Sturmfels, P.; Lundberg, S. M.; and Lee, S.-I. 2021. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature machine intelligence*, 3(7): 620–631.
- Fan, W.; Zhao, X.; Chen, X.; Su, J.; Gao, J.; Wang, L.; Liu, Q.; Wang, Y.; Xu, H.; Chen, L.; et al. 2022. A comprehensive survey on trustworthy recommender systems. *arXiv preprint arXiv:2209.10117*.
- Gaiger, K.; Barkan, O.; Tsipory-Samuel, S.; and Koenigstein, N. 2023. Not All Memories Created Equal: Dynamic User Representations for Collaborative Filtering. *IEEE Access*, 1–1.
- Gurevitch, L.; Bogina, V.; Barkan, O.; Schein, Y.; Elisha, Y.; and Koenigstein, N. 2025. LXR: Learning to eXplain Recommendations. *ACM Transactions on Recommender Systems*.
- Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4): 1–19.
- Haug, J.; Zürn, S.; El-Jiz, P.; and Kasneci, G. 2021. On baselines for local feature attributions. *arXiv preprint arXiv:2101.00905*.
- He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 639–648.
- He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.
- Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.
- Kapishnikov, A.; Venugopalan, S.; Avci, B.; Wedin, B.; Terry, M.; and Bolukbasi, T. 2021. Guided integrated gradients: An adaptive path method for removing noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5050–5058.
- Katz, O.; Barkan, O.; Koenigstein, N.; and Zabari, N. 2022. Learning to ride a buy-cycle: A hyper-convolutional model for next basket repurchase recommendation. In *Proceedings of the 16th ACM Conference on Recommender Systems*, 316–326.
- Koenigstein, N. 2025. Without Fidelity, Explanations Are Just Stories: Rethinking Evaluation in Explainable Recommender Systems. *SSRN (September 26, 2025)*.
- Koh, P. W.; and Liang, P. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, 1885–1894. PMLR.
- Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.
- Kunkel, J.; Donkers, T.; Michael, L.; Barbu, C.-M.; and Ziegler, J. 2019. Let me explain: Impact of personal and impersonal explanations on trust in recommender systems. In *Proceedings of the 2019 CHI conference on human factors in computing systems*, 1–12.
- Li, L.; Zhang, Y.; and Chen, L. 2021. Personalized Transformer for Explainable Recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4947–4957.

- Liang, D.; Krishnan, R. G.; Hoffman, M. D.; and Jebara, T. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, 689–698.
- Lundberg, S.; and Lee, S.-I. 2017a. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 4765–4774.
- Lundberg, S. M.; and Lee, S.-I. 2017b. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.
- Melchiorre, A. B.; Rekabsaz, N.; Ganhör, C.; and Schedl, M. 2022. ProtoMF: Prototype-based Matrix Factorization for Effective and Explainable Recommendations. In *Sixteenth ACM Conference on Recommender Systems (RecSys '22)*, 11. Seattle, WA, USA: ACM.
- Mohammadi, A. R.; Peintner, A.; Müller, M.; and Zangerle, E. 2025. Beyond Top-1: Addressing Inconsistencies in Evaluating Counterfactual Explanations for Recommender Systems. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, 515–520.
- Nóbrega, C.; and Marinho, L. 2019. Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 1671–1678.
- Rendle, S.; Krichene, W.; Zhang, L.; and Koren, Y. 2022. Revisiting the performance of ials on item recommendation benchmarks. In *Proceedings of the 16th ACM Conference on Recommender Systems*, 427–435.
- Ribeiro, M. T.; Singh, S.; and Guestrin, C. 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*, 1135–1144.
- Samek, W.; Binder, A.; Montavon, G.; Lapuschkin, S.; and Müller, K.-R. 2016. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11): 2660–2673.
- Sanyal, S.; and Ren, X. 2021. Discretized Integrated Gradients for Explaining Language Models. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 10285–10299. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.
- Shenbin, I.; Alekseev, A.; Tutubalina, E.; Malykh, V.; and Nikolenko, S. I. 2020. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th international conference on web search and data mining*, 528–536.
- Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *International conference on machine learning*, 3145–3153. PMIR.
- Sikdar, S.; Bhattacharya, P.; and Heese, K. 2021. Integrated directional gradients: Feature interaction attribution for neural NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 865–878.
- Singh, R. H.; Maurya, S.; Tripathi, T.; Narula, T.; and Srivastav, G. 2020. Movie recommendation system using cosine similarity and KNN. *International Journal of Engineering and Advanced Technology*, 9(5): 556–559.
- Sturmfels, P.; Lundberg, S.; and Lee, S.-I. 2020. Visualizing the Impact of Feature Attribution Baselines. *Distill*. <https://distill.pub/2020/attribution-baselines>.
- Sugahara, K.; and Okamoto, K. 2024. Hierarchical matrix factorization for interpretable collaborative filtering. *Pattern Recognition Letters*, 180: 99–106.
- Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.
- Tintarev, N. 2025. Measuring Explanation Quality—A Path Forward. In *ECAI 2025*, 22–29. IOS Press.
- Tintarev, N.; and Masthoff, J. 2015. Explaining recommendations: Design and evaluation. In *Recommender systems handbook*, 353–382. Springer.
- Tintarev, N.; and Masthoff, J. 2022. Beyond explaining single item recommendations. *Recommender Systems Handbook*, 711–756.
- Tran, K. H.; Ghazimatin, A.; and Saha Roy, R. 2021. Counterfactual explanations for neural recommenders. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1627–1631.
- Varasteh, M.; McKinnie, E.; Aird, A.; Acuña, D.; and Burke, R. 2024. Comparative Explanations for Recommendation: Research Directions. *IntRS'24: Joint Workshop on Interfaces and Human Decision Making for Recommender Systems*.
- Vig, J.; Sen, S.; and Riedl, J. 2009. Tagsplanations: explaining recommendations using tags. In *Proceedings of the 14th international conference on Intelligent user interfaces*, 47–56.
- Wang, N.; Wang, H.; Jia, Y.; and Yin, Y. 2018. Explainable recommendation via multi-task learning in opinionated text data. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 165–174.
- Xu, S.; Venugopalan, S.; and Sundararajan, M. 2020. Attribution in scale and space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9680–9689.
- Zhang, Y.; Chen, X.; et al. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1): 1–101.
- Zhang, Y.; Lai, K.; Zhang, W.; Zhang, Y.; Liu, Y.; and Ma, S. 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*.
- Zhong, J.; and Negre, E. 2022. Shap-enhanced counterfactual explanations for recommendations. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, 1365–1372.