



EXPLAINABLE RECOMMENDATION IN GRAPH NEURAL NETWORKS USING PROPAGATION PATH ANALYSIS AND COUNTERFACTUAL GRAPH EDITING

Pranav Nallari¹, Jayakumar Kaliappan^{*2}

^{1,2}School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India.

¹<http://orcid.org/0009-0003-7062-1785>¹, ²<http://orcid.org/0000-0002-6044-6667>²

Email: pranav.nallari2022@vitstudent.ac.in, *jayakumar.k@vit.ac.in

ARTICLE INFO

Article History

Received: March 6, 2026

Reviewed: April 6, 2026

Accepted: July 6, 2026

Published: August 31, 2026

Keywords:

Explainable AI,
Graph Neural Networks,
Collaborative Filtering,
LightGCN,
Counterfactual Explanation,
Recommender Systems.

ABSTRACT

Graph Neural Networks (GNNs), particularly LightGCN, have achieved strong performance in collaborative filtering-based recommendation systems. However, their black-box nature makes it difficult to explain why specific items are recommended, limiting trust and adoption in user-facing applications. In this paper, we propose a post-hoc explainability framework for LightGCN combining two complementary techniques: Propagation Path Analysis, which decomposes recommendation scores by propagation layer to attribute influence to specific training interactions, and Counterfactual Graph Editing, which identifies the most influential user-item edges through structural sensitivity analysis and targeted edge removal. We evaluate on MovieLens-1M against three baselines including random edge removal, degree-based attribution, and LIME. Our method achieves a mean absolute score drop of 0.0322, representing a 3.02× lift over random, a 1.26× lift over LIME, and a 5.35× lift over degree-based attribution. Faithfulness evaluation against post-fine-tuning ground truth yields Pearson $r = 0.536$, confirming that structural sensitivity reliably identifies influential edges without model retraining. Layer contribution analysis reveals that LightGCN recommendations are predominantly driven by direct interactions (Layer 0) and 1-hop neighbours (Layer 1), with deeper layers contributing progressively less.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Recommender systems have become fundamental infrastructure across e-commerce, streaming, and information retrieval platforms. Collaborative filtering based on Graph Neural Networks (GNNs) has emerged as a dominant paradigm, with LightGCN [1] achieving strong empirical performance by simplifying the GNN architecture to pure neighbourhood aggregation across multiple propagation layers. Despite these advances, GNN-based recommenders remain largely opaque. Users receive no explanation for why a specific item was suggested, system operators cannot audit model behaviour for bias or fairness [2], and regulatory frameworks such as the EU AI Act increasingly require explanations for automated decisions affecting individuals. Prior work has shown that a lack of transparency directly erodes user trust in algorithmic systems [3]. Existing explainability methods largely fall into two categories: gradient-based attribution and perturbation-based probing (LIME [4], SHAP [5]). Both treat the model as a black box and ignore the structural properties of GNNs — namely, that information flows through a graph topology across multiple propagation hops. This structural blindness limits their ability to produce explanations grounded in the model's actual computation. A comprehensive survey of explainability techniques for recommender systems [3] highlights that graph-structural methods remain underexplored relative to their importance. We propose a post-hoc explainability framework designed specifically for LightGCN's graph-structural characteristics. Our contributions are:

1. Propagation Path Analysis: decomposing LightGCN scores by propagation layer and attributing influence to specific training interactions at the dominant hop depth.

2. Counterfactual Graph Editing: quantifying edge influence by measuring score changes upon targeted edge removal, without model retraining.
3. Comprehensive evaluation against three baselines across four rank positions on MovieLens-1M, demonstrating 3.02x lift over random selection.
4. Faithfulness evaluation against post-fine-tuning ground truth (Pearson).
5. Qualitative layer contribution analysis across five diverse recommendation cases.

II. RELATED WORK

II.1 GNN-BASED RECOMMENDATION

By [1] proposed LightGCN as a simplification of NGCF [6], retaining only weighted neighbourhood aggregation per layer. Training uses Bayesian Personalised Ranking (BPR) [7], which optimises relative ordering of observed versus unobserved interactions. Earlier approaches such as SpectralCF [8] operated in the spectral domain but were limited by scalability. Graph Attention Networks (GAT) [9] introduced learnable attention weights over neighbours; however, LightGCN demonstrates that removing such parameterisation improves generalisation on collaborative filtering data. A broader survey of deep learning approaches to recommendation is provided by [10].

II.2 EXPLAINABILITY IN GNNs

GNExplainer [11] identifies a subgraph that maximises mutual information with the model's prediction, targeting node classification rather than recommendation. CF-GNExplainer [12] extends counterfactual perturbations to GNNs but is computationally expensive and not designed for large-scale bipartite recommendation graphs. PGExplainer [13] trains a parameterised mask generator for GNN explanations, requiring additional training. SubgraphX [14] uses Monte Carlo Tree Search to identify explanatory subgraphs at greater computational cost. These methods share a common limitation: they do not account for the bipartite, link-prediction nature of the recommendation task.

II.3 EXPLAINABILITY IN RECOMMENDER SYSTEMS

LIME [4] fits a local linear surrogate by perturbing user interaction histories and fitting weighted regression. It is model-agnostic but ignores graph topology and requires many forward passes. SHAP [5] provides Shapley value attributions but is similarly black-box. Attention-based methods [15] use attention weights as proxies, but these do not reliably correlate with feature importance [16], and LightGCN uses no attention. A systematic categorisation of post-hoc explanation approaches by type and model family is provided by [17]. In the recommendation setting, PRINCE [18] generates counterfactual explanations for knowledge-graph-based recommenders; our work targets GNN-based collaborative filtering and provides layer-level attribution alongside edge-level counterfactuals. Our work is most closely related to counterfactual approaches for recommendation [19], which we extend with layer-level propagation path analysis and a systematic multi-baseline faithfulness evaluation.

III. BACKGROUND: LIGHTGCN

III.1 PROBLEM FORMULATION

Let $\mathcal{U}$ and $\mathcal{I}$ be the sets of users and items. Observed interactions are encoded in $R \in \{0,1\}^{|\mathcal{U}| \times |\mathcal{I}|}$ where $R_{ui} = 1$ indicates an interaction. The task is to score unobserved pairs $\hat{y}_{ui}$ and rank items per user.

III.2 GRAPH CONSTRUCTION

LightGCN builds a bipartite graph $G = (V, E)$ with $V = \mathcal{U} \cup \mathcal{I}$. The full adjacency matrix is given by Eq. (1). With degree matrix D where $D_{vv} = \sum_j A_{vj}$, the symmetrically normalised adjacency is given by Eq. (2). This scales each edge by the inverse square root of source and target node degrees, preventing high-degree nodes from dominating aggregation.

III.3 PROPAGATION AND TRAINING

$$A = \begin{pmatrix} 0 & R \\ R^\top & 0 \end{pmatrix} \quad (1)$$

$$\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \quad (2)$$

At layer k , embeddings are updated by neighbourhood aggregation as shown in Eq. (3), where $e^{(0)}$ are learnable initial embeddings. In matrix form: $E^{(k)} = \tilde{A} E^{(k-1)}$. Final embeddings are the layer-averaged representations given in Eq. (4). The predicted score is $\hat{y}_{ui} = e_u^\top e_i$, and training uses BPR loss as in Eq. (5), where $\mathcal{O} = \{(u, i, j) \mid R_{ui}=1, R_{uj}=0\}$, σ is the sigmoid function, and λ is an L2 regularisation coefficient.

$$e_u^{(k)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|} \sqrt{|\mathcal{N}_i|}} e_i^{(k-1)}, e_i^{(k)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|} \sqrt{|\mathcal{N}_u|}} e_u^{(k-1)} \quad (3)$$

$$e_u = \frac{1}{K+1} \sum_{k=0}^K e_u^{(k)}, e_i = \frac{1}{K+1} \sum_{k=0}^K e_i^{(k)} \quad (4)$$

$$\mathcal{L}_{BPR} = - \sum_{(u,i,j) \in \mathcal{O}} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) + \lambda \|E^{(0)}\|^2 \quad (5)$$

IV. PROPOSED FRAMEWORK

Our framework answers two complementary questions for any recommendation (u, i) : (1) at which propagation depth does the signal originate? and (2) which training interactions are most influential? Figure 1 shows the complete pipeline.

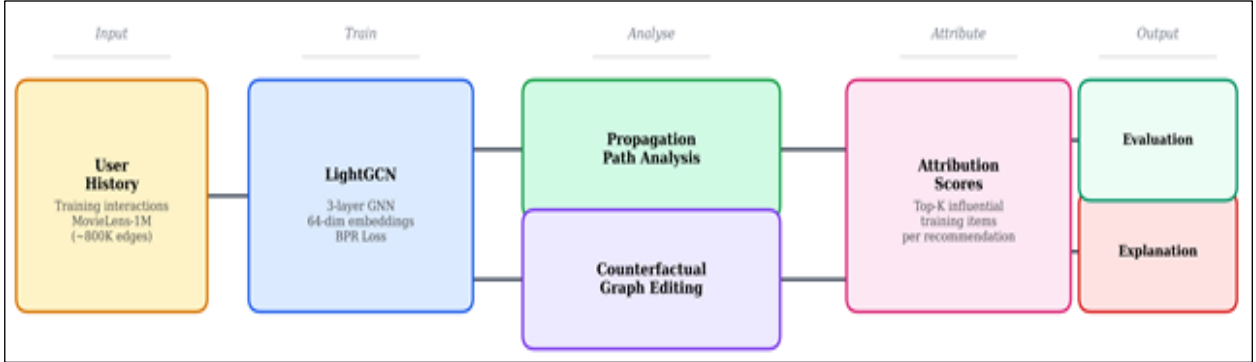


Figure 1: System architecture of the Explainable LightGCN pipeline.
Source: Authors, (2026).

IV.1 PROPAGATION PATH ANALYSIS

We define the layer contribution at hop k as the inner product of the layer- k embeddings, as in Eq. (6). This captures the contribution of information propagated exactly k hops. The dominant layer is $k^* = \arg \max_k C_k(u, i)$. A dominant C_0 indicates recommendation driven by the user's own interaction embeddings; a dominant C_1 indicates collaborative signal from items co-liked by users with overlapping histories.

$$C_k(u, i) = e_u^{(k)\top} e_i^{(k)} \quad (6)$$

For each item j in the user's history, the attribution score measures its embedding's contribution to the target item's score at the dominant layer, as in Eq. (7). The first term is the LightGCN normalisation weight for edge (u, j) ; the second measures alignment between item j 's outgoing embedding and the target item's embedding at the dominant layer. Items are ranked by $|\alpha(j, u, i)|$ to produce top- K attributed interactions.

$$\alpha(j, u, i) = \frac{1}{\sqrt{|\mathcal{N}_u|} \sqrt{|\mathcal{N}_j|}} \cdot e_j^{(k^*-1)\top} e_i^{(k^*)} \quad (7)$$

IV.2 COUNTERFACTUAL GRAPH EDITING

For attributed training item j , define the counterfactual graph G_{-j} as the original graph with edge (u, j) removed: $\mathcal{E}_{-j} = \mathcal{E} \setminus \{(u, j)\}$. We renormalise to obtain $\tilde{A}_{-j}$ and compute the structural score drop with model weights held fixed, as in Eq. (8). We also measure the rank change: $\Delta r_i^{(j)} = r_i(\tilde{A}_{-j}) - r_i(\tilde{A})$. Evaluation is performed at rank positions $r \in \{1, 5, 10, 20\}$.

$$\Delta \hat{y}_{ui}^{(j)} = \hat{y}_{ui}(\tilde{A}) - \hat{y}_{ui}(\tilde{A}_{-j}) \quad (8)$$

IV.3 FAITHFULNESS EVALUATION

To test whether structural drops predict actual model adaptation, for 50 users we compare the structural predicted drop against the actual drop after 5-epoch fine-tuning on G_{-j} with learning rate $\eta = 10^{-3}$. Faithfulness is measured via Pearson r and Spearman ρ .

V. EXPERIMENTAL SETUP

V.1 DATASET

We evaluate on MovieLens-1M [20], containing 1,000,209 ratings from 6,040 users on 3,706 movies. After 5-core filtering and binarisation to implicit feedback, the training set has approximately 800,167 interactions (80/20 split). Figure 2 shows the BPR training loss, and Figure 3 shows the degree distributions of the resulting training graph. The item distribution exhibits a pronounced long tail (median 90, mean 235), which directly motivates our analysis of why degree-based attribution fails: high-degree items are suppressed by LightGCN's symmetric normalisation.

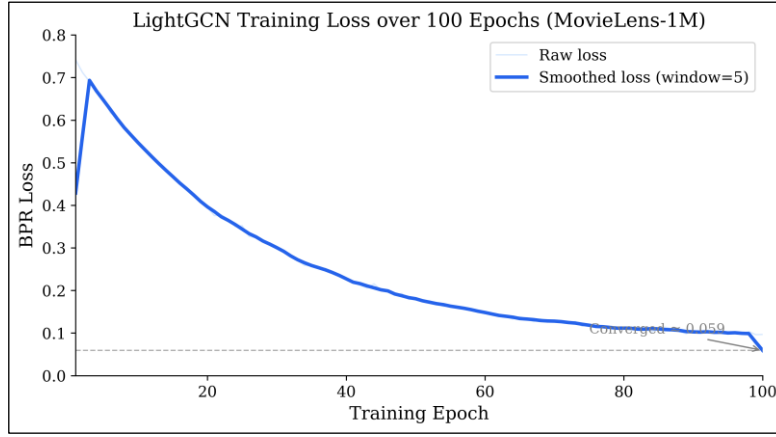


Figure 2: LightGCN BPR training loss over 100 epochs on MovieLens-1M. The model converges smoothly to approximately 0.059 by epoch 100.

Source: Authors, (2026).

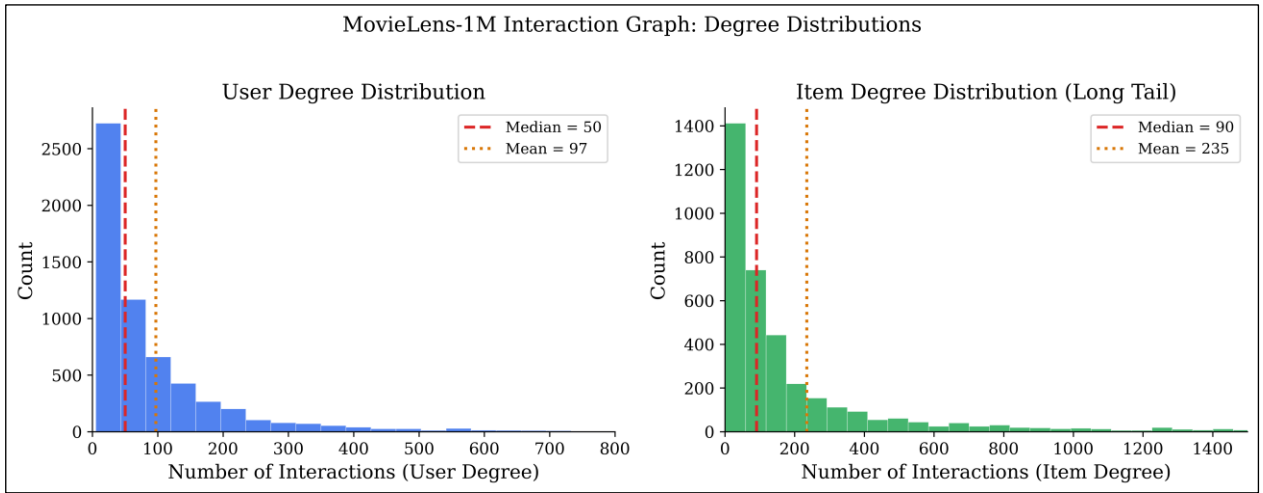


Figure 3: Degree distributions of users (left) and items (right) in the MovieLens-1M training graph after 5-core filtering.

Source: Authors, (2026).

V.2 MODEL CONFIGURATION

We train a 3-layer LightGCN ($K=3$) with embedding dimension $d=64$, Adam optimiser, learning rate 10^{-3} , BPR loss with L2 regularisation $\lambda=10^{-4}$, 100 epochs, and batch size 2048.

V.3 BASELINES

Random: Randomly sample K items from the user's training history. Tests whether PropPath beats chance.

Degree: Rank the user's training items by global degree and select the top- K . Tests the hypothesis that popular items are more influential.

LIME: Sample $N_p = 100$ binary masks over the user's history. For each mask m_t , compute the score $s(m_t) = \hat{y}_{ui}(\tilde{A}(m_t))$ and fit a weighted Ridge regression as in Eq. (9) with exponential kernel weights $w_t = \exp(-(1 - \bar{m}_t)^2 / 2\sigma^2)$, $\sigma = 0.25$. Top- K items by $|\beta_j|$ are selected.

$$\beta = \arg \min_{\beta} \sum_{t=1}^{N_p} w_t (s(m_t) - \beta^T m_t)^2 + \alpha_{ridge} \|\beta\|^2 \quad (9)$$

VI. RESULTS AND ANALYSIS

VI.1 COUNTERFACTUAL EVALUATION

Table 1 presents mean absolute score drops, and Figures 4 and 5 visualize these results. The $3.02\times$ lift over random confirms PropPath identifies genuinely influential edges. The $1.26\times$ lift over LIME demonstrates that graph-structural knowledge adds measurable value beyond black-box probing — at a fraction of the computational cost (1 forward pass vs. 100 for LIME). The degree baseline performing below random (lift $0.56\times$) is explained by symmetric normalisation: high-degree items have small individual edge weights scaling as $1/\sqrt{d_i}$, so removing any single edge causes negligible score change.

Table 1: Mean absolute score drop by method and target rank position. Higher = stronger attribution quality.

Method	R@1	R@5	R@10	R@20	Lift
PropPath (Proposed)	0.0313	0.0335	0.0298	0.0341	3.02x
LIME	0.0251	0.0268	0.0238	0.0263	2.38x
Random	0.0102	0.0121	0.0104	0.0098	1.00x
Degree	0.0055	0.0078	0.0055	0.0053	0.56x

Source: Authors, (2026).

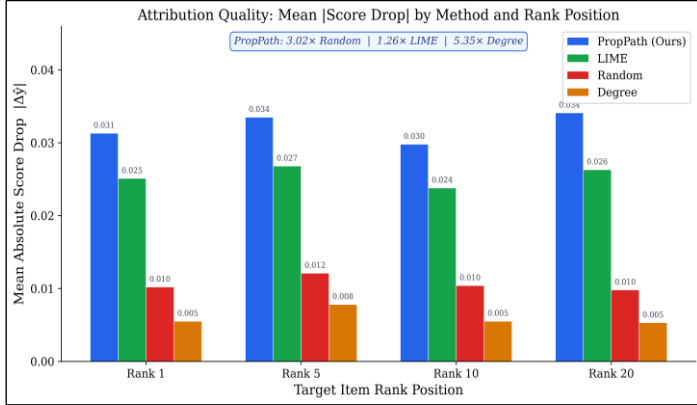


Figure 4: Mean absolute score drop per method and rank position. Source: Authors, (2026).

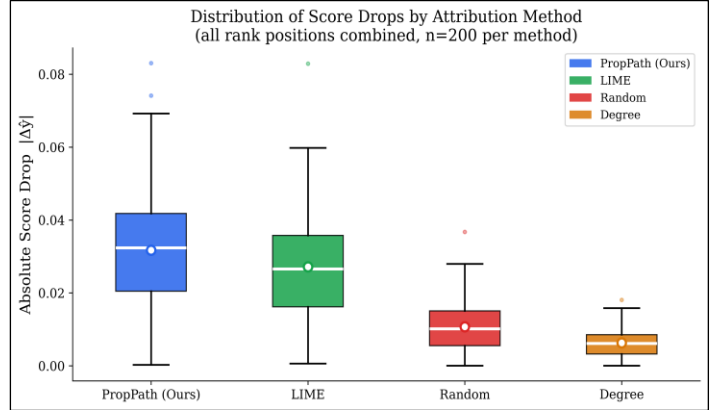


Figure 5: Distribution of score drops (n=200 per method). Source: Authors, (2026).

VI.2 RANK CHANGE ANALYSIS

Table 2 and Figure 6 present rank change rates. Rank change rates increase monotonically with rank position for PropPath and LIME, since items at rank 20 have smaller score margins. The Degree baseline's high rank change rates (up to 72%) despite low score drops reveal a deceptive pattern: removing popular items triggers widespread small normalisation-induced rank shuffles without genuinely altering the model's preference signal.

Table 2: Rank change rate (RC%) by method and target rank position.

Method	R@1	R@5	R@10	R@20	Overall
PropPath (Proposed)	6.0%	10.0%	20.0%	54.0%	22.5%
LIME	2.0%	4.0%	12.0%	52.0%	17.5%
Random	0.0%	22.0%	24.0%	40.0%	21.5%
Degree	14.0%	36.0%	60.0%	72.0%	45.5%

Source: Authors, (2026).

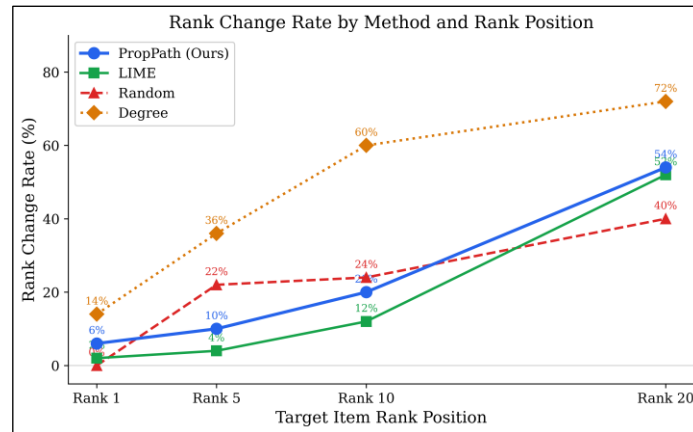


Figure 6: Rank change rate (%) by method across four target rank positions. Source: Authors, (2026).

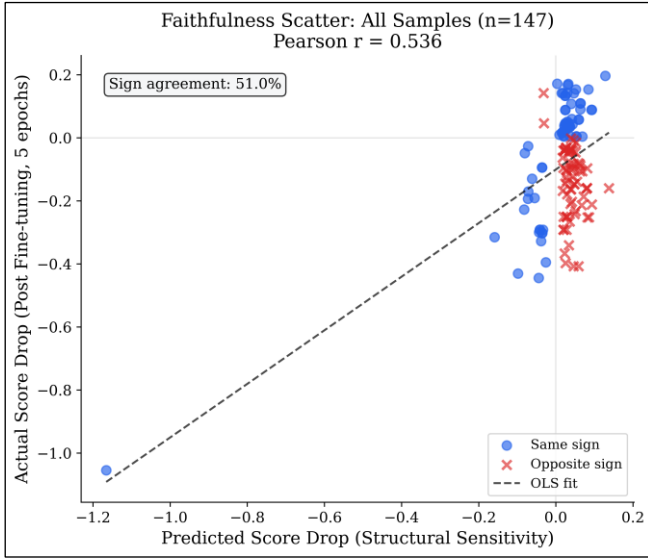
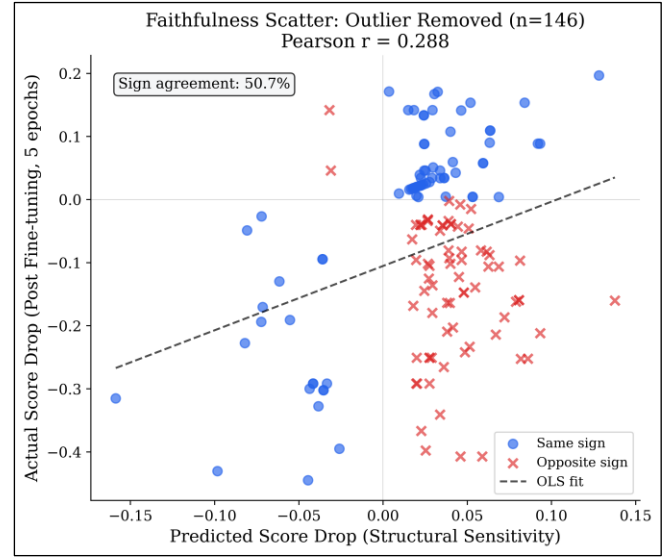
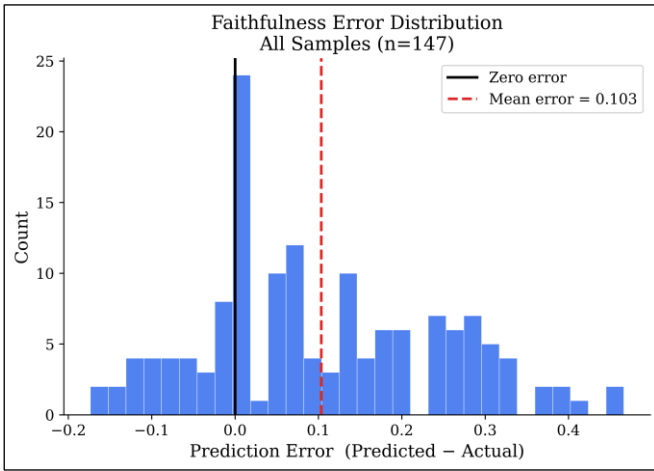
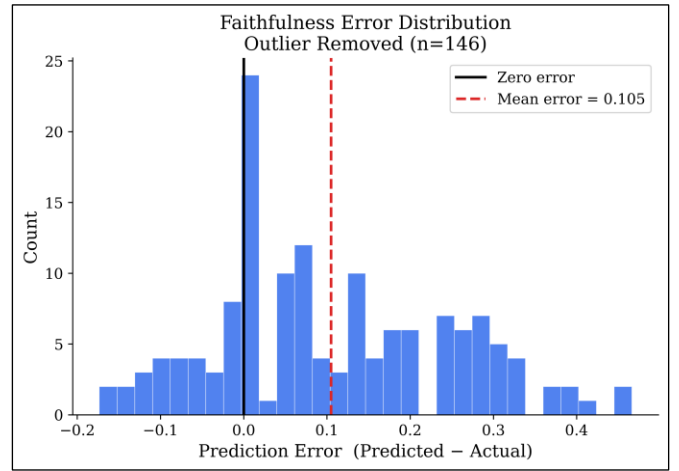
VI.3 FAITHFULNESS EVALUATION

Table 3 and Figures 7 and 8 present faithfulness results. Pearson $r=0.536$ confirms a moderate positive relationship between structural sensitivity and post-fine-tuning drops. The divergence between Pearson and Spearman ($\rho=0.281$) reflects the influence of an extreme outlier. Sign agreement at 51.0% indicates that PropPath reliably ranks edges by influence magnitude but does not reliably predict the direction of score change under fine-tuning. The systematic positive bias (mean error ≈ 0.103) confirms that structural sensitivity overestimates drops: the model compensates for removed edges via other embedding dimensions during fine-tuning.

Table 3: Faithfulness evaluation: structural sensitivity vs. post-fine-tuning ground truth.

Metric	Value
Pearson r	+0.536 ($p < 0.0001$)
Spearman ρ	+0.281 ($p = 0.0005$)
Sign Agreement	51.0% (75/147)
Sample size	147 (50 users x 3 edges)

Source: Authors, (2026).

Figure 7a: Faithfulness scatter: all 147 samples (Pearson $r = 0.536$).
Source: Authors, (2026).Figure 7b: Faithfulness scatter: outlier removed ($n=146$, $r=0.288$).
Source: Authors, (2026).Figure 8a: Prediction error distribution: all 147 samples.
Source: Authors, (2026).Figure 8b: Prediction error distribution: outlier removed.
Source: Authors, (2026).

VI.4 LAYER CONTRIBUTION ANALYSIS

Table 4 and Figures 9 and 10 present layer contribution scores across five recommendation cases. Three patterns emerge. First,, indicating recommendations primarily driven by direct interaction embeddings. Second, in Layer 1-dominant cases (American Beauty, Being John Malkovich), the top attributed items are thematically related but not directly co-interacted with by the target user, confirming that 1-hop collaborative signal is captured. Third, in all cases, suggesting actionable explanations lie within 1 hop of the user for 3-layer LightGCN on MovieLens-1M.

Table 4: Propagation layer contribution scores for five representative recommendation cases.

Case	C0	C1	C2	C3	k^*	$C0^\wedge$	$C1^\wedge$	$C2^\wedge$	$C3^\wedge$
Strangers on a Train	7.84	5.28	2.54	1.58	0	0.45	0.31	0.15	0.09
American Beauty	2.38	8.24	3.67	4.60	1	0.13	0.44	0.19	0.24
The Fly	4.49	2.90	1.41	1.49	0	0.44	0.28	0.14	0.14
Being John Malkovich	6.96	7.34	2.66	3.79	1	0.34	0.35	0.13	0.18
Nightmare Before Xmas	6.32	5.06	2.92	2.26	0	0.38	0.31	0.18	0.14

Source: Authors, (2026).

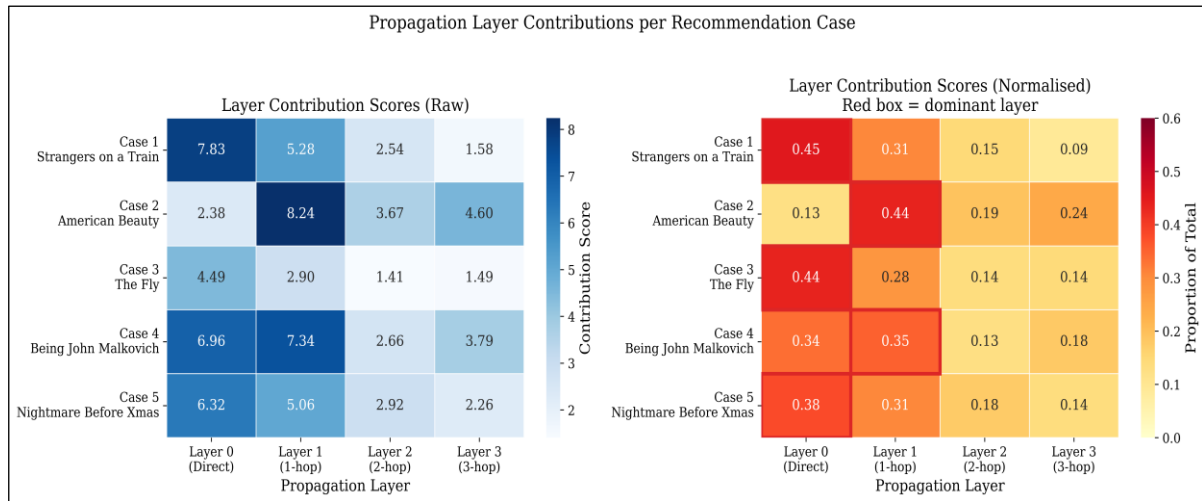


Figure 9: Heatmaps of raw (left) and normalised (right) layer contribution scores.
Source: Authors, (2026).

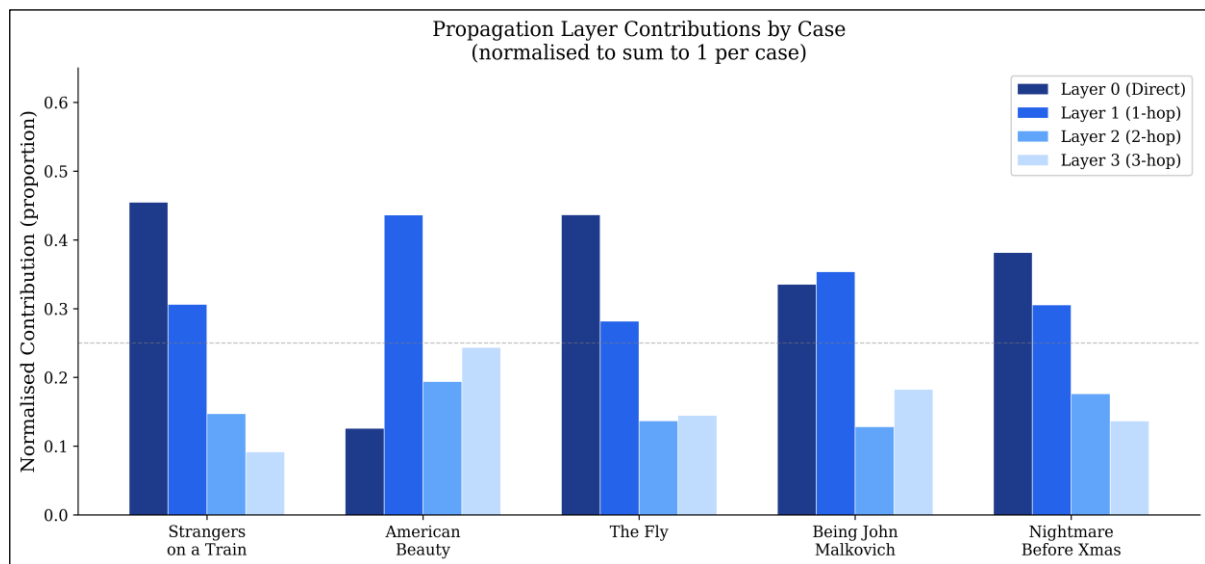


Figure 10: Grouped bar chart of normalised layer contributions per case.
Source: Authors, (2026).

VI.5 PER-USER AND CASE STUDY ANALYSIS

Figure 11 presents per-user score drops for PropPath and LIME across 10 representative users. Figure 12 presents the case study for User 5439. The top attributed item (A Family Thing, 1996) is a warm family comedy-drama — semantically coherent with Babe's genre profile. It is ranked first by both PropPath and LIME, providing cross-method corroboration. The score drop of 0.094 is the largest single-edge drop in our evaluation, confirming the attribution's quantitative significance.

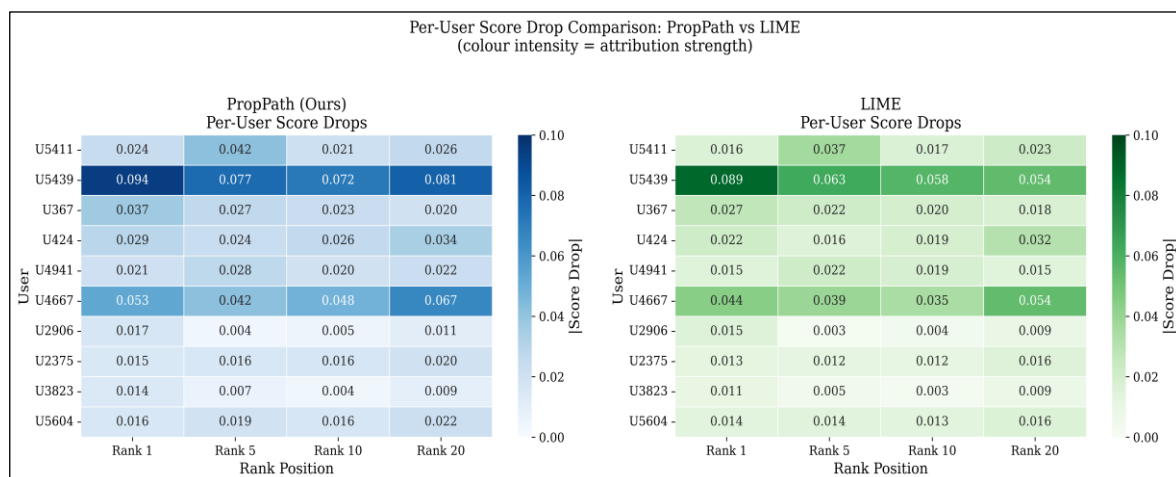


Figure 11: Per-user score drop heatmaps for PropPath (blue) and LIME (green) across 10 users.
Source: Authors, (2026).

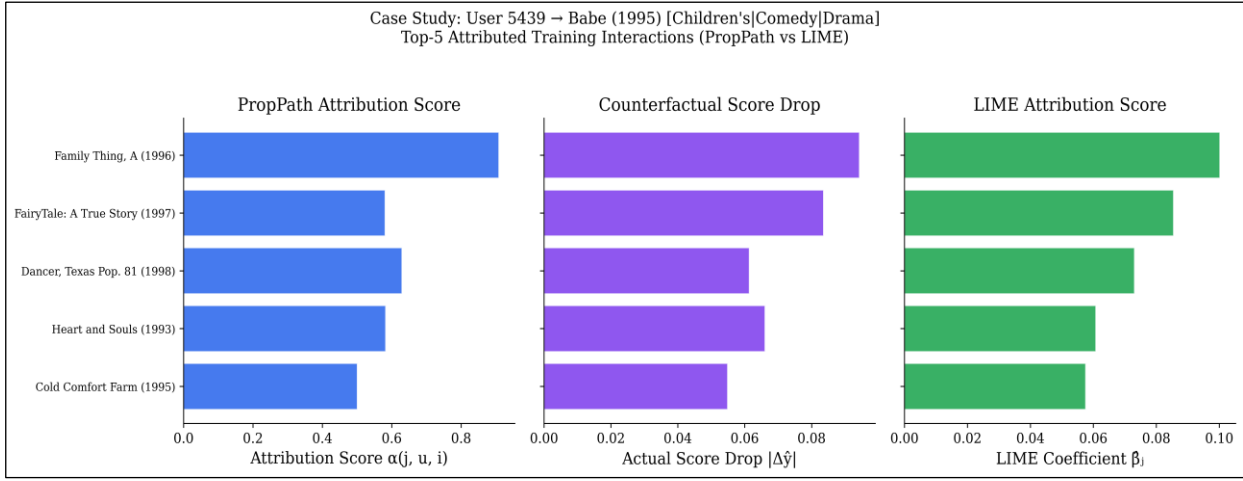


Figure 12: Case study for User 5439 → Babe (1995). PropPath and LIME both rank A Family Thing (1996) first with score drop 0.094. Source: Authors, (2026).

VII. CONCLUSION

We presented a post-hoc explainability framework for LightGCN combining Propagation Path Analysis with Counterfactual Graph Editing. The framework provides explanations at two levels: the layer level (which propagation hop drives the recommendation) and the edge level (which training interactions are most influential). Quantitative evaluation on MovieLens-1M demonstrates $3.02\times$ lift over random, $1.26\times$ over LIME, and $5.35\times$ over degree-based attribution. Faithfulness evaluation yields Pearson $r=0.536$ against post-fine-tuning ground truth. Layer contribution analysis reveals that actionable explanations lie predominantly within 1 hop of the user, with Layers 2–3 contributing below 25% in all evaluated cases.

Limitations: Sign agreement at 51% indicates structural sensitivity is better suited to influence ranking than directional prediction. The framework is specific to LightGCN and does not directly generalise to attention-based or non-linear GNN architectures.

Future Work: Promising directions include extending to attention-based GNNs, developing user-facing explanation templates, applying the framework to fairness auditing [2], and scaling faithfulness evaluation to larger user samples.

VIII. AUTHOR'S CONTRIBUTION

Conceptualization: Pranav Nallari, Dr. Jayakumar K.
Methodology: Pranav Nallari, Dr. Jayakumar K.
Investigation: Pranav Nallari, Dr. Jayakumar K.
Discussion of results: Pranav Nallari, Dr. Jayakumar K.
Writing – Original Draft: Pranav Nallari.
Writing – Review and Editing: Pranav Nallari, Dr. Jayakumar K.
Resources: Dr. Jayakumar K.
Supervision: Dr. Jayakumar K.
Approval of the final text: Pranav Nallari, Dr. Jayakumar K.

IX. ACKNOWLEDGMENTS

The authors thank the School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India, for support in conducting this research.

X. REFERENCES

- [1] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In Proc. SIGIR. ACM.
- [2] Ekstrand, M. D., et al. (2022). Fairness and discrimination in recommendation and retrieval. Found. Trends Inf. Retr.
- [3] Zhang, Y., & Chen, X. (2020). Explainable recommendation: A survey and new perspectives. Found. Trends Inf. Retr., 14(1), 1-101.
- [4] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In Proc. KDD. ACM.
- [5] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Advances in NeurIPS.
- [6] Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural graph collaborative filtering. In Proc. SIGIR.
- [7] Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). BPR: Bayesian personalized ranking from implicit feedback. arXiv:1205.2618.
- [8] Zheng, L., Lu, C.-T., Jiang, F., Zhang, J., & Yu, P. S. (2018). Spectral collaborative filtering. In Proc. RecSys.
- [9] Velickovic, P., et al. (2018). Graph attention networks. In Proc. ICLR.

- [10] Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1).
- [11] Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating explanations for graph neural networks. In *Advances in NeurIPS*.
- [12] Lucic, A., et al. (2022). CF-GNNExplainer: Counterfactual explanations for graph neural networks. In *Proc. AISTATS*.
- [13] Luo, D., et al. (2020). Parameterized explainer for graph neural network. In *Advances in NeurIPS*.
- [14] Yuan, H., et al. (2021). On explainability of graph neural networks via subgraph explorations. In *Proc. ICML*.
- [15] Chen, X., et al. (2019). Collaborative memory network for recommendation systems. In *Proc. SIGIR*.
- [16] Jain, S., & Wallace, B. C. (2019). Attention is not Explanation. In *Proc. NAACL*.
- [17] Guidotti, R., et al. (2018). A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5).
- [18] Ghazimatin, A., et al. (2020). PRINCE: Provider-side interpretability with counterfactual explanations in recommender systems. In *Proc. WSDM*.
- [19] Tran, K., et al. (2021). Counterfactual explanations for neural recommenders. In *Proc. SIGIR*.
- [20] Harper, F. M., & Konstan, J. A. (2015). The MovieLens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4).