# Can Explanations Improve Recommendations? A Joint Optimization with LLM Reasoning

**Yuyan Wang***
Stanford University
Stanford, CA 94305
yuyanw@stanford.edu

**Pan Li**
Georgia Institute of Technology
Atlanta, GA 30332
pan.li@scheller.gatech.edu

**Minmin Chen**
Google DeepMind
Mountain View, CA 94043
minminc@google.com

## Abstract

Explanations in recommender systems have long been valued in business applications for enabling consumers to make informed decisions and providing firms with actionable insights. However, existing approaches either explain the data, which is not directly tied to the model, or explain a trained model in an ad hoc manner. Neither demonstrates that explanations can improve recommendation performance. Recent advances in large language models (LLMs) show that reasoning can improve LLM performance, but LLM-based recommenders still *underperform* deep neural networks (DNNs). We propose **RecPIE**, *Recommendation with Prediction-Informed Explanations*, a framework that jointly optimizes recommendations and explanations by learning what LLM-generated explanations are most useful for recommendations. For **prediction-informed explanations**, the recommendation task guides the learning of consumer embeddings, which serve as soft prompts to fine-tune LLMs to generate contrastive explanations (why a consumer may or may not like a product). For **explanation-informed predictions**, these learned explanations are then fed back into the recommendation component to improve predictive accuracy. The two tasks are trained in an alternating fashion, with the LLM continuously fine-tuned via proximal policy optimization (PPO). Extensive experiments on multiple industrial datasets show that RecPIE significantly outperforms strong baselines, achieving 3–34% gains in predictive performance. Further analysis reveals that these gains mainly come from LLMs' *reasoning* capabilities, rather than their external knowledge or summarization skills.

## 1 Introduction

Recommender systems are integral to many businesses [Ricci et al., 2010, Schafer et al., 1999], but state-of-the-art models are often black boxes that predict user behavior without explaining why items are recommended. In an era of information overload, transparency has become increasingly important: users want to know why certain products are suggested, sellers and creators seek insights into user preferences, and platforms need to understand which product attributes attract specific segments. This demand has driven growing interest in explanations for recommender systems [McSherry, 2005, Tintarev and Masthoff, 2007, Zhang et al., 2020].

Existing explanation methods fall into two categories: (1) generating insights from the data itself, e.g., using LLMs or multimodal models to better represent consumers or content [Wang et al., 2024, Acharya et al., 2023, Wang et al., 2025], or (2) explaining an already-trained model with ad-hoc methods such as SHAP [Lundberg, 2017]. The former describe only the data rather than the model [Lubos et al., 2024], while the latter do not improve predictive performance. Thus, current approaches fail to show how explanations can make recommender systems *better*. Meanwhile, recent work shows

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

that reasoning can substantially improve LLM performance [Wei et al., 2022, Guo et al., 2025], including in LLM-based recommenders [Bao et al., 2023, Zhao et al., 2024, Tsai et al., 2024]. Yet LLM recommenders still underperform deep neural networks (DNNs), since recommendation is a discriminative task while LLMs are generative [Ye et al., 2024]. Companies therefore continue to rely on DNN-based models [Zhai et al., 2024, Covington et al., 2016], and it remains unclear what natural language explanations can actually help.

We propose **RecPIE—Recommendation with Prediction-Informed Explanations**, a framework that learns which LLM-generated explanations are most useful for recommendations. RecPIE jointly optimizes two components: *prediction-informed explanations*, where recommendation tasks guide consumer embeddings used as soft prompts to fine-tune LLMs to generate contrastive, personalized explanations; and *explanation-informed predictions*, where these explanations feed back into the recommender to improve accuracy. The two tasks are trained in an alternating loop, with the LLM fine-tuned using proximal policy optimization (PPO) [Schulman et al., 2017].

To evaluate RecPIE, we conducted experiments on four real-world datasets covering destinations, movies, restaurants, and hotels. Across all datasets, RecPIE consistently outperformed state-of-the-art black-box, LLM-based, and explainable recommenders, achieving 3–34% gains over the best baselines. These results show that RecPIE improving predictive accuracy while also providing explanations, overcoming the common trade-off between accuracy and interpretability in recommender systems.

We further isolate the source of improvement through diagnostic studies. Using GPT-2 [Radford et al., 2019], trained before our datasets were created, RecPIE still outperforms state-of-the-art baselines, confirming that gains are not due to prior knowledge. Models with stronger reasoning (e.g., Llama 3.1 [Meta, 2024]) outperform those with weaker reasoning (e.g., Mixtral 8×7b [Jiang et al., 2024]), and replacing reasoning prompts with summarization prompts degrades performance. Profile augmentation with LLMs yields little benefit. Together, these findings show that the improvements stem from LLMs' reasoning ability to identify key decision variables, rather than external knowledge or summarization.

## 2 Proposed Framework: RecPIE

RecPIE consists of two components: an explanation generator (a generative task) and a recommendation model (a discriminative task). These components are trained in an alternating optimization loop so that each task improves the other. In the *prediction-informed explanations* stage, the recommendation task guides consumer embeddings that are used as soft prompts to fine-tune an LLM for generating personalized explanations $e$. In the *explanation-informed predictions* stage, these explanations are fed back into the recommender to improve rating prediction accuracy. Formally, the ground truth rating is $y$, and the prediction with explanations is $f(x, e)$, where $f$ is the DNN-based recommendation model and $x$ denotes other standard inputs (user, item, and contextual features). The LLM is fine-tuned via proximal policy optimization (PPO) [Schulman et al., 2017]. Figure 1 provides an overview.

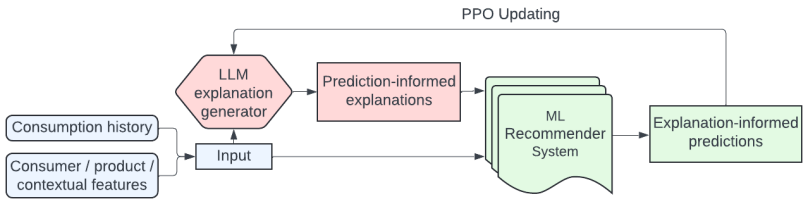



Figure 1: (Color online) Overview of RecPIE.

### 2.1 Prediction-Informed Explanations

We fine-tune an LLM to generate explanations without relying on costly ground-truth annotations. Instead, the recommendation task itself provides supervision. Given consumer $i$'s history of consumed

products $j_{i1}, \ldots, j_{in}$, the LLM is prompted to generate *positive* (why the user may like a product) and *negative* (why not) explanations. The quality of these explanations is measured by how much they reduce the rating prediction error. Specifically, we define the reward for fine-tuning the LLM as

$$r(e) = \frac{1}{1 + |y - f(x, e)|}, \tag{1}$$

where $f(x, e)$ is the recommendation output with explanation $e$. This reward is used to update the LLM through PPO.

To personalize explanations, we adopt a *soft prompt* mechanism. Each consumer $i$ has an embedding $e_i$ from the consumer embedding table $E_c$. This embedding is prepended to the LLM input, ensuring that the generated explanations reflect consumer-specific preferences. The "Explanation component" of RecPIE is illustrated in Figure 2. Prompts used for fine-tuning are detailed in Appendix A.1.

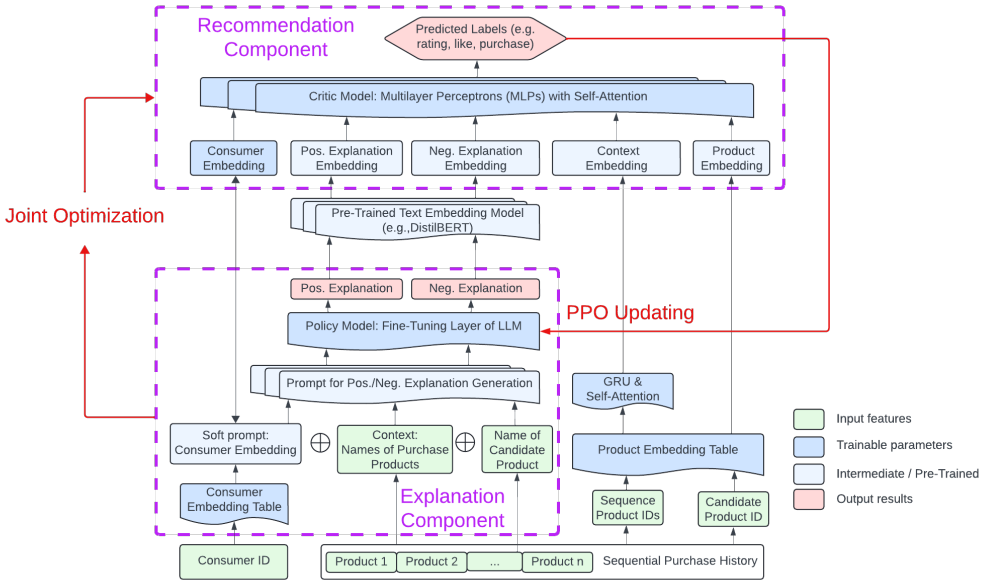



Figure 2: (Color online) Detailed architecture of RecPIE.

### 2.2 Explanation-Informed Predictions

The generated explanations are embedded using a pre-trained text encoder (e.g., DistillBERT; Sanh et al., 2019) and concatenated with standard user, item, and context features as inputs to the recommendation model. Specifically, the DNN input includes: embeddings of positive and negative explanations, the consumer embedding, product embedding, sequential history embedding, and contextual features. This input is processed by the DNN to predict ratings or outcomes such as click-through probability. Importantly, the consumer embeddings learned here are reused as soft prompts in the explanation component, forming the alternating loop between prediction and explanation. The "Recommendation component" is illustrated in Figure 2.

Statistical insights into why LLM-generated explanations can aid learning are provided in Appendix B.

## 3 Experiments

### 3.1 Setup

We evaluate RecPIE on four benchmark datasets: *Amazon Movie*, *Yelp Restaurant*, *TripAdvisor Hotel*, and *Google Maps*. Dataset details are provided in Appendix C.1. For all datasets, the task is to predict the item rating given the user's past consumption history. We compare RecPIE with 17 state-of-the-art baselines spanning four categories: LLM-based recommender systems, aspect-based recommender

| | TripAdvisor | | Yelp | | Amazon Movie | | Google Map | |
|---|---|---|---|---|---|---|---|---|
| | RMSE ↓ | MAE ↓ | RMSE ↓ | MAE ↓ | RMSE ↓ | MAE ↓ | RMSE ↓ | MAE ↓ |
| **RecPIE (Ours)** | **0.1733** (0.0010) | **0.1333** (0.0008) | **0.2020** (0.0010) | **0.1620** (0.0009) | **0.1653** (0.0010) | **0.1165** (0.0009) | **0.2976** (0.0011) | **0.1976** (0.0010) |
| **% Improved** | +13.18%*** | +21.63%*** | +16.63%*** | +21.78%*** | +21.25%*** | +34.18%*** | +4.31%*** | +3.80%*** |
| RecSAVER | 0.1949 (0.0017) | 0.1512 (0.0013) | 0.2279 (0.0017) | 0.1764 (0.0014) | 0.1755 (0.0016) | 0.1242 (0.0013) | 0.3173 (0.0015) | 0.2104 (0.0012) |
| LLMRG | 0.1956 (0.0017) | 0.1514 (0.0013) | 0.2268 (0.0017) | 0.1760 (0.0014) | 0.1758 (0.0016) | 0.1246 (0.0013) | 0.3177 (0.0015) | 0.2101 (0.0012) |
| TallRec | 0.1924 (0.0017) | 0.1498 (0.0013) | 0.2238 (0.0017) | 0.1732 (0.0014) | 0.1720 (0.0016) | 0.1222 (0.0013) | 0.3118 (0.0015) | 0.2072 (0.0012) |
| A3NCF | 0.2103 (0.0019) | 0.1811 (0.0013) | 0.2607 (0.0023) | 0.2181 (0.0016) | 0.2241 (0.0029) | 0.1903 (0.0018) | 0.3238 (0.0035) | 0.2124 (0.0021) |
| SULM | 0.2191 (0.0021) | 0.1872 (0.0013) | 0.2823 (0.0019) | 0.2258 (0.0015) | 0.2477 (0.0027) | 0.1980 (0.0019) | 0.3244 (0.0034) | 0.2150 (0.0021) |
| AARM | 0.2083 (0.0019) | 0.1803 (0.0014) | 0.2582 (0.0021) | 0.2162 (0.0015) | 0.2162 (0.0027) | 0.1845 (0.0018) | 0.3238 (0.0035) | 0.2124 (0.0021) |
| SASRec | 0.2089 (0.0007) | 0.1731 (0.0006) | 0.2491 (0.0011) | 0.2135 (0.0009) | 0.2176 (0.0013) | 0.1869 (0.0008) | 0.3118 (0.0011) | 0.2058 (0.0010) |
| DIN | 0.2022 (0.0009) | 0.1709 (0.0007) | 0.2479 (0.0009) | 0.2116 (0.0008) | 0.2155 (0.0009) | 0.1853 (0.0007) | 0.3134 (0.0011) | 0.2070 (0.0010) |
| BERT4Rec | 0.2003 (0.0009) | 0.1701 (0.0006) | 0.2460 (0.0009) | 0.2101 (0.0008) | 0.2126 (0.0009) | 0.1832 (0.0008) | 0.3110 (0.0011) | 0.2054 (0.0010) |
| PETER | 0.1996 (0.0019) | 0.1715 (0.0013) | 0.2423 (0.0015) | 0.2071 (0.0013) | 0.2099 (0.0013) | 0.1770 (0.0010) | 0.3126 (0.0021) | 0.2059 (0.0013) |
| UCEPic | 0.2035 (0.0015) | 0.1723 (0.0011) | 0.2477 (0.0015) | 0.2099 (0.0012) | 0.2228 (0.0011) | 0.1801 (0.0009) | 0.3148 (0.0021) | 0.2078 (0.0014) |
| PARSRec | 0.2008 (0.0009) | 0.1703 (0.0007) | 0.2471 (0.0009) | 0.2106 (0.0008) | 0.2133 (0.0009) | 0.1837 (0.0007) | 0.3166 (0.0021) | 0.2089 (0.0014) |

Table 1: Recommendation performance on four datasets. "%Improved" shows the gains of RecPIE over the best baseline (underlined). Lower is better for RMSE and MAE. ***p<0.01; **p<0.05.

| | TripAdvisor | | | Yelp | | | Amazon Movie | | | Google Map | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Coverage | Informativeness | Fluency | Coverage | Informativeness | Fluency | Coverage | Informativeness | Fluency | Coverage | Informativeness | Fluency |
| **RecPIE** | **0.3784** | **0.5744** | **0.4193** | **0.3984** | **0.5936** | **0.4380** | **0.3580** | **0.5276** | **0.4003** | **0.3138** | **0.4997** | **0.3836** |
| TallRec | 0.3668 | 0.5583 | 0.4158 | 0.3958 | 0.5901 | 0.4358 | 0.3479 | 0.5155 | 0.3530 | 0.3099 | 0.4930 | 0.3818 |
| PETER | 0.3730 | 0.4762 | 0.3772 | 0.3770 | 0.5327 | 0.4012 | 0.3026 | 0.4748 | 0.3264 | 0.3074 | 0.4525 | 0.3597 |

Table 2: Explanation quality comparison.

systems, sequential recommender systems, and interpretable recommender systems (summarized in Appendix C.2).

### 3.2 Main Results

Table 1 reports the main results. RecPIE consistently outperforms all baseline groups across the four datasets. In particular, LR-Recsys achieves 2.9–3.7% improvements in AUC over the strongest baselines, demonstrating both the efficacy of RecPIE and the value of LLM-based contrastive explanations for improving recommendation accuracy.

Appendix C.4 present an example explanation generated with RecPIE alongside corresponding recommendations. Table 2 further compares explanation quality against the best explanation-generation baselines (evaluation metrics detailed in Appendix C.3). RecPIE consistently produces the highest-quality explanations, confirming the advantage of jointly optimizing recommendation and explanation in a single framework.

### 3.3 Understanding the Improvements

In Appendix D, we provide additional analyses to better understand the sources of RecPIE's performance gains. Specifically, we show that: (1) RecPIE substantially improves learning efficiency, achieving the predictive accuracy of the best baseline trained on 100% of the data using only 25% of the training data (Appendix D.1); (2) users and products with the highest prediction uncertainty benefit the most, alleviating cold-start challenges (Appendix D.2); (3) the improvements mainly come from LLMs' *reasoning* capability, rather than external knowledge or summarization (Appendix D.3); (4) RecPIE learns item similarities in the embedding space with far fewer examples than state-of-the-art recommenders (Appendix D.4); and (5) both positive and negative explanations contribute to predictive performance (Appendix D.5).

## 4 Conclusion

We proposed RecPIE, a framework that jointly optimizes recommendations and explanations. Unlike prior approaches that treat recommendation (a discriminative task) and explanation (a generative task) separately, RecPIE demonstrates that the two can mutually reinforce each other when trained together. The framework is general and practical for large-scale deployment: unlike LLM-based recommenders that are expensive to serve in real time, RecPIE allows explanations to be pre-computed and stored for each user. Beyond recommender systems, the idea of integrating prediction-informed explanations

with explanation-informed predictions offers a promising direction for broader applications such as consumer targeting and behavior prediction.

## References


Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2010.

J Ben Schafer, Joseph Konstan, and John Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*, pages 158–166, 1999.

David McSherry. Explanation in recommender systems. *Artificial Intelligence Review*, 24(2):179–197, 2005.

Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *2007 IEEE 23rd international conference on data engineering workshop*, pages 801–810. IEEE, 2007.

Yongfeng Zhang, Xu Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.

Jianling Wang, Haokai Lu, Yifan Liu, He Ma, Yueqi Wang, Yang Gu, Shuzhou Zhang, Ningren Han, Shuchao Bi, Lexi Baugher, et al. Llms for user interest exploration in large-scale recommendation systems. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pages 872–877, 2024.

Arkadeep Acharya, Brijraj Singh, and Naoyuki Onoe. Llm based generation of item-description for recommendation system. In *Proceedings of the 17th ACM conference on recommender systems*, pages 1204–1207, 2023.

Haoting Wang, Jianling Wang, Hao Li, Fangjun Yi, Mengyu Fu, Youwei Zhang, Yifan Liu, Liang Liu, Minmin Chen, Ed H Chi, et al. Serendipitous recommendation with multimodal llm. *arXiv preprint arXiv:2506.08283*, 2025.

Scott Lundberg. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

Sebastian Lubos, Thi Ngoc Trang Tran, Alexander Felfernig, Seda Polat Erdeniz, and Viet-Man Le. Llm-generated explanations for recommender systems. In *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pages 276–285, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, pages 1007–1014, 2023.

Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, et al. Recommender systems in the era of large language models (llms). *IEEE Transactions on Knowledge and Data Engineering*, 36(11):6889–6907, 2024.

Alicia Y Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang, Lichan Hong, Ed H Chi, and Xinyang Yi. Leveraging llm reasoning enhances personalized recommender systems. *arXiv preprint arXiv:2408.00802*, 2024.

Zikun Ye, Hema Yoganarasimhan, and Yufeng Zheng. Lola: Llm-assisted online learning algorithm for content experiments. *arXiv preprint arXiv:2406.02611*, 2024.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152*, 2024.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Recsys'16*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

AI Meta. Introducing llama 3.1: Our most capable models to date. *Meta AI Blog*, 2024.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78(5):947–1012, 2016.

Jianqing Fan, Cong Fang, Yihong Gu, and Tong Zhang. Environment invariant linear least squares. *arXiv preprint arXiv:2303.03092*, 2023.

Yihong Gu, Cong Fang, Peter Bühlmann, and Jianqing Fan. Causality pursuit from heterogeneous environments via neural adversarial invariance learning. *arXiv preprint arXiv:2405.04715*, 2024.

Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, 48(4):1875–1897, 2020. doi: 10.1214/19-AOS1875. URL `https://projecteuclid.org/euclid.aos/1598955880`.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197, 2019.

Lei Li, Yongfeng Zhang, and Li Chen. Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, 41(4):1–26, 2023a.

Yan Wang, Zhixuan Chu, Xin Ouyang, Simeng Wang, Hongyan Hao, Yue Shen, Jinjie Gu, Siqiao Xue, James Y Zhang, Qing Cui, et al. Enhancing recommender systems with large language model reasoning graphs. *arXiv preprint arXiv:2308.10835*, 2023.

Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan S Kankanhalli. Aˆ 3ncf: An adaptive aspect attention model for rating prediction. In *IJCAI*, pages 3748–3754, 2018.

Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 717–725, 2017.

Xinyu Guan, Zhiyong Cheng, Xiangnan He, Yongfeng Zhang, Zhibo Zhu, Qinke Peng, and Tat-Seng Chua. Attentive aspect modeling for review-aware recommendation. *ACM Transactions on Information Systems (TOIS)*, 37(3):1–27, 2019.

Zhiyong Cheng, Xiaojun Chang, Lei Zhu, Rose C Kanjirathinkal, and Mohan Kankanhalli. Mmalfm: Explainable recommendation by leveraging reviews and images. *ACM Transactions on Information Systems (TOIS)*, 37(2):1–28, 2019.

Jin Yao Chin, Kaiqi Zhao, Shafiq Joty, and Gao Cong. Anr: Aspect-based neural recommender. In *Proceedings of the 27th ACM International conference on information and knowledge management*, pages 147–156, 2018.

Trung-Hoang Le and Hady W Lauw. Explainable recommendation with comparative constraints on product aspects. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 967–975, 2021.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *ICDM'18*, pages 197–206. IEEE, 2018.

Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1059–1068, 2018.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM'19*, pages 1441–1450, 2019.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593, 2022.

Deng Pan, Xiangrui Li, Xin Li, and Dongxiao Zhu. Explainable recommendation via interpretable feature mapping and evaluation of explainability. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2690–2696, 2021.

Lei Li, Yongfeng Zhang, and Li Chen. Personalized transformer for explainable recommendation. *arXiv preprint arXiv:2105.11601*, 2021.

Jiacheng Li, Zhankui He, Jingbo Shang, and Julian McAuley. Ucepic: Unifying aspect planning and lexical constraints for generating explanations in recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1248–1257, 2023b.

Ehsan Gholami, Mohammad Motamedi, and Ashwin Aravindakshan. Parsrec: Explainable personalized attention-fused recurrent sequential recommendation using session partial actions. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 454–464, 2022.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. Bartscore: Evaluating generated text as text generation. *Advances in neural information processing systems*, 34:27263–27277, 2021.

S Patro. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

# A Details of the framework

## A.1 Prompts

The prompts for positive and negative explanations follow this template:

> *"Provide a reason for why this user purchased (or did not purchase) this product, based on the provided profile of the past products the user purchased, and the profile of the current product. Answer with exactly one sentence in the following format: 'The user purchased (or did not purchase) this product because the user ... and the product ...'."*

The prompts can be adapted to fit specific domains; for example, "purchased this product" can be changed to "watched this movie". The prompt is then concatenated with the user's consumption history, which is presented as a sequence of product profiles, and the candidate product profile.

## B Statistical Insights into Why Explanations Improve Learning

We provide mathematical insights into why and how explanations can help machine learning (ML) models learn more efficiently. In a typical ML setting, the model is trained in a single *environment*, represented by the observed training data. The goal is to predict outcomes $Y$ from high-dimensional inputs $X$, which may include user behavior, product attributes, and contextual signals. When we prompt large language models (LLMs) to explain why a user prefers or dislikes a product, we ask them to reveal *probable decision-making factors*—that is, which features in $X$ likely drive $Y$. Incorporating this knowledge into model training improves learning efficiency and predictive performance with the same amount of data.

We formalize this using high-dimensional statistical learning theory. Suppose observations $\{\boldsymbol{x}_k, y_k\}_{k=1}^N$ follow:

$$y_k = f(\boldsymbol{x}_k) + \epsilon_k, \tag{2}$$

where $\boldsymbol{x}_k = (x_{k1}, ..., x_{kp})$ is a $p$-dimensional feature vector, $y_k$ is the label, $\epsilon_k$ are i.i.d. errors, and $f$ is the true data-generating function. Let $S^* \subset \{1, ..., p\}$ be the (unknown) set of important features, with $|S^*| = s^* \ll p$. Then:

$$y_k = f^*(\boldsymbol{x}_{k,S^*}) + \epsilon_k. \tag{3}$$

LLM-generated explanations can be interpreted as direct estimates of $S^*$. For instance, in Figure **??**, the LLM highlights features like `hasBoughtOrange` as relevant to the user liking orange juice. This implicit identification of $S^*$ enables more targeted and efficient learning.

**LLMs Approximate $S^*$ via Multi-Environment Learning.** LLMs are trained on data from many domains—analogous to *multi-environment learning* [Peters et al., 2016]. Let $\mathcal{E}$ be a set of environments. Each $e \in \mathcal{E}$ provides $n$ i.i.d. samples $\{\boldsymbol{x}_k^{(e)}, y_k^{(e)}\}_{k=1}^n$ from the model:

$$y_k^{(e)} = {\boldsymbol{\beta}_{S^*}^*}^T \boldsymbol{x}_{k,S^*}^{(e)} + \epsilon_k^{(e)}. \tag{4}$$

The feature relevance (i.e., $S^*$ and $\boldsymbol{\beta}^*$) is invariant across environments, but feature distributions may vary. LLMs trained on web-scale corpora effectively learn from large $|\mathcal{E}|$, enabling robust inference of $S^*$.

The *environment invariant linear least squares* (EILLS) estimator [Fan et al., 2023] formalizes this. It solves:

$$\hat{\boldsymbol{\beta}}_L = \arg\min_{\boldsymbol{\beta}} \hat{R}(\boldsymbol{\beta}) + \gamma \hat{J}(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_0, \tag{5}$$

where $\hat{R}(\boldsymbol{\beta})$ is empirical loss across environments and $\hat{J}(\boldsymbol{\beta})$ encourages cross-environment invariance. Under regularity conditions:

$$\mathbf{P}(\text{supp}(\hat{\boldsymbol{\beta}}_L) = S^*) \to 1 \quad \text{as } n \gg s^* \beta_{\min}^{-2} \log p. \tag{6}$$

Thus, LLMs—trained on vast, multi-environment corpora—can identify $S^*$ with high probability, outperforming models trained in a single environment.

**Knowing $S^*$ Improves Learning Efficiency.** Let $\boldsymbol{\beta}^*$ be the true coefficients. Without knowledge of $S^*$, Lasso regression estimates:

$$\hat{\boldsymbol{\beta}}_{\text{Lasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2n} \sum_{k=1}^{n} (y_k - \boldsymbol{x}_k^T \boldsymbol{\beta})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}, \tag{7}$$

with error rate:

$$\|\hat{\boldsymbol{\beta}}_{\text{Lasso}} - \boldsymbol{\beta}^*\|_2 = O_P\left(\sqrt{\frac{s^* \log p}{n}}\right). \tag{8}$$

If $S^*$ is known (oracle case), OLS yields:

$$\|\hat{\boldsymbol{\beta}}_{\text{Oracle}} - \boldsymbol{\beta}^*\|_2 = O_P\left(\sqrt{\frac{s^*}{n}}\right), \tag{9}$$

and achieves better convergence by a factor of $\sqrt{\log p}$. This efficiency gain is particularly valuable in recommender systems, where $p$ (e.g., item attributes) can be very large.

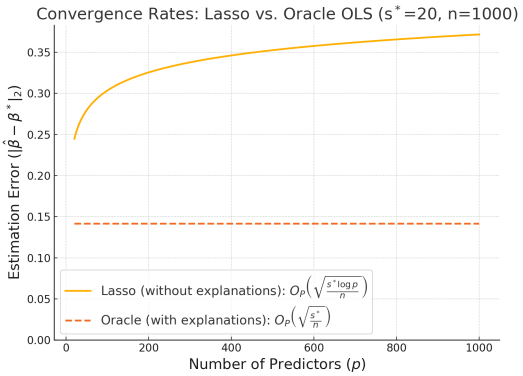



Figure 3: (Color online) Convergence rate comparison: Lasso (unknown $S^*$) vs. Oracle (known $S^*$).

**Extensions to Nonlinear Models.** Though the above theory focuses on linear models, results extend to nonlinear settings. Gu et al. [2024] consider multi-environment neural networks:

$$\min_{g \in \mathcal{G}} \max_{f^{(e)} \in \mathcal{F}_g} \sum_{e \in \mathcal{E}} \mathbb{E}_{\mu^{(e)}} [\ell(y, g(x))] + \gamma \sum_{e \in \mathcal{E}} \mathbb{E}[\{y - g(x)\} f^{(e)}(x) - f^{(e)}(x)^2/2]. \tag{10}$$

They show that the estimated set $\hat{S}$ includes all true features with high probability.

Finally, Schmidt-Hieber [2020] show that for neural networks, convergence rate improves from $O_P(n^{-2/(2+p)})$ to $O_P(n^{-2/(2+s^*)})$ if $S^*$ is known. Thus, LLM-generated explanations—by identifying $S^*$—help black-box ML models converge faster and generalize better.

In practice, our framework retains environment-specific signals (e.g., embeddings) alongside explanations, allowing flexibility while preserving the benefits of explanation-informed learning.

# C Details on the Experiments

## C.1 Datasets

**Amazon Movie**: This dataset captures user purchasing behavior in the Movies & TV category on Amazon [Ni et al., 2019][2]. Collected in 2023, it contains 17,328,314 records from 6,503,429 users and 747,910 unique movies. Each record includes the user ID, movie ID, movie title, user rating (on a 1-5 scale), purchasing timestamp, user's past purchasing history (as a sequence of movie IDs), and three aspect terms summarizing key movie attributes (e.g., "thriller", "exciting", "director").

**Yelp Restaurant**: This dataset documents users' restaurant check-ins on the Yelp platform[3]. It spans 11 metropolitan areas in the United States and comprises 6,990,280 check-in records from 1,987,897 users across 150,346 restaurants. Each record includes the user ID, restaurant ID, check-in timestamp, user rating (on a 1-5 scale), user's historic visits (as a sequence of restaurant IDs), and three aspect terms summarizing key restaurant features (e.g., "atmosphere", "service", "expensive").

**TripAdvisor Hotel**: This dataset captures users' hotel stays on the TripAdvisor platform [Li et al., 2023a]. Collected in 2019, it contains 343,277 hotel stay records from 9,765 users and 6,280 unique hotels. Each record includes the user ID, hotel ID, check-in timestamp, user rating, the list of hotels previously visited by the user, and three aspect terms describing key hotel attributes (e.g., "beach", "price", "service").

**Google Map Mountain View**: This dataset contains review information up to Sep 2021 on Google Map (ratings and review text), along with business metadata (address, descriptions, category, etc...) in the United States [4]. We focus on the sample of the data around the Google Headquarters (i.e., Mountain View, CA) by filtering with the zip codes between 94035 and 94043. To align with the

[2] https://nijianmo.github.io/amazon/index.html

[3] https://www.yelp.com/dataset

[4] https://mcauleylab.ucsd.edu/public_datasets/gdrive/googlelocal/

other three datasets, we infer three aspect terms describing key business attributes using LLM based on the review text and business descriptions.

### C.2 Baselines

We compare RecPIE against state-of-the-art black-box recommender systems, LLM-based recommender systems, and a wide range of explainable recommender systems. Specifically, we identify the following four groups of 14 state-of-the-art baselines:

1. **LLM-Based Recommender Systems**:
    - **RecSAVER** [Tsai et al., 2024] automatically assesses the quality of LLM reasoning responses through a self-verification process, without the requirement of curated gold references or human raters. The identified reasoning will then be used for the rating prediction task.
    - **LLMRG** [Wang et al., 2023] leverages LLMs to construct personalized reasoning graphs, which link a user's profile and behavioral sequences through causal and logical inferences, representing the user's interests in an interpretable way.
    - **TallRec** [Bao et al., 2023] fine-tunes LLMs with a large set of recommendation data to align LLMs with the recommendation tasks.
2. **Aspect-Based Recommender Systems**: These models utilize ground-truth aspect terms as additional information to facilitate preference reasoning and generate recommendations.
    - **A3NCF** [Cheng et al., 2018] constructs a topic model to extract user preferences and item characteristics from reviews, and capture user attention on specific item aspects via an attention network.
    - **SULM** [Bauman et al., 2017] predicts the sentiment of a user about an item's aspects, identifies the most valuable aspects of their potential experience, and recommends items based on these aspects.
    - **AARM** [Guan et al., 2019] models interactions between similar aspects to enrich aspect connections between users and products, using an attention network to focus on aspect-level importance.
    - **MMALFM** [Cheng et al., 2019] applies a multi-modal aspect-aware topic model to estimate aspect importance and predict overall ratings as a weighted linear combination of aspect ratings.
    - **ANR** [Chin et al., 2018] learns aspect-based user and item representations through an attention mechanism and models multi-faceted recommendations using a neural co-attention framework.
    - **MTER** [Le and Lauw, 2021] generates aspect-level comparisons between target and reference items, producing recommendations based on these comparative explanations.
3. **Sequential Recommender Systems**: These models use sequences of past user behaviors to predict the next likely purchase, leveraging various neural network architectures.
    - **SASRec** [Kang and McAuley, 2018] utilizes self-attention to capture long-term semantics in user actions and identify relevant items in a user's history.
    - **DIN** [Zhou et al., 2018] adopts a local activation unit to adaptively learn user interest representations from historical behaviors and predict preferences for candidate items.
    - **BERT4Rec** [Sun et al., 2019] adopts bidirectional self-attention and the Cloze objective to model user behavior sequences and avoid information leakage, enhancing recommendation efficiency.
    - **UniSRec** [Hou et al., 2022] uses contrastive pre-training to learn universal sequence representations of user preferences, improving recommendation accuracy.
4. **Interpretable Recommender Systems**: These models focus on generating high-quality recommendations accompanied by intuitive explanations.
    - **AMCF** [Pan et al., 2021] maps uninterpretable general features to interpretable aspect features, optimizing for both recommendation accuracy and explanation clarity through dual-loss minimization.

- **PETER** [Li et al., 2021] predicts words in target explanations using IDs, endowing them with linguistic meaning to generate personalized recommendations.
- **UCEPic** [Li et al., 2023b] combines aspect planning and lexical constraints to produce personalized explanations through insertion-based generation, improving recommendation performance.
- **PARSRec** [Gholami et al., 2022] leverages common and individual behavior patterns via an attention mechanism to tailor recommendations and generate explanations based on these patterns.

We split each dataset into training and test sets using an 80-20 ratio at the user-temporal level. To ensure a fair comparison, we adopted Grid Search [Bergstra et al., 2011] and allocated equal computational resources—in terms of training time and memory usage—to optimize the hyperparameters for both RecPIE and all baseline models.

### C.3 Explanation quality evaluation

We look at the following three metrics for evaluating explanation quality:

- **Coverage**: Percentage of aspect terms covered in the explanations[5].
- **Informativeness**: The informativeness score (INFO) measured by BARTScore [Yuan et al., 2021], which measures how well the generated text captures the key ideas of the source text. Here the source text is the review text for each (user, item) pair.
- **Fluency**: The fluency score (FLU) measured by BARTScore [Yuan et al., 2021], which measures whether the text has no formatting problems, capitalization errors or obviously ungrammatical sentences (e.g., fragments, missing components) that make the text difficult to read.

The results are reported in Table 2.

### C.4 Case study

To illustrate how our model operates, we present a case study from the Yelp dataset below. In this scenario, the task is to recommend a premium Thai restaurant to a specific consumer. However, after incorporating the positive and negative reasoning generated by the contrastive-explanation generator in our RecPIE, the model determines that an alternative Japanese restaurant would be a better fit for the consumer. Consequently, our recommendation system predicts a rating of 1.14 for the Thai restaurant, closely aligning with the ground-truth rating of 1. This prediction significantly outperforms the baseline model PETER [Li et al., 2021], which predicts a rating of 1.89.

**Consumer Past Visiting History**: O-Ku Sushi, Zen Japanese, MGM Grand Hotel, Sen of Japan, Sushi Bong
**Restaurant Profile**: "Siam Thai Kitchen is a Thai restaurant that offers a unique dining experience in the city. The restaurant is known for its authentic Thai cuisine and its warm and inviting atmosphere. The menu features a variety of traditional Thai dishes, as well as some modern twists on classic Thai flavors. The restaurant is perfect for couples, families, and groups of friends who are looking for a delicious and authentic Thai dining experience."
**Generated Positive Explanation**: "The consumer is looking for a unique and flavorful dining experience and the restaurant offers a variety of Asian cuisine."
**Generated Negative Explanation**: "The consumer is looking for a traditional Japanese experience and wants to escape the busy city life, while the restaurant is not a traditional Japanese experience and is located in a city."
**Positive Explanation Attention Value**: 0.23
**Negative Explanation Attention Value**: 0.87
**RecPIE Predicted Consumer Rating**: 1.14
**Ground-Truth Consumer Rating**: 1
**PETER-Predicted Consumer Rating**: 1.89

[5]Each (user, item) pair is accompanied by a list of aspect terms in the dataset.

| | TripAdvisor | | Yelp | | Amazon Movie | |
|---|---|---|---|---|---|---|
| | RMSE ↓ | MAE ↓ | RMSE ↓ | MAE ↓ | RMSE ↓ | MAE ↓ |
| RecPIE | 0.1889 | 0.1444 | 0.2149 | 0.1685 | 0.1673 | 0.1180 |
| 100% Training Data | (0.0010) | (0.0008) | (0.0010) | (0.0009) | (0.0010) | (0.0009) |
| RecPIE | 0.1938 | 0.1503 | 0.2188 | 0.1774 | 0.1791 | 0.1308 |
| 50% Training Data | (0.0017) | (0.0013) | (0.0018) | (0.0017) | (0.0021) | (0.0017) |
| RecPIE | 0.2017 | 0.1679 | 0.2356 | 0.1958 | 0.1997 | 0.1703 |
| 25% Training Data | (0.0026) | (0.0021) | (0.0027) | (0.0027) | (0.0039) | (0.0028) |
| RecPIE | 0.2098 | 0.1796 | 0.2557 | 0.2140 | 0.2175 | 0.1866 |
| 12% Training Data | (0.0036) | (0.0030) | (0.0039) | (0.0038) | (0.0066) | (0.0044) |
| PETER | 0.1996 | 0.1715 | 0.2423 | 0.2071 | 0.2099 | 0.1770 |
| 100% Training Data | (0.0019) | (0.0013) | (0.0015) | (0.0013) | (0.0013) | (0.0010) |

Table 3: Recommendation performance across three datasets using varying percentages of training data for RecPIE.

# D Understanding the Improvements

## D.1 Improved learning efficiency.

Based on the theoretical insights, incorporating explanations into the recommendation process is expected to significantly improve learning efficiency. This implies that our proposed RecPIE should require *less* training data to achieve recommendation performance comparable to the baselines. To demonstrate this, we randomly sample subsets of the three datasets, keeping 12%, 25%, and 50% of the original training data, and train RecPIE on these subsets while keeping the same test set for evaluation. The results, presented in Table 3, show that our model achieves performance equivalent to the best-performing baseline, PETER [Li et al., 2021], using as little as 25% of the training data. These findings validate the improved learning efficiency of RecPIE.

## D.2 "Harder" examples benefit more from RecPIE

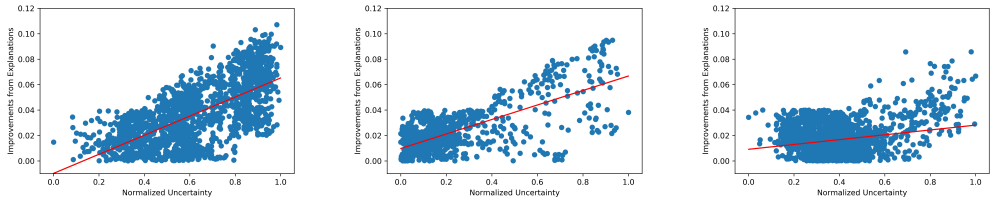
By leveraging LLMs' reasoning capabilities to identify important variables, RecPIE should intuitively provide greater benefits for "harder" examples where users' decisions are less obvious. To test this hypothesis, we compute the prediction uncertainty for each observation in our datasets, measured as the variance—or "disagreement"—across the predictions made by our model and all baseline models from Table 1. Intuitively, higher prediction uncertainty indicates a more challenging, or "harder", prediction task.

In Fig.4, we created plots for each dataset, where the x-axis represents the normalized uncertainty level (scaled between 0 and 1 using min-max normalization [Patro, 2015]), and the y-axis represents the performance improvement of our RecPIE over the best-performing baseline (measured by RMSE). As shown by the regression lines in Figures 4(a), 4(b), and 4(c), there is a statistically significant *positive correlation* between uncertainty and performance improvements. Specifically, the performance gains from incorporating explanations are consistently larger for high-uncertainty examples across all three datasets, validating the insight that our RecPIE is more beneficial for examples that are "harder" or more uncertain.

This observation aligns with the theoretical insights in Appendix B. For "harder" examples, the model is likely uncertain about which input variables to rely on for making predictions, leading to higher prediction uncertainty. In such cases, the knowledge provided by LLMs about the important variables becomes particularly valuable, allowing the model to focus on the most relevant features. Consequently, the performance gains of our RecPIE are larger for these more challenging cases.

## D.3 The gain of RecPIE is from LLM's reasoning capability

The theoretical insights highlight that the advantage of RecPIE lies in leveraging LLMs' strong *reasoning* capabilities to identify the important variables. Therefore, LLMs with better reasoning capabilities are expected to lead to better recommendation performance. To validate this, we conduct additional experiments within the RecPIE framework, using different LLMs with varying reasoning




(a) Amazon Dataset. (b) TripAdvisor Dataset. (c) Yelp Dataset.

Figure 4: Performance improvement of RecPIE against (normalized) pediction uncertainty.

capabilities. As shown in Table 4, the performance of RecPIE with Llama 3.1 is significantly better than RecPIE with Llama 3, Mixtral-8×7b, Vicuna-7b-v1.5, Qwen2-7B, or GPT-2. This aligns with the reasoning capability leaderboard at `https://huggingface.co/spaces/allenai/ZebraLogic`, where Llama 3.1 demonstrates the highest reasoning capabilities among the tested models. Furthermore, Llama 3 and Mixtral-8×7b also outperform Vicuna-7b-v1.5, Qwen2-7B, and GPT-2 in the reasoning leaderboard, which is also aligned with the results observed in RecPIE. These results confirm that better *reasoning* capabilities in LLMs directly translate to improved performance within the RecPIE framework.

Moreover, RecPIE significantly outperforms an alternative approach that uses LLMs directly for recommendations—without generating explicit explanations ("LLM Direct Recommendation (with Llama 3.1)" row in Table 4). This suggests that only using LLMs for recommendation without tapping into their reasoning abilities is insufficient. Additionally, we find that including LLM-generated product profile information plays only a minor role in the overall effectiveness of the model, as RecPIE continues to significantly outperform baseline models even when these augmented profiles are removed ("RecPIE w/o Profile Augmentation" row in Table 4).

Furthermore, we confirm that the observed performance improvements are not due to information leakage or pre-existing dataset knowledge. For example, the Amazon Movie dataset was collected in 2023, while GPT-2 was pre-trained on data available only up to 2019. Despite this, when GPT-2 is used within the RecPIE framework, our approach still outperforms other baselines.

Finally, we also tested utilizing LLMs' summarization capabilities instead of their reasoning abilities within RecPIE. Specifically, we replace the positive and negative explanation prompts in the contrastive-explanation generator with the following prompt:

> *"Given the profiles of the watching history of this user {movie_profile_seq}, can you provide a summary of the user preference of candidate movies?"*

In other words, we leverage the LLM's summarization skills to condense the user's consumption history, and then use this summarized information as input to the DNN instead of the positive and negative explanations. As shown in the last row of Table 4 ("Consumption History Summarization"), the performance of using LLM for summarization is significantly worse than that of RecPIE using LLM for explanations. This suggests that the gain from RecPIE specifically comes from the *reasons* (positive and negative explanations) provided by LLMs, rather than their ability to summarize consumption history.

Collectively, these findings support the conclusion that the performance gains observed with RecPIE are primarily driven by the LLMs' reasoning capabilities, *not* their external dataset knowledge or summarization skills.

### D.4 RecPIE learns similarity between items much faster than state-of-the-art recommender systems

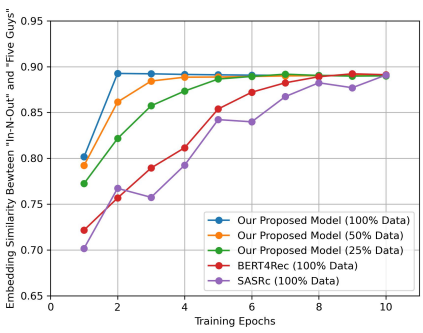
Figure 5 illustrates the embedding distances between two similar items (In-N-Out vs. Five Guys) and two dissimilar items (Mountain View Farmers Market vs. Kirin Chinese Restaurant) in the Google Maps dataset. RecPIE learns these relationships significantly faster than state-of-the-art DNN-based recommender systems: embedding distances (cosine similarities) converge quickly to a high value

| | TripAdvisor | | Yelp | | Amazon Movie | |
|---|---|---|---|---|---|---|
| | RMSE ↓ | MAE ↓ | RMSE ↓ | MAE ↓ | RMSE ↓ | MAE ↓ |
| RecPIE with **Llama 3.1** (Ours) | 0.1889 (0.0010) | 0.1444 (0.0008) | 0.2149 (0.0010) | 0.1685 (0.0009) | 0.1673 (0.0010) | 0.1180 (0.0009) |
| RecPIE with **Llama 3** | 0.1934 (0.0010) | 0.1491 (0.0008) | 0.2166 (0.0010) | 0.1697 (0.0009) | 0.1695 (0.0010) | 0.1203 (0.0009) |
| RecPIE with **Mixtral-8** ×7b | 0.1910 (0.0010) | 0.1462 (0.0008) | 0.2163 (0.0010) | 0.1693 (0.0009) | 0.1691 (0.0010) | 0.1199 (0.0009) |
| RecPIE with **Vicuna-7b-v1.5** | 0.1949 (0.0010) | 0.1502 (0.0008) | 0.2175 (0.0010) | 0.1703 (0.0009) | 0.1703 (0.0010) | 0.1210 (0.0009) |
| RecPIE with **Qwen2-7B** | 0.1966 (0.0010) | 0.1520 (0.0008) | 0.2193 (0.0010) | 0.1724 (0.0009) | 0.1727 (0.0010) | 0.1235 (0.0009) |
| RecPIE with **GPT-2** | 0.1940 (0.0014) | 0.1582 (0.0010) | 0.2211 (0.0015) | 0.1801 (0.0013) | 0.1799 (0.0015) | 0.1304 (0.0013) |
| LLM Direct Recommendation (with Llama 3.1) | 0.2233 (0.0033) | 0.1838 (0.0020) | 0.2976 (0.0044) | 0.2402 (0.0029) | 0.2680 (0.0046) | 0.2055 (0.0031) |
| RecPIE w/o Profile Augmentation | 0.1973 (0.0013) | 0.1520 (0.0011) | 0.2193 (0.0012) | 0.1719 (0.0012) | 0.1744 (0.0013) | 0.1239 (0.0012) |
| Consumption History Summarization | 0.1971 (0.0011) | 0.1700 (0.0008) | 0.2409 (0.0011) | 0.2051 (0.0009) | 0.2077 (0.0011) | 0.1751 (0.0010) |

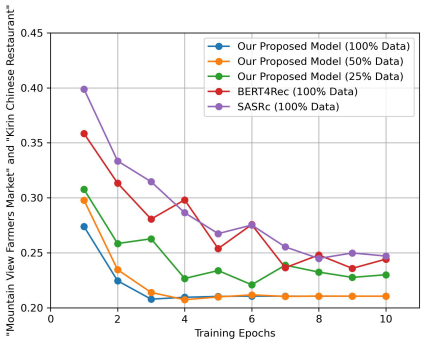
Table 4: Recommendation performance across three datasets using LLMs with varying levels of reasoning capability (lower is better for RMSE and MAE).

for similar items and to a low value for dissimilar items, and the convergence is even faster with more training data.

This result highlights the advantage of incorporating LLM reasoning into RecPIE. Traditional recommenders, built on collaborative filtering principles, require many co-occurrence observations (e.g., consumers visiting both In-N-Out and Five Guys) before capturing such similarities. In contrast, RecPIE leverages LLMs' reasoning ability to infer relationships with far fewer examples, accelerating convergence and improving representation quality.




(a) In-N-Out vs. Five Guys.




(b) Mountain View Farmers Market vs. Kirin Chinese Restaurant.

Figure 5: Embedding distances in the Google Maps dataset: RecPIE learns similarities (In-N-Out vs. Five Guys) and dissimilarities (Mountain View Farmers Market vs. Kirin Chinese Restaurant) more quickly.

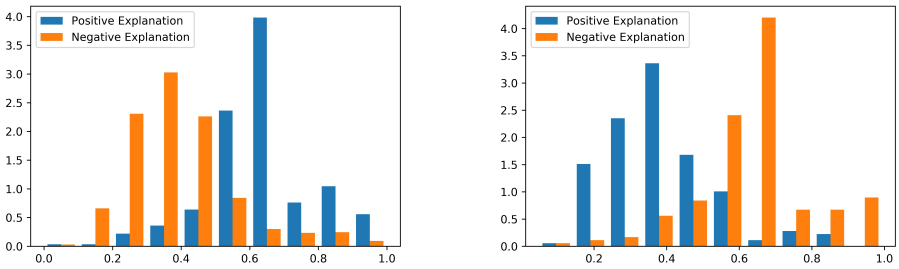
### D.5 The role of positive and negative explanations.

To test this, we conduct an *attention value analysis* to quantify the contributions of positive and negative explanations to the classification task. The attention value for the positive explanation $\bar{\alpha}_{pos}$ is computed as the average of all the relevant pairwise attention values in the attention layer of the

DNN component. We define:

$$\bar{\alpha}_{pos} = \frac{1}{|I|}\sum_{i \in I} \alpha_{i,pos}, \qquad \bar{\alpha}_{neg} = \frac{1}{|I|}\sum_{i \in I} \alpha_{i,neg}, \tag{11}$$

where $I = [\text{pos}, \text{neg}, c, p, \text{seq}, \text{context}]$ represents the index set corresponding to each element in the input $X_{\text{input}}$. In other words, $\bar{\alpha}_{pos}$ captures the average "attention" that the model puts on the positive explanation embedding when generating the final prediction, and $\bar{\alpha}_{neg}$ captures the average "attention" put on the negative explanation embedding. Therefore, $\bar{\alpha}_{pos}$ and $\bar{\alpha}_{neg}$ estimates the relative importance of the positive and negative explanations in producing the final recommendation results.




(a) Positive examples (Yelp Dataset). (b) Negative examples (Yelp Dataset).

Figure 6: (Color online) Distribution of attention values on positive and negative explanations for positive and negative examples.

We find that when a product receives a high rating, RecPIE assigns *more* attention to positive explanations than negative ones, as indicated by the distribution of attention weights for positive explanations (blue bars) being skewed further to the *right* compared to negative explanations (orange bars) (Fig.6(a)). Conversely, for products receiving a low rating, RecPIE assigns more attention to negative explanations, with the distribution of attention weights for positive explanations shifting further to the *left* compared to negative explanations (Fig. 6(b),). This confirms the value of the dual explanations: By providing both positive and negative explanations through the contrastive-explanation generator, RecPIE is able to intelligently decide which type of explanation to rely on, in order to generate the most accurate predictions.