Learning Invariant Link Formation Mechanisms via Pretraining and Adaptation

Author Name
Affiliation
email@example.com

Abstract

Link prediction (LP) is a central task in graph-based recommendation systems, enabling the discovery of potential user-item interactions. However, existing LP models often struggle with data sparsity, leading to spurious correlations and poor generalization. In this work, we explore pretraining as a scalable approach to causal learning for LP, aiming to extract invariant link formation mechanisms from large and diverse graphs. We propose a modular framework that decomposes link prediction into node-level and edge-level reasoning, and introduce a Mixture-of-Experts (MoE) architecture to model heterogeneous causal patterns across data subsets. For deployment, we adopt a parameter-efficient adaptation strategy that aggregates expert outputs without full model retraining. Our approach, PALP, achieves state-of-the-art performance and efficiency on six real-world datasets, demonstrating the promise of pretraining and modular adaptation as a scalable path toward causal representation learning in recommendation systems.

1 Introduction

Graph-based recommender systems have emerged as a powerful framework for modeling complex user-item interactions, enabling personalized content delivery in applications such as e-commerce, social networks, and media platforms [Perozzi *et al.*, 2014] [Fan *et al.*, 2019]. A central task in these systems is link prediction (LP), which estimates the likelihood of unseen interactions between entities based on the observed graph structure and node attributes.

Despite its importance, existing LP models often struggle with *data sparsity*, which manifests itself in two forms: limited training data within individual datasets and limited diversity across training distributions. As a result, models tend to learn spurious correlations by overfitting to dataset-specific artifacts or shortcuts, leading to poor generalization when deployed on new graphs or under distribution shifts.

Causal learning offers a promising framework to address these challenges by focusing on learning *invariant mechanisms*—factors that causally influence link formation and remain stable across environments [Xu *et al.*, 2025] [Gao *et al.*, 2024]. However, applying causal learning to LP and recommendation systems remains underexplored. Many existing approaches require interventional data, depend on strong assumptions about the data-generating process, or suffer from scalability limitations in large graphs [Zhu *et al.*, 2024] [Wang *et al.*, 2020].

In this work, we explore *pretraining* as a practical and scalable approach to causal learning for LP. By pretraining on large-scale and diverse graphs, we aim to expose models to a wide spectrum of generative patterns and enable them to distill generalizable knowledge that more closely approximates the true causal mechanisms behind link formation. Viewed through a causal lens, pretraining serves as an implicit form of environment diversity, supporting the discovery of stable and transferable factors that generalize across domains.

To capture a comprehensive set of causal factors, we propose a two-branch LP framework that decomposes link formation into *node-level* and *edge-level* processes. This modular design allows the model to isolate and learn from different generative signals, i.e., *semantic similarity* and *topological structure* [Mao *et al.*, 2023]. To further disentangle heterogeneous causal factors and enhance representational capacity, we introduce a Mixture-of-Experts (MoE) architecture [Ma *et al.*, 2024], in which each expert is responsible for modeling a distinct subset of the training data, thereby capturing different patterns of link generation. Accompanying this, a routing network dynamically selects which experts to consult for each input query, enabling fine-grained specialization.

During downstream deployment, we adopt a parameter-efficient adaptation strategy. Rather than fine-tuning the entire model, we learn a small set of aggregation weights over the pretrained experts, which is both computationally efficient and mitigates the risk of catastrophic forgetting.

Empirically, our framework, *Pretraining and Adaptation for Link Prediction* (PALP), achieves state-of-the-art generalization and efficiency across six real-world datasets. It consistently outperforms both classical and pretrained LP baselines while requiring over $10^4\times$ fewer FLOPs. These results support the view that pretraining combined with modular adaptation offers a scalable and effective path toward causal learning in recommendation systems. These findings reinforce the potential of combining large-scale pretraining with modular adaptation as a scalable and causally enhanced solution for robust recommendation systems.

Contributions. This work makes the following key contributions:

- We propose a novel perspective that leverages pretraining as an approach to discover invariant link formation mechanisms across diverse graphs.
- We develop a scalable framework that combines a two-branch design for semantic and structural modeling with a Mixture-of-Experts architecture to capture heterogeneous causal patterns.
- We propose a parameter-efficient adaptation strategy for robust deployment to new domains.
- We demonstrate the effectiveness of our approach (PALP) on six real-world datasets, achieving superior performance and efficiency compared to strong LP baselines.

2 Background

Link prediction (LP) aims to infer missing edges between node pairs in partially observed graphs. Given a graph G with adjacency matrix $A \in R^{n\times n}$ and node feature matrix $X \in R^{n\times d}$, the task is to estimate the probability of forming an edge (i, j). Prior work suggests that link formation is driven by two primary mechanisms: **feature proximity (FP)** and **structure proximity (SP)** [Mao *et al.*, 2023].

FP reflects homophily—the tendency of similar nodes to connect, and is captured by node encoders such as Message Passing Neural Networks (MPNNs) [Gilmer *et al.*, 2017], which produce embeddings used to compute link probabilities:

$$H = \text{NodeEncoder}(A, X), \quad p_{ij} = \text{ScoreFunction}(H_i \odot H_j).$$

where $\odot$ denotes the Hadamard product. However, MPNNs struggle to model higher-order structures like triangles, limiting their ability to approximate heuristics such as common neighbors, Adamic-Adar (AA), and Resource Allocation (RA) [Srinivasan and Ribeiro, 2019].

SP, on the other hand, captures topological relationships through pairwise structural encodings that consider path overlaps and neighborhood intersections [Katz, 1953] [Newman, 2001] [Li *et al.*, 2020]:

$$e_{ij} = \text{EdgeEncoder}(A, i, j), \quad p_{ij} = \text{ScoreFunction}(e_{ij}).$$

To exploit both FP and SP, recent approaches have proposed hybrid models that fuse node and edge representations [Wang *et al.*, 2024] [Yun *et al.*, 2021] [Wang *et al.*, 2023] [Shomer *et al.*, 2024]. A common strategy is to combine their outputs before scoring:

$$p_{ij} = \text{ScoreFunction}(H_i \odot H_j \mid e_{ij}).$$

3 Methodology

We propose **PALP**, a scalable link prediction framework that decouples feature-based and structure-based modeling during pretraining while enabling effective fusion during adaptation. PALP adopts a *two-branch pretraining* strategy with *Mixture-of-Experts (MoE)* in each branch to encode diverse link formation patterns. It then applies a lightweight fusion module to combine expert predictions at test time. Figure 1 illustrates the architecture.

3.1 Pretraining with MoE

Node module. We use NAGPhormer [Wang *et al.*, 2024] as the node encoder due to its linear complexity and ability to aggregate information from multiple hops, which is well-suited for personalized receptive fields in link prediction. The representation of node i is computed by:

$$h_i = \text{NAG}([x_i^0 \mid x_i^1 \mid \cdots \mid x_i^K]),$$

where x_i^k aggregates k-hop neighborhood information. Given node representations h_i and h_j, the link probability is estimated by:

$$p_{ij}^{\text{node}} = \sigma(\text{MLP}(h_i \odot h_j)),$$

where $\odot$ is the Hadamard product.

Edge module. To model structure proximity, we use BUDDY [Li *et al.*, 2020]—a heuristic structural encoder that captures critical link formation factors (e.g., common neighbors, Adamic-Adar, Resource Allocation) using node pair distances:

$$e_{ij} = \{B_{ij}[d], A_{ij}[d_u, d_v]\},$$

where $A_{ij}[d_u, d_v]$ counts nodes at distances d_u and d_v from i and j, respectively, and $B_{ij}[d] = \sum_{d_v > k} A_{ij}[d, d_v]$. The edge probability is computed as:

$$p_{ij}^{\text{edge}} = \sigma(\text{MLP}(e_{ij})).$$

MoE architecture. To increase the diversity of knowledge encoded during pretraining, we instantiate K expert MLPs per module and use a learnable gating function to softly assign edges to experts. The gating score is computed using:

$$z_{ij} = \text{MLP}(g_{ij}), \quad g_{ij} = x_i + x_j, \quad w_{ij}^k = -\|z_{ij} - c_k\|,$$

where c_k is the learnable centroid of the k-th cluster. We apply Gumbel-Softmax to ensure differentiability:

$$p_{ij}^k = \frac{\exp((w_{ij}^k + G_k)/\tau)}{\sum_{k'} \exp((w_{ij}^{k'} + G_{k'})/\tau)},$$

where $G_k \sim \text{Gumbel}(0, 1)$ and τ is annealed across epochs to promote early exploration and late exploitation. This mechanism enables finer-grained modeling of heterogeneity in link formation and helps prevent expert collapse.

Training. We optimize the sum of binary cross-entropy losses from both branches. Given positive edges E^+ and negative samples E^-, the loss is:

$$\mathcal{L} = -\frac{1}{|E^+| + |E^-|}\left(\sum_{(i,j)\in E^+} \log p_{ij} + \sum_{(i,j)\in E^-} \log(1 - p_{ij})\right).$$

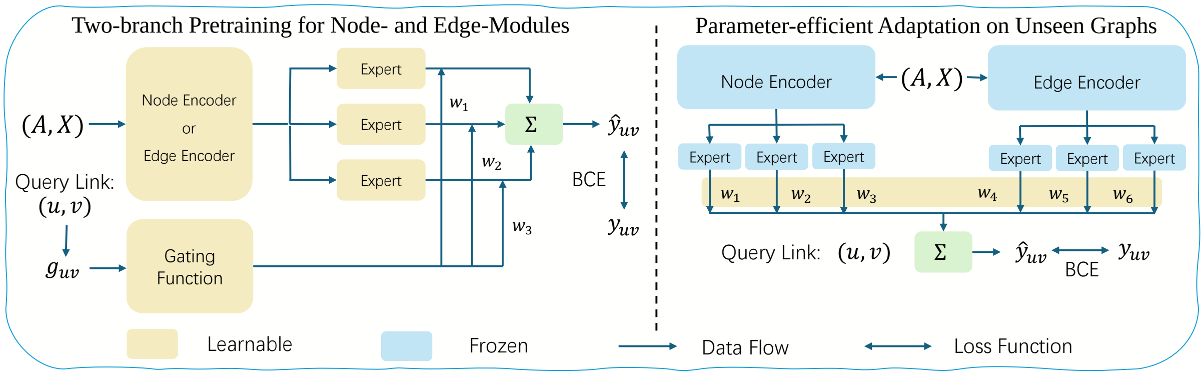

Figure 1: An overview of PALP. (Left) We individually train the node module and edge module on the pretraining dataset. Each module consists of a node/edge encoder to generate node/edge representations, a learnable gating function to route edges to different experts, and a set of score functions (experts) to output the predicted probabilities. (Right) During adaptation, we learn a weight vector to adaptively aggregate outputs from different experts. Note that all pretrained modules are kept frozen during adaptation, and only the weight vector is updated using downstream training data.

3.2 Adaptation and Fusion Strategy

After pretraining, PALP adapts to downstream graphs through a lightweight and efficient fusion mechanism. Specifically, we learn a global fusion vector $p \in R^K$ that assigns weights to the K pretrained experts across both the node and edge modules. Given the logits l_{ij}^k from each expert for edge (i, j), the final prediction is computed as:

$$p_{ij} = \sigma\left(\sum_k p^k \cdot l_{ij}^k\right),$$

where only the fusion weights p^k are updated during adaptation. All expert parameters are kept frozen. This strategy enables the downstream model to selectively aggregate knowledge from diverse pretraining signals while maintaining high efficiency. By leveraging soft aggregation over all experts, the model benefits from ensemble effects, improving robustness and generalization without incurring additional training overhead.

3.3 Complexity Analysis

PALP is designed for scalability, with its computational cost primarily arising from two components: representation generation and link scoring. For the edge module, structural features are computed using BUDDY during preprocessing, incurring no runtime cost during training. The node module uses NAGPhormer, which, after precomputing multi-hop propagated features, has a training complexity of $O(NKF^2)$, where N is the number of nodes, K is the number of hops, and F is the feature dimension. Since node embeddings are shared across edges, PALP supports efficient mini-batch training. The score function, used to compute probabilities for E edges, adds $O(EF^2)$ to the total cost. Altogether, the overall pretraining complexity of PALP is $O(NKF^2+EF^2)$, enabling efficient training on large-scale graphs with millions of nodes and edges.

4 Experiments

In this section, we evaluate the effectiveness and efficiency of PALP on six benchmark graphs. We compare PALP with a range of baselines and analyze how its design impacts performance and computational cost.

4.1 Experimental Setup

Pretraining Data. We pretrain PALP on the ogbn-papers100M dataset [Hu *et al.*, 2020], the largest publicly available academic graph. To address memory constraints stemming from the large-scale feature matrix and to improve training efficiency by eliminating on-the-fly sampling, we partition the graph into multiple subgraphs using the METIS algorithm, following the procedure in [Song *et al.*, 2024]. This preprocessing enables scalable training while preserving local structure. Detailed statistics of the resulting subgraph partitions are provided in Table 1.

Table 1: Summary of METIS partitions on ogbn-papers100M

#Graphs	Avg. #Nodes	Avg. #Edges	#Node Range	#Edge Range
11105	10000.90	61357.03	303 - 45748	328 - 122644

Evaluation Data. We evaluate PALP on six benchmark graphs spanning citation and e-commerce domains: Cora, Pubmed, Art, Business, History, and Child. The first four datasets are derived from academic citation networks and reflect literature recommendation scenarios. Since they share semantic and structural similarities with the pretraining corpus (ogbn-papers100M), they are considered in-domain. In contrast, History and Child are sampled from e-commerce interaction graphs [Chen *et al.*, 2024], representing product and book recommendation tasks. Due to their differing data distributions and domain semantics, they serve as cross-domain evaluation settings. For all datasets, we use a fixed edge split of 40% for training, 10% for validation, and 50% for testing. All node features are 384-dimensional SentenceBERT embeddings [Reimers and Gurevych, 2019], and structural features are precomputed using the sketching method from BUDDY [Wang *et al.*, 2024]. We evaluate model performance using Mean Reciprocal Rank (MRR), computed over 100 negative samples per positive link.

Baselines. We compare PALP against two classes of methods. The first includes general-purpose models such as MLP,

Table 2: Statistics of downstream datasets

Dataset Name	#Nodes	#Edges	Domain
Cora	2,708	10,858	Citation
Pubmed	19,717	88,670	Citation
Art	58,373	7,184	Citation
Business	4,279	36,697	Citation
History	4,153	12,622	E-commerce
Child	3,819	45,408	E-commerce

GCN [Kipf and Welling, 2016], and SAGE [Hamilton *et al.*, 2017]. The second includes specialized link prediction models: NCN [Wang *et al.*, 2023], Neo-GNN [Yun *et al.*, 2021] and BUDDY [Wang *et al.*, 2024]. All baselines are trained end-to-end using the same node features.

4.2 Performance Comparison

Effectiveness. Table 3 presents the effectiveness comparison across all six datasets. PALP consistently outperforms baseline methods on most datasets. Notably, it achieves significant improvements on Cora and History, surpassing the second-best model by over 3% in MRR. These gains are likely due to strong semantic alignment between the downstream tasks and the pretraining corpus. On graphs with lower similarity to the pretraining data, e.g., Child, PALP slightly underperforms the top baseline but remains highly competitive. These results highlight PALP's robust generalization ability, even under domain shifts, and demonstrate its effectiveness in both in-domain and cross-domain scenarios.

Efficiency. Beyond predictive accuracy, PALP offers substantial efficiency benefits. As illustrated in Figure 2, PALP reduces training-time FLOPs by over 10,000× compared to end-to-end models. This is achieved by freezing all pretrained modules and updating only a lightweight fusion vector. The adaptation process consists of a single pass through the expert modules (node and edge) to extract logits, followed by training a logistic regressor with only K parameters. This design makes PALP a practical solution for link prediction on resource-constrained platforms or rapid deployment scenarios.

Table 3: Performance comparison on benchmark datasets. Metric: Mean Reciprocal Rank (MRR).

	Cora	Pubmed	Art	Business	History	Child
MLP	54.97	66.86	56.71	40.76	64.53	70.48
GCN	53.53	70.56	62.98	41.25	66.41	**75.79**
SAGE	54.40	**73.02**	56.17	41.59	65.11	75.16
Neo	50.52	65.68	55.87	26.40	63.89	68.59
NCN	57.47	70.46	63.43	40.90	71.26	75.23
BUDDY	58.28	69.45	63.93	39.61	67.02	72.54
PALP	**63.94**	71.33	**65.77**	**42.35**	**74.88**	72.65

4.3 Ablation Study

We conduct an ablation study to assess the contribution of different components in PALP. Specifically, we compare the following four variants:

- **Node-only**: Uses only the pretrained node module for link prediction.

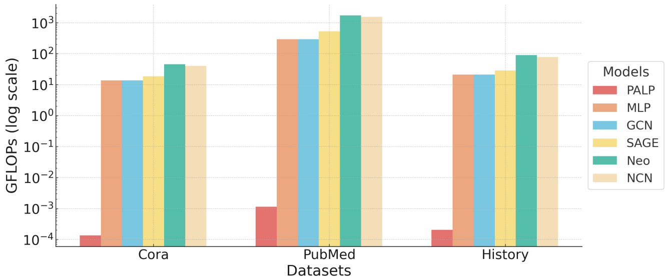

Figure 2: Comparison of per-epoch FLOPs for different methods. Numbers shown in log-scale.

- **Edge-only**: Uses only the pretrained edge module for link prediction.
- **PALP w/o MoE**: A simplified version of PALP where each module contains only a single expert, removing the Mixture-of-Experts architecture.
- **PALP**: Our full method that adaptively fuses expert outputs using parameter-efficient tuning.

Table 4 reports the results on the Cora and Child datasets. First, Node-only achieves strong results on both datasets, confirming the importance of feature proximity. However, Edge-only lags behind, suggesting that structural signals alone can be insufficient to model the factors of link formation. Second, PALP w/o MoE achieves performance close to Node-only, but slightly lower on both datasets. This indicates that naively fusing semantic and structural information may not provide benefits and even introduce conflicts across branches. Finally, the full PALP model achieves the best performance, showing the clear advantage of combining both modules with MoE. This confirms that MoE plays a critical role in capturing diverse link formation patterns and enabling effective adaptation across heterogeneous datasets. Overall, these results validate that each component contributes meaningfully to PALP's overall effectiveness.

Table 4: Ablation study results on Cora and Child datasets. We compare Node-only, Edge-only, PALP, and PALP w/o MoE.

Method	Cora	Child
Node-only	70.70	71.41
Edge-only	50.76	68.53
PALP w/o MoE	70.45	70.42
PALP	72.35	76.07

5 Conclusion

In this work, we introduced PALP, a scalable and causally motivated framework for link prediction in graph-based recommendation systems. By leveraging large-scale pretraining and a modular two-branch architecture, PALP effectively captures diverse generative factors underlying link formation, including both semantic and structural signals. The integration of a Mixture-of-Experts architecture enables the model to specialize in heterogeneous data regimes, while our parameter-efficient adaptation strategy ensures robust and efficient deployment across diverse downstream domains. Empirical results on six real-world datasets demonstrate that PALP achieves state-of-the-art performance while drastically reducing computational overhead. These findings highlight the potential of pretraining, when properly modularized and adapted, to serve as a practical pathway toward scalable and causally grounded link prediction.

References

[Chen *et al.*, 2024] Zhikai Chen, Haitao Mao, Jingzhe Liu, Yu Song, Bingheng Li, Wei Jin, Bahare Fatemi, Anton Tsitsulin, Bryan Perozzi, Hui Liu, et al. Text-space graph foundation models: Comprehensive benchmarks and new insights. *arXiv preprint arXiv:2406.10727*, 2024.

[Fan *et al.*, 2019] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. Graph neural networks for social recommendation. In *The world wide web conference*, pages 417–426, 2019.

[Gao *et al.*, 2024] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. Causal inference in recommender systems: A survey and future directions. *ACM Transactions on Information Systems*, 42(4):1–32, 2024.

[Gilmer *et al.*, 2017] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[Hu *et al.*, 2020] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[Katz, 1953] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Li *et al.*, 2020] Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. *Advances in Neural Information Processing Systems*, 33:4465–4478, 2020.

[Ma *et al.*, 2024] Li Ma, Haoyu Han, Juanhui Li, Harry Shomer, Hui Liu, Xiaofeng Gao, and Jiliang Tang. Mixture of link predictors. *arXiv preprint arXiv:2402.08583*, 2024.

[Mao *et al.*, 2023] Haitao Mao, Juanhui Li, Harry Shomer, Bingheng Li, Wenqi Fan, Yao Ma, Tong Zhao, Neil Shah, and Jiliang Tang. Revisiting link prediction: A data perspective. *arXiv preprint arXiv:2310.00793*, 2023.

[Newman, 2001] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, and Steven S. Skiena. Deepwalk: online learning of social representations. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[Shomer *et al.*, 2024] Harry Shomer, Yao Ma, Haitao Mao, Juanhui Li, Bo Wu, and Jiliang Tang. Lpformer: An adaptive graph transformer for link prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2686–2698, 2024.

[Song *et al.*, 2024] Yu Song, Haitao Mao, Jiachen Xiao, Jingzhe Liu, Zhikai Chen, Wei Jin, Carl Yang, Jiliang Tang, and Hui Liu. A pure transformer pretraining framework on text-attributed graphs. *arXiv preprint arXiv:2406.13873*, 2024.

[Srinivasan and Ribeiro, 2019] Balasubramaniam Srinivasan and Bruno Ribeiro. On the equivalence between positional node embeddings and structural graph representations. *arXiv preprint arXiv:1910.00452*, 2019.

[Wang *et al.*, 2020] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. Causal inference for recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 426–431, 2020.

[Wang *et al.*, 2023] Xiyuan Wang, Haotong Yang, and Muhan Zhang. Neural common neighbor with completion for link prediction. *arXiv preprint arXiv:2302.00890*, 2023.

[Wang *et al.*, 2024] Zehong Wang, Zheyuan Zhang, Chuxu Zhang, and Yanfang Ye. Subgraph pooling: Tackling negative transfer on graphs. In *International Joint Conferences on Artificial Intelligence*, 2024.

[Xu *et al.*, 2025] Shuyuan Xu, Jianchao Ji, Yunqi Li, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. Causal inference for recommendation: Foundations, methods, and applications. *ACM Transactions on Intelligent Systems and Technology*, 16(3):1–51, 2025.

[Yun *et al.*, 2021] Seongjun Yun, Seoyoon Kim, Junhyun Lee, Jaewoo Kang, and Hyunwoo J Kim. Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. *Advances in Neural Information Processing Systems*, 34:13683–13694, 2021.

[Zhu *et al.*, 2024] Yaochen Zhu, Jing Yi, Jiayi Xie, and Zhenzhong Chen. Deep causal reasoning for recommendations. *ACM Transactions on Intelligent Systems and Technology*, 15(4):1–25, 2024.