Task-Aware Retrieval Augmentation for Dynamic Recommendation

Zhen Tao[1,2*], Xinke Jiang[3*], Qingshuai Feng[1], Haoyu Zhang[4], Lun Du[5], Yuchen Fang[6], Hao Miao[7], Bangquan Xie[1], Qingqiang Sun[1,8†]

[1]Great Bay University
[2]Nanjing University
[3]Independent Researcher
[4]City University of Hong Kong
[5]Ant Research
[6]University of Electronic Science and Technology of China
[7]Hong Kong Polytechnic University
[8]Dongguan Key Laboratory of Intelligence Equipment and Smart Industry
zhentao.tz@gmail.com, qqsun@gbu.edu.cn

Abstract

Dynamic recommendation systems aim to provide personalized suggestions by modeling temporal user-item interactions across time-series behavioral data. Recent studies have leveraged pre-trained dynamic graph neural networks (GNNs) to learn user-item representations over temporal snapshot graphs. However, fine-tuning GNNs on these graphs often results in generalization issues due to temporal discrepancies between pre-training and fine-tuning stages, limiting the model's ability to capture evolving user preferences. To address this, we propose TarDGR, a task-aware retrieval-augmented framework designed to enhance generalization capability by incorporating task-aware model and retrieval-augmentation. Specifically, TarDGR introduces a Task-Aware Evaluation Mechanism to identify semantically relevant historical subgraphs, enabling the construction of task-specific datasets without manual labeling. It also presents a Graph Transformer-based Task-Aware Model that integrates semantic and structural encodings to assess subgraph relevance. During inference, TarDGR retrieves and fuses task-aware subgraphs with the query subgraph, enriching its representation and mitigating temporal generalization issues. Experiments on multiple large-scale dynamic graph datasets demonstrate that TarDGR consistently outperforms state-of-the-art methods, with extensive empirical evidence underscoring its superior accuracy and generalization capabilities.

1 Introduction

Dynamic recommendation systems aim to generate personalized suggestions by modeling how user-item interactions evolve over time through rich temporal behavior data (Zhu et al. 2021b; Yang et al. 2023b; Yu et al. 2024b; Chen et al. 2025). To capture such temporal dynamics, recent approaches have adopted Graph Neural Networks (GNNs) (Yu et al. 2024d,c; Jiang et al. 2025, 2023) that learn user-item representations over sequences of time-evolving snapshot graphs (Yang et al. 2024; Yu et al. 2024b). These GNN-based methods typically follow a pretraining–finetuning paradigm, where models are first trained on historical graphs to learn transferable structural patterns, and subsequently fine-tuned on recent temporal graphs to adapt to evolving user behavior. However, despite their success, these models often suffer from *generalization issues* when fine-tuned on new temporal graphs, due to temporal discrepancies between pre-training and fine-tuning interaction graphs (Cong et al. 2024; Lu et al. 2024; Tao et al. 2025; Yu et al. 2023, 2025c; Liang, Gel, and Chen 2025). As the temporal context shifts and user interests evolve, previously learned patterns may no longer align with the current data distribution, limiting the model's ability to deliver accurate recommendations for future interactions.

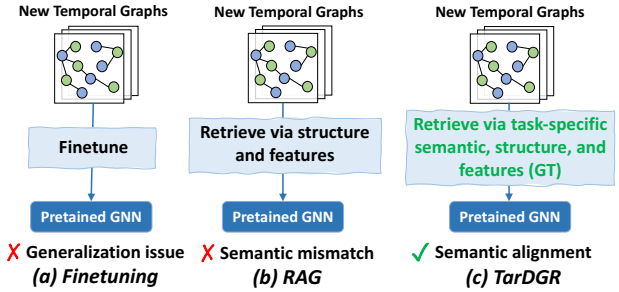


Figure 1: Comparison of current methods with TarDGR in dynamic graph recommendation.

Recent advances in Retrieval-augmented Generation (RAG) techniques have shown promise in addressing generalization issues by incorporating dynamic retrieval mechanisms, significantly enhancing model capabilities without requiring parameter updates (Wu et al. 2024; Parashar et al. 2024; Jiang* et al. 2025). For dynamic recommendation tasks, providing appropriate context for predicting time-sensitive recommendation representations is crucial for improving generalization capability. While existing works have introduced external graph data through structural and feature similarity-based retrieval, they often overlook the seman-

*These authors contributed equally.
†Corresponding author.

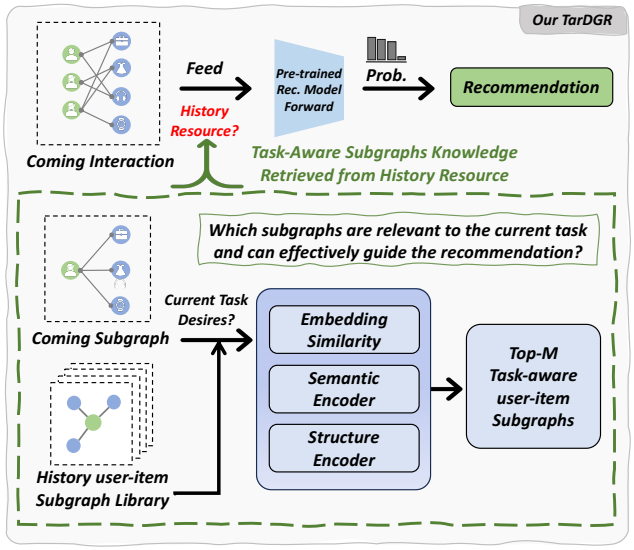


Figure 2: Task-Aware Retrieval Recommendation.

tic task relevance between retrieval and query graphs (Jiang et al. 2024). This limitation is particularly problematic for recommendation tasks, where structurally similar subgraphs may not be beneficial when semantic inconsistency exists among nodes. A comparison of these paradigms is illustrated in Figure 1, highlighting their respective temporal adaptation strategies and limitations. To fully realize the potential of retrieval-augmented recommendation in dynamic environments, we must overcome the following key challenges: **C1. How to effectively identify task-relevant subgraphs for recommendation?** Current retrieval-based approaches for graph models rely primarily on structural and feature similarity when retrieving historical subgraphs. They overlook semantic task relevance between retrieval and query graphs, assuming all structurally similar subgraphs provide equal value. For recommendation tasks, this assumption proves problematic as structurally similar subgraphs may contribute little or even negatively when semantic inconsistency exists. Existing frameworks lack mechanisms to evaluate whether retrieved subgraphs actually benefit the specific recommendation task at hand.

C2. How to enable models to understand task-specific requirements without manual annotation? In the realm of graph recommendation, models struggle to interpret what kind of subgraph information best serves the current query recommendation graph. Unlike language model research where task-awareness can be incorporated through dataset construction (Qiao et al. 2025; Sun et al. 2025; Liu et al. 2024), graph data's inherent complexity makes manual dataset creation nearly infeasible. Without properly understanding task requirements, models cannot effectively transfer knowledge from historical data to new temporal contexts, resulting in suboptimal recommendations in dynamic environments.

To address these challenges, we propose **TarDGR** (**T**ask-**A**ware **R**etrieval Augmentation for **D**ynamic **G**raph **R**ecommendation), a novel framework that enhances generalization capability by incorporating task-aware model and retrieval-augmentation. As illustrated in Figure 2, TarDGR identifies and injects task-specific knowledge into the query which enriches the representation. First, to tackle **C1**, TarDGR introduces a *Task-Aware Evaluation Mechanism* that automatically identifies semantically relevant historical resource subgraphs by evaluating how their integration affects similarity with positive recommendation samples. This enables the construction of task-specific datasets without manual annotations, defining clear criteria for "task-beneficial" and "task-harmful" subgraphs. Second, addressing **C2**, TarDGR presents a *Graph Transformer-based Task-Aware Model* that combines semantic and structural encodings to evaluate the relevance of subgraphs in relation to current recommendation task requirements. During inference, the model retrieves and fuses task-relevant subgraphs with the query subgraph, thereby enhancing its representation ability and mitigating generalization issues induced by temporal shifts. We summarize our contributions as follows:

- We propose a task-aware retrieval-augmented framework for dynamic graph recommendation, offering the first exploration of task-awareness in graph learning for recommendation.
- We introduce a novel automatic task-aware evaluation framework that distinguishes task-beneficial and task-harmful subgraphs, enabling the construction of task-aware datasets without manual annotations.
- We design a Graph Transformer-based Task-Aware Model that effectively captures subgraph relevance by integrating semantic and structural encodings, enabling more accurate relevance estimation and robust knowledge transfer under temporal shifts.
- We apply the task-aware retrieval-augmentation technique to dynamic graph recommendation systems and demonstrate that TarDGR consistently outperforms state-of-the-art methods across three real-world datasets.

2 Related Works

Dynamic Recommendation. Dynamic recommendation has been addressed through sequential models such as BERT4Rec (Sun et al. 2019) and DCRec (Yang et al. 2023a), which rely on fixed historical sequences without explicitly modeling temporal graph structures. To better capture structural dynamics, dynamic graph neural networks (DGNNs) have emerged, including EvolveGCN (Pareja et al. 2020), ROLAND (You, Du, and Leskovec 2022), and WinGNN (Zhu et al. 2023), which explicitly model structural evolution over time. In parallel, the pretraining–finetuning paradigm has proven effective for transferring structural knowledge in graph learning (Yu et al. 2025a,b, 2024a). GraphPro (Yang et al. 2024) extends this to dynamic recommendation by incorporating temporal prompts during pretraining and fine-tuning, achieving promising performance. However, significant temporal shifts between pretraining and fine-tuning snapshots can lead to generalization issues, ultimately limiting the model's ability to adapt to evolving user preferences.

Retrieval-Augmented Generation. RAG enhances pretrained language models by retrieving external knowledge

to construct informative context for downstream tasks (Zhu et al. 2021a; Lewis et al. 2021). These systems typically retrieve documents or entities from large corpora to guide generation, improving factuality and interpretability (Sarthi et al. 2024; Gao et al. 2022). RAG has been successfully extended to various modalities, including vision, code, audio, and video (Zheng et al. 2025; Yang et al. 2025; Xue et al. 2024; Singh et al. 2025). In the graph domain, RAG has been applied to knowledge graphs by leveraging node-level textual features (Xu et al. 2024). Recent works like RAGRAPH (Jiang et al. 2024) propose plug-and-play retrieval to augment pre-trained GNNs. However, these methods do not consider task-specific relevance during retrieval. As a result, semantically misaligned subgraphs may be introduced, degrading the quality of the downstream recommendation. Our approach addresses this gap by incorporating task-aware retrieval mechanisms.

3 Preliminaries

Problem Formulation We model dynamic recommendation scenarios as a sequence of temporal user-item interaction graphs. Formally, the dynamic graph is denoted by $\mathcal{G} = \{G_t\}_{t=1}^{T}$, where each snapshot at time step t is represented as $G_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{X}_t, \mathcal{A}_t)$. Here, $\mathcal{V}_t$ denotes the node set, $\mathcal{E}_t$ the edge set, $\mathcal{X}_t$ the feature matrix, and $\mathcal{A}_t$ the adjacency matrix at time t. To facilitate temporal modeling and evaluation, the complete temporal graph $\mathcal{G}$ is partitioned along the time axis into a training set $\mathcal{G}_{\text{train}}$ and a test set $\mathcal{G}_{\text{test}}$ (Jiang et al. 2024). Given a temporal graph $\mathcal{G}$, we formulate dynamic graph recommendation as the task of learning a predictive model to forecast future user-item interactions. To improve generalization under temporal distribution shifts, we adopt an augmented learning framework in which the model enhances the query subgraph by retrieving and incorporating relevant subgraphs from historical interactions.

Recommendation Subgraph Library. To improve structural generalization in dynamic recommendation, we construct a *Recommendation Subgraph Library* from historical user-item interactions, utilizing FAISS-based embedding retrieval (Douze et al. 2024). Each resource subgraph is extracted as a k-hop neighborhood centered around nodes involved in past interactions, forming a collection $\mathcal{G}_R = \{G(v_r)\}_{r=1}^{R}$. Specifically, each subgraph is defined as:

$$G(v_r) = \left(v_r, \mathcal{N}^{(k)}(v_r), \mathcal{X}_r, \mathcal{A}_r\right), \tag{1}$$

where v_r denotes the central node, and $\mathcal{N}^{(k)}(v_r)$ represents its k-hop neighborhood, including both user and item nodes. In our formulation, the embedding of each subgraph serves as the *key*, while its graph structure and representation jointly constitute the *value*.

Task Relevance Estimation. To retrieve semantically relevant subgraphs for a given recommendation query, we define a task relevance function that evaluates the utility of a candidate subgraph. Given a task $\mathcal{T}$, a query subgraph $G(v_q)$, and a candidate subgraph $G(v_r)$ from the resource pool, the task relevance score is defined as:

$$\text{REL}(G(v_q), G(v_r) \mid \mathcal{T}) = \mathcal{R}_\theta\Big(G(v_q), G(v_r)\Big), \tag{2}$$

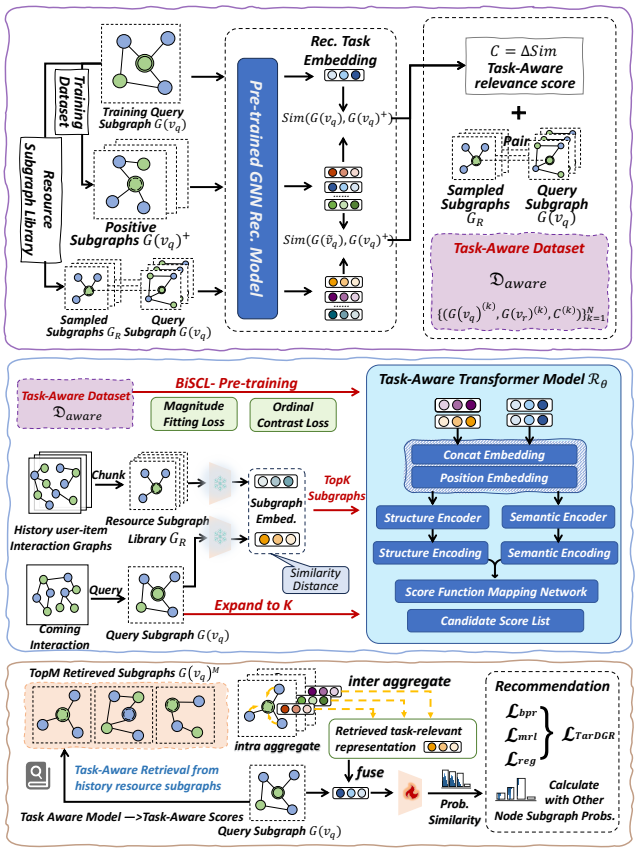


Figure 3: Overview of the TarDGR framework.

where $\mathcal{R}_\theta$ is a task-aware neural model parameterized by θ, designed to jointly encode both the query graph and the candidate subgraph as dual inputs.

4 TarDGR Framework

In this section, we present **TarDGR**, a novel task-aware retrieval framework designed to enhance the generalization ability of dynamic graph recommendations. As illustrated in Figure 3, the top depicts *the evaluation mechanism part*, which automatically constructs task-specific supervision signals. The middle presents *the model part*, detailing the input formulation, architectural design, and BiSCL pretraining. The bottom presents *the inference and training pipeline*, where task-aware subgraphs are retrieved and integrated to enhance dynamic recommendation. A formal theoretical analysis of the generalization benefits of TarDGR is provided in Appendix.

4.1 Task-Aware Evaluation Mechanism

We propose an automated task-aware evaluation mechanism that quantifies the contribution of a candidate historical subgraph $G(v_r)$ to the current recommendation query $G(v_q)$.

Using a pre-trained GNN recommendation model f_{pre}^{θ}, both subgraphs are encoded into embeddings $\text{ENC}\big(G(v_q)\big)$ and $\text{ENC}\big(G(v_r)\big)$. We first compute the average cosine similarity between the query embedding and a set of positive

historical subgraphs $\{G(v_q)^+\}$:

$$\overline{\text{SIM}}_{\text{before}} = \frac{1}{N^+}\sum_{i=1}^{N^+} \text{COS}\Big(\text{ENC}\big(G(v_q)\big), \text{ENC}\big(G(v_q)_i^+\big)\Big). \quad (3)$$

Next, we fuse the query subgraph $G(v_q)$ with the candidate subgraph $G(v_r)$ by constructing inter-subgraph links between the central modes and applying graph convolution to obtain a combined representation:

$$\text{ENC}\big(G(\tilde{v_q})\big) = f_{\text{fuse}}\Big(\text{ENC}\big(G(v_q) \oplus G(v_r)\big)\Big). \quad (4)$$

The updated similarity to the positive set is then:

$$\overline{\text{SIM}}_{\text{after}} = \frac{1}{N^+}\sum_{i=1}^{N^+} \text{COS}\big(\text{ENC}\big(G(\tilde{v_q})\big), \text{ENC}\big((G(v_q)_i^+\big)\big). \quad (5)$$

We define the relative similarity shift as: $\Delta\text{REL} = \overline{\text{SIM}}_{\text{after}} - \overline{\text{SIM}}_{\text{before}}$. This ΔREL is utilized as the task relevance score C_r, quantifying the degree to which the candidate subgraph $G(v_r)$ contributes to current recommendation. Specifically:

- If $\Delta\text{REL} > 0$, the candidate subgraph is positively correlated with the task and is considered a beneficial sample.
- If $\Delta\text{REL} \approx 0$, it is deemed irrelevant.
- If $\Delta\text{REL} < 0$, it is negatively correlated and considered harmful to task performance.

Based on this scoring, we construct a task-aware dataset $\mathcal{D}_{\text{aware}} = \{(G(v_q), G(v_r), C_r)\}$, where each triplet consists of a query subgraph, a candidate recommendation subgraph, and their associated task relevance score. This dataset provides task-aligned supervision signals for the subsequent training of a task-aware graph recommendation model.

4.2 Graph Transformer-based Task-Aware Model

To equip the dynamic graph recommendation system with task-aware retrieval augmentation, we propose the **Graph Transformer-based Task-Aware Model** denoted as $\mathcal{R}_\theta$. This model is designed to effectively capture the complex interactions between a query subgraph and candidate subgraphs by leveraging the expressive power of graph transformers, thereby enhancing the relevance estimation and retrieval performance in dynamic recommendation scenarios.

Subgraph Semantic Encoder To capture the temporal evolution of user preferences and interaction patterns, we initialize node embeddings using a pre-trained dynamic GNN (Yang et al. 2024), which encodes historical temporal dependencies across graph snapshots. Specifically, the initial node representations at the fine-tuning step t are obtained via forward propagation over the graph at time $t-1$: $h_t = \text{forward}(h_{t-1}; G_{t-1})$.

Each resource subgraph $G(v_r)$ is encoded by applying L-layer graph convolutions on its temporally contextualized node embeddings to yield subgraph-level representations:

$$h_r = \sum_{l=0}^{L} \text{GCONV}(h_t^r, \mathcal{G}(v_r)) \in \mathbb{R}^d. \quad (6)$$

Given the query subgraph embedding $h_q \in \mathbb{R}^d$, we compute pairwise L2 distances to resource subgraph embeddings $\{h_r\}_{r=1}^R$: $\text{dist}(h_q, h_r) = h_q^\top h_q + h_r^\top h_r - 2h_q^\top h_r$. The top-$K$ resource subgraphs with the smallest distances to the query subgraph are retrieved to form the initial candidate set:

$$\mathcal{G}(v_q)^K = \text{TOPK}_{\text{search}}\left(G(v_q), G(v_r)\right). \quad (7)$$

Each candidate is paired with the query subgraph to form the matching set: $\big\{\langle G(v_q), G(v_i)\rangle \mid G(v_i) \in \mathcal{G}(v_q)^K\big\}$. To enable pairwise semantic modeling, we jointly encode both the query and candidate subgraphs:

$$h = \left[\sum_{l=0}^{L} \text{GCONV}(h_t^q, \mathcal{G}(v_q)) \,\middle\|\, \sum_{l=0}^{L} \text{GCONV}(h_t^i, \mathcal{G}(v_i))\right]. \quad (8)$$

A positional embedding P is added to h to encode the relative positional order, $h_{\text{pos}} = h + P$. We input h_{pos} into a multi-head self-attention module to capture fine-grained relational dependencies between query and candidate subgraphs. It is projected into query, key, and value matrices: $Q = h_{\text{pos}}W_Q, K = h_{\text{pos}}W_K, V = h_{\text{pos}}W_V$. Attention weights are computed as: $\text{Attn} = \text{SOFTMAX}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$, where d_k represents the dimensionality of the key vectors. The final subgraph-level semantic representation is:

$$h_{\text{sem}} = \text{CONCAT}(\text{Attn}_1, \ldots, \text{Attn}_H)W_O, \quad (9)$$

where W_O is a trainable output projection matrix. The resulting embedding h_{sem} serves as the task-aware semantic encoding of the query-candidate subgraph pair.

Subgraph Structure Encoder We further employ a dedicated subgraph structure encoder to encode structural dependencies within each subgraph. The positional-enhanced embedding $h_{\text{pos}} \in \mathbb{R}^{2d}$ is first linearly projected into a lower-dimensional latent space: $h_{\text{hid}} = h_{\text{pos}}W + b$.

We then apply multi-layer, multi-head attention to capture fine-grained dependencies. At the l-th layer, query, key, and value matrices are computed via linear transformations, followed by attention calculation and concatenation across heads. Each attention layer is followed by residual connection and layer normalization:

$$h_{\text{hid}}^{(l+1)} = \text{LAYERNORM}(h_{\text{hid}}^{(l)} + \text{Attn}_{\text{output}}^{(l)}) \quad (10)$$

We apply position-wise feedforward network (FFN) to enhance representation and introduce non-linearity. The final FFN output is denoted as h_{ffn}. To encode subgraph-level structural patterns, the FFN output is aggregated via normalized adjacency propagation:

$$h_{\text{str}} = \mathcal{D}^{-1}(\mathcal{A}_{\text{s}} + \mathbf{I})h_{\text{ffn}}W, \quad (11)$$

where $\mathcal{D}$ is the degree matrix, $\mathcal{A}_{\text{s}}$ is corresponding adjacency matrix and $W \in \mathbb{R}^{d_{\text{hid}} \times d}$.

The semantic encoding h_{sem} and structural encoding h_{str} are concatenated to form a task-aware fused representation: $h_{\text{task}} = \text{CONCAT}(h_{\text{sem}}, h_{\text{str}})$. The representation is transformed into a task-specific relevance score via a parametric scoring function $\mathcal{S}_\psi(\cdot)$:

$$s_i = \mathcal{S}_\psi(h_{\text{task}}) = w^\top \text{RELU}(Wh_{\text{task}} + b), \quad (12)$$

where w maps the hidden representation to a scalar score.

BiSCL Pretraining of Task-Aware Model We introduce Bi-Level Supervised Correlation Loss (**BiSCL**), which pretrains the task-aware model by jointly supervising numerical fidelity and ordinal consistency.

Given $\mathcal{D}_{\text{aware}} = (G(v_q), G(v_r), C)$, each subgraph is encoded into semantic-structural embeddings z_q and z_r as in Equation 6. The concatenated pairwise feature $h_{q,r} = [z_q, |, z_r]$ and adjacency matrix $\mathcal{A}_{\text{s}}$ are then fed into $\mathcal{R}_\theta$ to compute the predicted task relevance: $\mathcal{R}_\theta(h_{q,r}, \mathcal{A}_{\text{s}})$.

We apply a magnitude fitting loss to minimize the discrepancy between predicted and task relevance scores:

$$\mathcal{L}_{\text{mtl}} = \frac{1}{N} \sum_{k=1}^{N} \left(\mathcal{R}_\theta(h_{q,r}, \mathcal{A}_{\text{s}}) - C\right)^2. \tag{13}$$

In parallel, we impose a pairwise ordinal constraint loss to preserve inter-sample ordering. Specifically, for every pair (k, l) such that $C^{(k)} > C^{(l)}$, the predicted scores must maintain this ranking:

$$\mathcal{L}_{\text{ocl}} = \log \left[1 + \sum_{k,l} \exp\left(\frac{\mathcal{R}_\theta(h_{q,r}^{(l)}, \mathcal{A}_{\text{s}}) - \mathcal{R}_\theta(h_{q,r}^{(k)}, \mathcal{A}_{\text{s}})}{\tau} \right) \right], \tag{14}$$

where τ is a temperature hyperparameter controlling the penalty's smoothness. The BiSCL loss is expressed as:

$$\mathcal{L}_{\text{BiSCL}} = \rho \cdot \mathcal{L}_{\text{ocl}} + (1 - \rho) \cdot \mathcal{L}_{\text{mtl}}, \tag{15}$$

where $\rho \in [0, 1]$ balances absolute fidelity and ordinal coherence. BiSCL injects task-aware inductive signals, facilitating robust pretraining for downstream retrieval task.

4.3 Task-Aware Retrieval Inference and Training

Given a query node v_q and subgraph $G(v_q)$, we obtain task-aware relevance score list $\{s^{(i)}\}_{i=1}^{K}$ by $\mathcal{R}_\theta$ based on task-aware model in Section 4.2. We select the top-M subgraphs with the highest task-specific relevance scores to construct the set for retrieval augmentation $\left\{G(v_m) \in \mathcal{G}(v_q)^M\right\}$. Through intra-graph aggregation, we obtain the internal representation of all task-relevant subgraphs:

$$h_m = \sum_{l=1}^{L} \textsc{GConv}^{(l)}(h_t, G(v_m)), \tag{16}$$

where h_t denotes features from dynamic encoder. Retrieved subgraphs are then aggregated via soft evidence aggregation:

$$H_{\text{rag}} = \sum_{i=1}^{M} \alpha_i \cdot h_m^i, \quad \sum_{i=1}^{M} \alpha_i = 1, \tag{17}$$

where α_i are normalized weights indicating retrieval confidence, either uniform or learned from $\mathcal{R}_\theta$ scores. Then, we employ a residual fusion mechanism to integrate retrieval representation into the query subgraph:

$$\tilde{h}_q = \beta h_q + (1 - \beta) H_{\text{rag}}, \tag{18}$$

where β is a learnable gate balancing original and retrieved task-relevant representation.

Semantic retrieval of historical subgraphs recovers latent dependencies beyond the query context, improving generalization across temporal recommendation tasks. Finally, we conduct a joint fine-tuning recommendation loss function. Structural robustness is encouraged by injecting stochastic perturbations into the graph topology: $\mathcal{E}' = \{(u, v) \in \mathcal{E} \mid \text{Bernoulli}(r) = 1\}, \quad \mathcal{G}' = (\mathcal{V}, \mathcal{E}')$.

User preference is modeled using a task-aware scoring function trained under a margin-based objective. For each training triplet (u, i^+, i^-), where u and i denote the user and item respectively, the corresponding subgraph inputs are constructed as: $X^+ = [h_u \,\|\, h_{i^+}]$ and $X^- = [h_u \,\|\, h_{i^-}]$, followed by relevance scoring via $\mathcal{R}_\theta$. The resulting margin ranking loss encourages the model to distinguish relevant items from irrelevant ones in a task-consistent manner:

$$\mathcal{L}_{\text{mrl}} = \frac{1}{|\mathcal{B}|} \sum_{(u,i^+,i^-)\in\mathcal{B}} \max(0, \gamma - (s^+ - s^-)). \tag{19}$$

To mitigate overfitting and promote stable training, we apply a penalty to the embedding norms:

$$\mathcal{L}_{\text{reg}} = \frac{1}{2N} \left(\sum_{u\in\mathcal{B}_u} \|h_u\|_2^2 + \sum_{i\in\mathcal{B}_i^+} \|h_i^+\|_2^2 + \sum_{i\in\mathcal{B}_i^-} \|h_i^-\|_2^2 \right). \tag{20}$$

The recommendation preference is captured via the bayesian personalized ranking loss (Rendle et al. 2012):

$$\mathcal{L}_{\text{bpr}} = - \sum_{(u,i^+,i^-)\in\mathcal{B}} \log \sigma(h_u^\top h_{i^+} - h_u^\top h_{i^-}), \tag{21}$$

where $\sigma(\cdot)$ denotes the Sigmoid function. The final loss aggregates all components:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{bpr}} + \lambda \cdot \mathcal{L}_{\text{mrl}} + \mu \cdot \mathcal{L}_{\text{reg}}, \tag{22}$$

where λ and μ are weighting hyperparameters. This optimization encourages the model to align recommendation signals with task-aware semantics, while improving robustness under perturbation and preventing overfitting.

5 Experiments

This section presents experiments designed to evaluate the performance of **TarDGR**, against state-of-the-art baselines on three dynamic graph datasets. Further experiments and analyses are provided in the Appendix.

5.1 Experimental Setup

Datasets. We evaluate our method on three public datasets spanning diverse dynamic recommendation scenarios: *Taobao*, with 10 days of implicit feedback from the Taobao platform; *Koubei*, a 9-week user–store interaction dataset from Alipay's location service released for IJCAI'16; and *Amazon*, containing 13 weeks of product review data. Additional details are provided in Appendix.

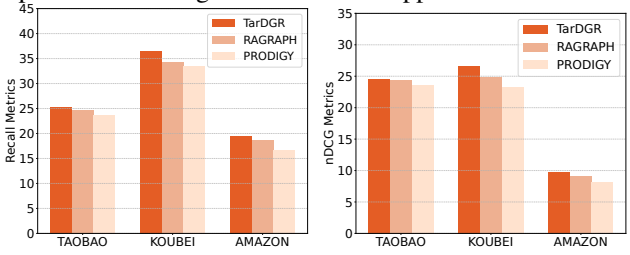
Methods and Baselines. We compare our approach against representative baselines spanning four major categories: GNN-based recommenders, including LightGCN (He et al. 2020) and its self-supervised variants such as SGL (Wu et al.

Method	TAOBAO		KOUBEI		AMAZON	
	Recall	nDCG	Recall	nDCG	Recall	nDCG
LightGCN	22.47±02.53	21.89±02.80	30.21±06.45	22.24±05.83	15.07±06.48	06.53±02.66
SGL	22.15±02.20	22.12±03.09	32.61±04.27	22.36±04.82	15.78±07.12	07.90±02.49
MixGCF	22.84±02.15	23.05±03.87	32.06±04.20	22.49±06.91	15.24±08.98	07.40±03.44
SimGCL	22.18±02.22	23.15±02.75	33.07±05.28	23.08±05.55	16.10±07.91	07.58±03.51
GraphPrompt	20.76±01.54	20.22±00.98	33.24±04.98	24.12±09.20	16.20±08.58	07.89±04.12
GPF	22.46±01.66	22.12±01.16	33.70±06.44	24.39±04.01	17.67±09.04	08.94±04.57
EvolveGCN-H	22.44±02.55	22.17±01.79	31.22±04.25	23.00±02.92	14.97±10.28	07.20±05.43
EvolveGCN-O	23.64±02.13	23.24±01.28	33.01±05.22	23.98±04.01	17.48±08.13	08.68±04.25
ROLAND	22.67±02.42	22.60±01.91	30.11±03.14	22.29±01.84	15.33±07.10	07.09±03.02
GraphPro+						
Vanilla/NF	20.10±01.50	20.12±01.30	21.31±04.59	15.31±03.11	12.56±07.45	06.31±03.92
Vanilla/FT	23.99±02.11	23.26±01.42	33.96±04.13	24.66±02.78	18.14±07.55	08.73±03.74
PRODIGY/NF	21.67±01.42	23.15±03.20	21.66±03.21	14.82±03.92	11.88±02.61	05.84±01.84
PRODIGY/FT	23.74±01.22	23.65±02.31	33.46±04.70	23.28±03.40	16.72±04.28	08.09±02.66
RAGRAPH/NF	20.31±01.60	20.45±01.44	22.86±03.44	16.68±02.48	13.78±05.54	06.52±02.69
RAGRAPH/FT	24.78±01.93	24.35±01.34	34.27±03.93	24.82±02.69	18.69±07.45	09.09±03.89
TarDGR/NF	20.39±02.41	20.91±02.18	24.83±03.68	17.90±02.62	14.26±05.37	06.74±02.56
TarDGR/FT	**25.20**±02.13	**24.59**±01.42	**36.52**±04.44	**26.63**±02.98	**19.56**±07.17	**09.70**±03.62

Table 1: Main performance comparison results of TarDGR. The best performance is bolded, and the second is underlined.

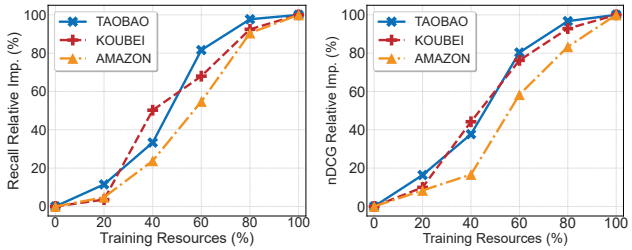
2021), MixGCF (Huang et al. 2021), and SimGCL (Yu et al. 2022); Dynamic GNNs, including EvolveGCN-H/O (Pareja et al. 2020), ROLAND (You, Du, and Leskovec 2022), and GraphPro (Yang et al. 2024); Graph prompting models, including GraphPrompt (Liu et al. 2023) and GPF (Fang et al. 2023); and Retrieval-augmented models, such as PRODIGY (Huang et al. 2023) and RAGRAPH (Jiang et al. 2024), where retrieved subgraphs are integrated into GraphPro to enhance contextual representation. Further details on the baselines are provided in Appendix.

Settings and Evaluation. We adopt the pre-trained dynamic graph dataset as the resource pool. For retrieval-based methods, we consider two variants: non-fine-tuned (NF), which applies plug-and-play retrieval augmentation without additional training on the target dataset, and fine-tuned (FT), which applies tuning on the training set. All models are pre-trained on historical snapshots and subsequently fine-tuned and evaluated on future snapshots. We report average performance over time using Recall@20 and nDCG@20 (He et al. 2020; Yu et al. 2022). Further evaluation metrics and experimental settings are detailed in Appendix.



(a) Recall (b) nDCG

Figure 4: Performance comparison of TarDGR and other RAG methods.



(a) Recall (b) nDCG

Figure 5: Training resource experiments for the Graph Transformer-based Aware Model applied to TarDGR.

5.2 Baseline Performance Comparison

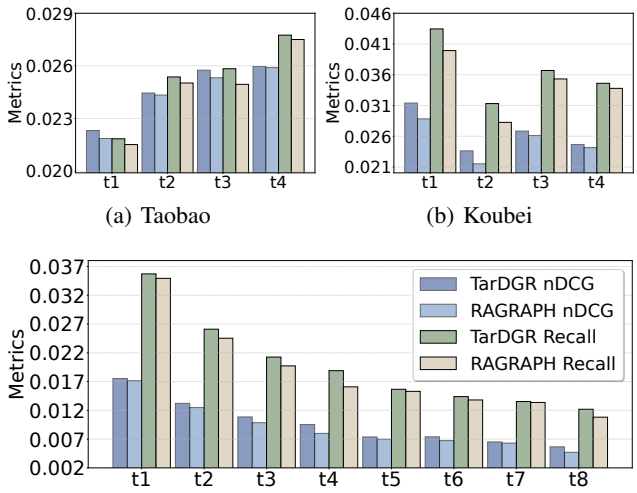
Table 1 summarizes the main experimental results, where TarDGR consistently outperforms all baselines across the three benchmark datasets. TarDGR/FT achieves the best overall performance, verifying the efficacy of our task-aware retrieval mechanism. By leveraging semantically aligned subgraphs from prior temporal resources, the model effectively transfers history knowledge to downstream recommendation tasks, reinforcing the value of task-aware enhancement in dynamic recommendation settings. By incorporating task-aware retrieval into the GraphPro framework, TarDGR achieves significant gains over RAGRAPH and PRODIGY. On Amazon, as shown in Table 4, it outperforms PRODIGY by 16.6% in nDCG and 14.5% in Recall, and RAGRAPH by 6.3% and 4.4%, respectively. These improvements highlight the importance of injecting task-specific subgraphs to enhance temporal generalization.

To further validate generalization over time, we provide a detailed comparison across individual time steps in Fig-

Method	TAOBAO		KOUBEI		AMAZON	
	Recall	nDCG	Recall	nDCG	Recall	nDCG
w/o all	24.63±01.81	24.02±01.79	34.14±03.57	24.83±02.13	18.42±06.71	08.91±03.04
w/o SEM	24.95±01.75	24.48±01.57	35.84±03.88	25.56±02.53	19.10±07.52	09.45±03.59
w/o STR	25.14±02.10	24.52±01.63	36.30±03.61	26.12±02.75	19.42±06.88	09.57±03.95
TarDGR	25.20±02.13	24.59±01.42	36.52±04.44	26.63±02.98	19.56±07.17	09.70±03.62

Table 2: Ablation Study on Graph Transformer-based Aware Model.

ure 6 between TarDGR and RAGRAPH. TarDGR consistently demonstrates stronger performance across all time snapshots, particularly during earlier stages where user interactions are more closely aligned with resource subgraphs. This indicates that TarDGR benefits from enhanced transferability of task-relevant signals from historical contexts.



(a) Taobao (b) Koubei

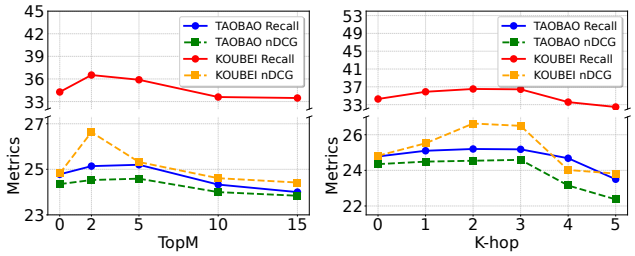
(c) Amazon

Figure 6: Comparison of TarDGR/FT and RAGRAPH/FT performance on each time step.

5.3 Ablation Study of TarDGR

Impact of Semantic and Structural Encoding. Table 2 presents an ablation study assessing the contributions of the semantic and structural encoders within the TarDGR framework. The removal of both components (w/o all) causes a significant performance drop, affirming the necessity of task-aware representation learning. Removing the semantic encoder (w/o SEM) results in a larger performance drop than removing the structural encoder (w/o STR), indicating that capturing semantic relevance between subgraph embeddings via attention is particularly vital for generating accurate relevance scores. Nonetheless, omitting structural encoding (w/o STR) also causes noticeable degradation, demonstrating its complementary value in modeling subgraph-level structural compatibility.

Effect of Task-Aware Resource Weighting. We examine the impact of task-aware training resource weight in TarDGR, as shown in Figure 5. The performance follows a non-linear trend: initial improvements are slow due to limited supervision; as more subgraphs are introduced, the model quickly gains expressiveness and improves; finally, performance saturates when the resource pool becomes redundant or overly dense. The trend underscores the importance of task-aligned supervision and highlights that the quantity of retrieved training subgraphs are essential for maximizing the effectiveness of the TarDGR framework. This trend underscores the importance of task-aligned supervision and validates our retrieval-based training strategy for enhancing generalization in dynamic recommendation.



(a) Hyper-parameter Top-M (b) Hyper-parameter k-hop

Figure 7: Hyper-parameter study results of M and k.

5.4 Hyperparameter Sensitivity

We investigate the effect of two key hyperparameters: the number of retrieved subgraphs Top-M and neighborhood depth k-hop, as shown in Figure 7. Increasing M introduces more retrieved knowledge, which initially enhances performance by providing richer contextual signals. However, excessive retrieval introduces noise and hinders generalization. Similarly, larger k values enable the model to aggregate broader structural context. Yet, excessive neighborhood expansion leads to oversized subgraphs with redundant information, which may overwhelm the graph encoder, reduce representation quality and impair generalization.

6 Conclusion

We present TarDGR, a task-aware retrieval-augmented framework for dynamic graph recommendation. By integrating a task-aware evaluation mechanism and a graph transformer-based task-aware model, TarDGR adaptively selects and fuses semantically relevant historical subgraphs to enhance representation learning under temporal dynamics. This design effectively mitigates generalization degradation caused by temporal shifts between pretraining and fine-tuning stages. Experimental results validate the effectiveness of TarDGR in dynamic graph recommendation tasks. In the future, our framework can be extended to other specific tasks by adapting the task-aware objective.

Acknowledgments

This work was supported by the Natural Science Foundation of Guangdong Province, China (No. 2024A1515110162).

References

Chen, W.; Yuan, M.; Zhang, Z.; Xie, R.; Zhuang, F.; Wang, D.; and Liu, R. 2025. FairDgcl: Fairness-aware recommendation with dynamic graph contrastive learning. *IEEE Transactions on Knowledge and Data Engineering*.

Cong, W.; Kang, J.; Tong, H.; and Mahdavi, M. 2024. On the generalization capability of temporal graph learning algorithms: Theoretical insights and a simpler method. *arXiv preprint arXiv:2402.16387*.

Douze, M.; Guzhva, A.; Deng, C.; Johnson, J.; Szilvasy, G.; Mazaré, P.-E.; Lomeli, M.; Hosseini, L.; and Jégou, H. 2024. The Faiss library. arXiv:2401.08281.

Fang, T.; Zhang, Y.; Yang, Y.; Wang, C.; and Chen, L. 2023. Universal prompt tuning for graph neural networks. *Advances in Neural Information Processing Systems*, 36: 52464–52489.

Gao, L.; Ma, X.; Lin, J.; and Callan, J. 2022. Precise Zero-Shot Dense Retrieval without Relevance Labels. arXiv:2212.10496.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*.

Huang, Q.; Ren, H.; Chen, P.; Kržmanc, G.; Zeng, D.; Liang, P.; and Leskovec, J. 2023. PRODIGY: Enabling In-context Learning Over Graphs. arXiv:2305.12600.

Huang, T.; Dong, Y.; Ding, M.; Yang, Z.; Feng, W.; Wang, X.; and Tang, J. 2021. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 665–674.

Jiang, X.; Qin, Z.; Xu, J.; and Ao, X. 2023. Incomplete graph learning via attribute-structure decoupled variational auto-encoder. In *WSDM oral 2023*, 304–312.

Jiang, X.; Qiu, R.; Xu, Y.; Zhu, Y.; Zhang, R.; Fang, Y.; Xu, C.; Zhao, J.; and Wang, Y. 2024. Ragraph: A general retrieval-augmented graph learning framework. *Advances in Neural Information Processing Systems*, 37: 29948–29985.

Jiang*, X.; Zhang*, R.; Xu*, Y.; Qiu*, R.; Fang, Y.; Wang, Z.; Tang, J.; Ding, H.; Chu, X.; Zhao, J.; et al. 2025. HyKGE: A Hypothesis Knowledge Graph Enhanced Framework for Accurate and Reliable Medical LLMs Responses. *ACL 2025*.

Jiang, X.; Zhang, W.; Fang, Y.; Gao, X.; Chen, H.; Zhang, H.; Zhuang, D.; and Luo, J. 2025. Time Series Supplier Allocation via Deep Black-Litterman Model. *AAAI 2025*, 39(11): 11870–11878.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; tau Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401.

Liang, P.; Gel, Y. R.; and Chen, Y. 2025. Topology-Informed Pre-training of Graph Neural Networks. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Liu, Y.; Cao, J.; Liu, C.; Ding, K.; and Jin, L. 2024. Datasets for large language models: A comprehensive survey. *arXiv preprint arXiv:2402.18041*.

Liu, Z.; Yu, X.; Fang, Y.; and Zhang, X. 2023. GraphPrompt: Unifying Pre-Training and Downstream Tasks for Graph Neural Networks. In *WWW*.

Lu, B.; Ma, T.; Gan, X.; Wang, X.; Zhu, Y.; Zhou, C.; and Liang, S. 2024. Temporal generalization estimation in evolving graphs. *arXiv preprint arXiv:2404.04969*.

Parashar, S.; Lin, Z.; Liu, T.; Dong, X.; Li, Y.; Ramanan, D.; Caverlee, J.; and Kong, S. 2024. The Neglected Tails of Vision-Language Models. In *CVPR*.

Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T.; and Leiserson, C. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 5363–5370.

Qiao, S.; Qiu, Z.; Ren, B.; Wang, X.; Ru, X.; Zhang, N.; Chen, X.; Jiang, Y.; Xie, P.; Huang, F.; et al. 2025. Agentic knowledgeable self-awareness. *arXiv preprint arXiv:2504.03553*.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Sarthi, P.; Abdullah, S.; Tuli, A.; Khanna, S.; Goldie, A.; and Manning, C. D. 2024. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. In *ICLR*.

Singh, A.; Ehtesham, A.; Kumar, S.; and Khoei, T. T. 2025. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1441–1450.

Sun, R.; Wang, Z.; Sun, J.; and Ranjan, R. 2025. Vision: How to fully unleash the productivity of Agentic AI? Decentralized Agent Swarm Network. In *ICML 2025 Workshop on Collaborative and Federated Agentic Workflows*.

Tao, Z.; Cao, Y.; Fang, Y.; Liu, Y.; Zhao, X.; and He, T. 2025. Dynamic Graph Recommendation via Sparse Augmentation and Singular Adaptation. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 726–735.

Wu, S.; Xiong, Y.; Cui, Y.; Wu, H.; Chen, C.; Yuan, Y.; Huang, L.; Liu, X.; Kuo, T.-W.; Guan, N.; et al. 2024. Retrieval-augmented generation for natural language processing: A survey. *arXiv preprint arXiv:2407.13193*.

Xu, Y.; He, S.; Chen, J.; Wang, Z.; Song, Y.; Tong, H.; Liu, G.; Liu, K.; and Zhao, J. 2024. Generate-on-graph: Treat llm as both agent and kg in incomplete knowledge graph question answering. *arXiv preprint arXiv:2404.14741*.

Xue, J.; Deng, Y.; Gao, Y.; and Li, Y. 2024. Retrieval augmented generation in prompt-based text-to-speech synthesis with context-aware contrastive language-audio pretraining. *arXiv preprint arXiv:2406.03714*.

Yang, Y.; Huang, C.; Xia, L.; Huang, C.; Luo, D.; and Lin, K. 2023a. Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2023*, 1063–1073.

Yang, Y.; Xia, L.; Luo, D.; Lin, K.; and Huang, C. 2024. Graphpro: Graph pre-training and prompt learning for recommendation. In *Proceedings of the ACM Web Conference 2024*, 3690–3699.

Yang, Z.; Chen, S.; Gao, C.; Li, Z.; Hu, X.; Liu, K.; and Xia, X. 2025. An empirical study of retrieval-augmented code generation: Challenges and opportunities. *ACM Transactions on Software Engineering and Methodology*.

Yang, Z.; He, X.; Zhang, J.; Wu, J.; Xin, X.; Chen, J.; and Wang, X. 2023b. A generic learning framework for sequential recommendation with distribution shifts. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 331–340.

You, J.; Du, T.; and Leskovec, J. 2022. ROLAND: graph learning framework for dynamic graphs. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2358–2366.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; and Nguyen, Q. V. H. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 1294–1303.

Yu, Q.; Zou, L.; Luo, X.; Zhao, X.; and Li, C. 2025a. Uniform Graph Pre-training and Prompting for Transferable Recommendation. *ACM Transactions on Information Systems*.

Yu, X.; Fang, Y.; Liu, Z.; Wu, Y.; Wen, Z.; Bo, J.; Zhang, X.; and Hoi, S. C. 2024a. A Survey of Few-Shot Learning on Graphs: from Meta-Learning to Pre-Training and Prompt Learning. *arXiv preprint arXiv:2402.01440*.

Yu, X.; Liu, Z.; Fang, Y.; Liu, Z.; Chen, S.; and Zhang, X. 2023. Generalized Graph Prompt: Toward a Unification of Pre-Training and Downstream Tasks on Graphs. *IEEE TKDE*.

Yu, X.; Liu, Z.; Fang, Y.; and Zhang, X. 2024b. DyGPrompt: Learning Feature and Time Prompts on Dynamic Graphs. In *ICLR*.

Yu, X.; Liu, Z.; Fang, Y.; and Zhang, X. 2024c. HGPROMPT: Bridging Homogeneous and Heterogeneous Graphs for Few-shot Prompt Learning. In *AAAI*.

Yu, X.; Zhang, J.; Fang, Y.; and Jiang, R. 2024d. Non-Homophilic Graph Pre-Training and Prompt Learning. *arXiv preprint arXiv:2408.12594*.

Yu, X.; Zhang, J.; Fang, Y.; and Jiang, R. 2025b. Non-homophilic graph pre-training and prompt learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 1844–1854.

Yu, X.; Zhou, C.; Kuai, Z.; Zhang, X.; and Fang, Y. 2025c. GCoT: Chain-of-Thought Prompt Learning for Graphs. In *SIGKDD*.

Zheng, X.; Weng, Z.; Lyu, Y.; Jiang, L.; Xue, H.; Ren, B.; Paudel, D.; Sebe, N.; Van Gool, L.; and Hu, X. 2025. Retrieval augmented generation and understanding in vision: A survey and new outlook. *arXiv preprint arXiv:2503.18016*.

Zhu, F.; Lei, W.; Wang, C.; Zheng, J.; Poria, S.; and Chua, T.-S. 2021a. Retrieving and Reading: A Comprehensive Survey on Open-domain Question Answering. arXiv:2101.00774.

Zhu, Y.; Cong, F.; Zhang, D.; Gong, W.; Lin, Q.; Feng, W.; Dong, Y.; and Tang, J. 2023. Wingnn: Dynamic graph neural networks with random gradient aggregation window. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 3650–3662.

Zhu, Z.; He, Y.; Zhao, X.; and Caverlee, J. 2021b. Popularity bias in dynamic recommendation. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2439–2449.