GLCSRec: Integrating Graph Collaborative Signals with Large Language Models for Item Cold-Start Recommendation

Ying LIN†, Chongxian CHEN†, Xin FAN†, and Hayato YAMANA††

† Graduate School of Fundamental Science and Engineering,
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
†† Faculty of Science and Engineering, Waseda University,
3-4-1 Okubo, Shinjuku-ku, Tokyo, 169-8555, Japan
E-mail: †alisa@ruri.waseda.jp, ††chenc@toki.waseda.jp, †††fan_xin@fuji.waseda.jp,
††††yamana@yama.info.waseda.ac.jp

Abstract Recommendation systems aim to precisely match users with items. However, newly introduced items often encounter the item cold-start problem due to insufficient historical interaction data. This data sparsity undermines the performance of collaborative filtering methods. Although large language models (LLMs) offer potential for processing new item content through their semantic reasoning capabilities, they inherently lack integration of collaborative signals, which constitutes a critical structural limitation. To address this, we propose GLCSRec (Graph-LLM Cold-Start Recommender). Unlike previous approaches that directly perform vector space alignment, this model utilizes LightGCN to extract user features from their historical interactions and projects them into soft prompts through a multilayer perceptron (MLP). The soft prompts consist of continuous and trainable vectors and are embedded in the input layer of the LLM, thereby guiding the causal attention mechanism of the LLM to conduct personalized semantic reasoning and preference prediction for the metadata of cold-start items. The model is trained with the Bayesian personalized ranking (BPR) loss, with optimization applied exclusively to the user embeddings and the projector parameters. A comprehensive experimental evaluation demonstrated the effectiveness of GLCSRec: (1) Compared to the representative baseline models, the Recall@10 increased from 0.0534 to 0.0765. (2) The ablation experiments confirmed the necessity of the collaborative signal, as replacing the GNN embeddings with semantic features led to a decrease in Recall@10 from 0.0805 to 0.0545. (3) The sensitivity and robustness analysis verified the model's stability, maintaining MRR@10 stable at 0.3484 even as data sparsity increased.

Key words Recommender Systems, Item Cold-Start, Graph Neural Networks (GNN), Large Language Models (LLM), Soft Prompt

1 Introduction

Recommender systems play a vital role in modern web services, acting as the primary engine for information filtering [1]. Collaborative filtering models [2], such as LightGCN [3], are academic benchmarks models due to their exceptional ability to capture high-order user-item interaction structures. However, these models fundamentally rely on the density of the interaction graph, leading to the item cold-start problem [4, 5]. Specifically, in the item cold-start scenario where new items in the test set have never appeared in the training set, traditional models fail to recommend due to they lack historical data. Our experiments show that in this scenario, LightGCN's Recall@10 drops to 0.0, highlighting the need to develop models that understand the content of cold-start items to recommend.

Large language models (LLMs) [6] possess semantic reasoning capabilities and have the potential to provide semantic reasoning for cold-start items. However, our experiments reveal that using LLMs alone results in poor performance (Recall@10 0.0067) due to there is a lack of collaborative signals from the user history; although LLMs can understand the cold-start items, they are unable to identify which users will interact with them [6].

How to effectively integrate the highly structured user-item relationships captured by graph neural networks (GNNs) [7] into the semantic reasoning capabilities of LLMs is a challenge. To bridge this gap, we propose the GLCSRec framework, which integrates the collaborative signals of GNNs and the semantic reasoning capabilities of LLMs. Instead of using a direct alignment method, we project the user fea-

tures extracted by LightGCN from the user interaction history through a multilayer perceptron (MLP) into the soft prompts [8]. The resulting collaborative signals are transformed into continuous vectors, used as soft prompts [9]. These prompts can effectively guide the LLM to enable the causal attention mechanism to perform personalized semantic reasoning and preference prediction for the metadata of cold-start item, thereby compensating for the lack of interaction data.

The contributions of this thesis are as follows:

1. Our proposed method effectively integrates the structured signals of GNNs into the semantic reasoning capabilities of LLMs, rather than only relying on the content's metadata. Compared with the representative baselines, it has increased the Recall@10 from 0.0534 to 0.0765.
2. The ablation experiment results show that after removing the GNN signal and only relying on the textual description, the Recall@10 drops by 32.3%. Therefore, we have confirmed that GNN can effectively capture the collaborative signals that LLM cannot derive, proving the irreplaceability of structural signals.
3. We verified the GLCSRec that even when the training interaction data is reduced by 46.7% (the cold-start ratio increases from 0.1 to 0.5), the MRR@10 still remains stable at 0.3484. This indicates that our architecture effectively integrates the GNN signal through the soft prompt to the LLM, and can maintain precise semantic reasoning and ranking even in sparse scenarios.

2 Related Work

This chapter reviews related work, addressing the cold-start problem, by classifying them into 1) content-based and hybrid models, 2) graph neural networks-based, and 3) large language models-based. By analyzing the limitations of related work, we identify the research gaps that motivate our proposed framework.

2.1 Content-based and Hybrid Models

Content-based filtering [10] addresses item cold-start by substituting interaction data with metadata. SBERT [11] encodes textual descriptions to calculate similarity with user preference [12]. While Ding et al. [13] utilized semantic retrieval, they focus on explicit relevance, overlooking high-order collaborative signals.

Hybrid models like DeepFM [14] rely on trained ID embeddings, failing for cold-start items lacking vectors. Other models [1,15] struggle to balance metadata and collaborative signals [5], potentially separating reasoning from ranking [6].

Alignment approaches learn from ID-based space to semantics [15]: MeLU [16] uses few-shot learning but requires initial signals; CLCRec [17] utilizes contrastive objectives but may lose information during vector mapping.

2.2 Graph Reconstruction for Cold-Start Recommendation

Graph neural networks (GNNs) model user-item interactions as bipartite graphs [7, 18], utilizing topological structure to capture high-order connectivity. Representative models like NGCF [18] and LightGCN [3] focus on collaborative smoothing via neighborhood aggregation; however, standard GNNs remain inherently transductive [19, 20], relying on edges to propagate signals and failing to generate representations for isolated cold-start nodes. To overcome this, CGRC [21] explores graph reconstruction by constructing edges from content features to alleviate isolation. This approach remains limited as surface-level similarity introduces structural noise [22], weakening user representation robustness. Furthermore, treating text as construction features rather than semantic guidance [21] lacks deep reasoning, failing to understand subtle preferences and text descriptions.

2.3 LLMs for Cold-start Recommendation

LLMs enable generative semantic reasoning [6, 23], interpreting complex intentions and contexts where traditional models struggle. In Zero-Shot settings, LLMs (e.g., TinyLlama [24]) rank preferences but encounter collaborative gaps [6, 25] from unrecognized historical behavior and popularity bias, failing to capture personalized requirements [25]. TALLRec [26] and RecLM [27] utilize lightweight fine-tuning or instruction tuning to integrate behavioral representations, yet lengthy natural language prompts increase computational demands and linear text limits capturing high-order collaborative relationships. Soft prompt methods [9, 28] address costs by freezing backbones and optimizing continuous virtual tokens. While GraphPrompter [8] encoded structural information for general graph learning, it remains unadapted to integrate collaborative signals into cold-start item inference lacking interaction history.

Table 1: Summary of representative related work

Category	Models	Overview	Research Gaps
Content & Hybrid	DeepFM [14], CLCRec [17]	Metadata encoding; contrastive alignment.	Information loss; difficulty in balancing signals.
GNN-based	LightGCN [3], CGRC [21]	Capture high-order connectivity; reconstructing graph.	Isolation of cold-start nodes; structural noise.
LLM-based	TALLRec [26], RecLM [27]	Generative reasoning; instruction tuning.	Lack of collaborative signals; high costs.

3 Preliminaries

This chapter explains the theoretical foundations and

mathematical notations required for our proposed GLCSRec framework.

3.1 Bayesian Personalized Ranking (BPR)

We adopt a pairwise optimization strategy,i.e., Bayesian personalized ranking (BPR) [29]. This method learns personalized rankings based on pairs of items, which assumes that the user's preference for observed items is higher than unobserved items. This optimization is robust for ranking tasks and helps maintain recommendation stability even in data-sparse scenarios. The BPR loss (L_{BPR}) is defined:

$$L_{BPR} = -\sum_{(u,i^+,i^-)\in D} \ln \sigma(\hat{y}_{ui^+} - \hat{y}_{ui^-}) + \lambda_\theta \|\Theta\|_2^2 \quad (1)$$

3.2 Graph Neural Networks (GNNs)

LightGCN simplifies message passing using normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, updating embeddings: $\mathbf{E}^{(k+1)} = \tilde{\mathbf{A}}\mathbf{E}^{(k)}$. However, standard GNNs are limited by transductive nature [20]; for isolated cold-start nodes $i \in I_{cold}$ with no interaction history, the propagation rule becomes ineffective:

$$\mathbf{e}_i^{(k+1)} = \tilde{\mathbf{A}}_{i,:}\mathbf{E}^{(k)} = \mathbf{0} \quad (2)$$

3.3 Parameter-efficient Fine-tuning (PEFT)

To address computational costs, soft prompt methods [9] freeze backbone architectures and optimize continuous virtual tokens $\mathbf{P} = \{p_1, \ldots, p_l\}$. Following GraphPrompter [8], GNNs encode local structures into vectors projected via MLP:

$$\hat{\mathbf{X}}_i = \text{MLP}(\text{GNN}(\mathbf{G}_{s_i})) \in \mathbb{R}^{d_l} \quad (3)$$

The final LLM input concatenates structural soft prompts and text embeddings $\mathbf{T}_{\text{emb}}$:

$$\mathbf{H}_{\text{input}} = [\hat{\mathbf{X}}_i; \mathbf{T}_{\text{emb}}] \quad (4)$$

This allows utilizing structural signals and semantic knowledge without complete parameter fine-tuning.

4 The Proposed Method: GLCSRec

The design of GLCSRec (Graph-LLM Cold-Start Recommender) is motivated by the need to bridge the information gap between structured collaborative signals and semantic text knowledge in the item cold-start recommendation task. We propose a collaborative prompt mechanism and use a parameter-efficient strategy to fine-tune the complete model. The overall architecture is shown in Figure 1.

4.1 Preparation of Data

Cold-Start Item Partitioning. Let $U = \{u_1, \ldots, u_m\}$ and $I = \{i_1, \ldots, i_n\}$ denote users and items, where interactions are represented by binary matrix $\mathbf{Y} \in \{0,1\}^{M\times N}$. The item set is partitioned into disjoint subsets: $I = I_{train} \cup I_{cold}$, where $I_{train} \cap I_{cold} = \emptyset$. I_{train} contains history for structural learning, while I_{cold} denotes new items lacking history. The objective is to learn mapping function $f : (U, I_{cold}) \to \mathbb{R}$ predicting preference score $\hat{y}_{uj}$ for cold-start item $j \in I_{cold}$, relying on metadata $\mathbf{T}_j$ and user collaborative context.

Preparation of GNN. We implement a masking protocol, removing all interactions for $j \in I_{cold}$ from the bipartite graph to eliminate transductive leakage. Since item j has no neighboring nodes, the messaging mechanism fails to propagate cooperative signals, resulting in decoupled representation:

$$\mathbf{e}_j = \alpha_0 \mathbf{e}_j^{(0)} + \sum_{k=1}^{K} \alpha_k \mathbf{0} = \alpha_0 \mathbf{e}_j^{(0)} \quad (5)$$

Item ID embedding $\mathbf{e}_j$ remains as unoptimized random noise from random initialization. Lacking interaction data, these embeddings cannot be updated during training, lead-

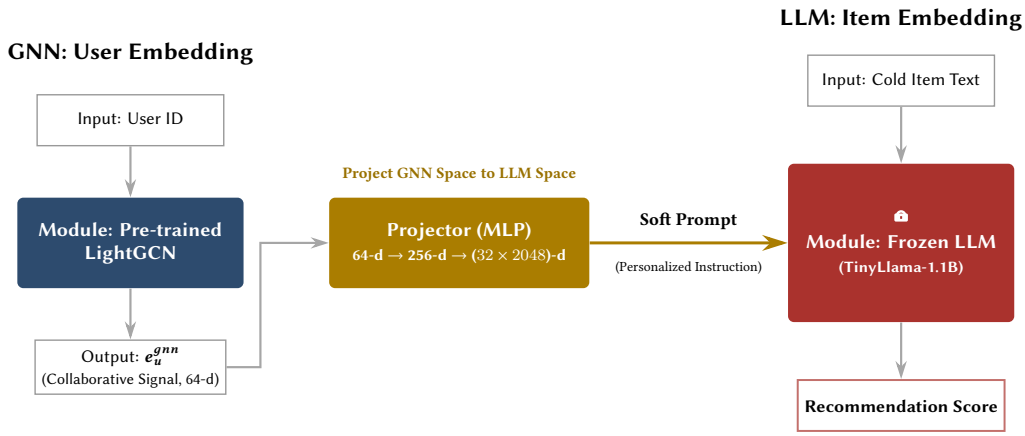

Figure 1: The overall architecture of GLCSRec. The process begins with a pre-trained LightGCN (Left) for extracting high-order user embeddings, followed by a multi-layer projector (Middle) that aligns these signals with the LLM input space. Finally, the Soft Prompts (Right) act as personalized instructions to guide the frozen LLM for item cold-start recommendation.

ing to failure in capturing user preferences. This forces the model to turn to semantic space, relying on cold-start item's metadata $\mathbf{T}_j$ (e.g., title and genres) to perform inductive reasoning through the alignment interface of the frozen LLM.

4.2 Collaborative Signal Encoder

We adopt LightGCN [3] to extract high-order user preference patterns from the historical interaction graph [7], acting as a low-pass filter to improve embedding quality in sparse graphs. Following standard settings [3], we set layer depth $K = 3$ to capture collaborative filtering while avoiding over-smoothing, adopting a layer fusion strategy to aggregate information. The final collaborative user representation $\mathbf{e}_u$ is obtained by $\mathbf{e}_u = \sum_{k=0}^{K} \alpha_k \mathbf{e}_u^{(k)}$, where $\mathbf{e}_u^{(k)}$ denotes the embedding vector at layer k and $\alpha_k = \frac{1}{K+1}$ denotes the mean pooling fusion weight. This uniform weighting strategy prevents over-smoothing while capturing high-order connectivity, allowing the final representation to capture local and global signals.

Two-Stage Training Strategy. To optimize efficiency and collaborative signal stability, we propose a decoupled two-stage training strategy. In the first stage, the GNN encoder is pre-trained independently on the observed user-item interaction graph to ensure initial user embeddings $\mathbf{e}_u$ are embedded with collaborative signals. We optimize GNN parameters using Bayesian personalized ranking (BPR) [29] to optimize pairwise ranking, making the structural foundation reliable. In the second stage, we import pre-trained user embedding vectors into the semantic alignment interface. To ensure high-order collaborative signals are not damaged, we froze the GNN propagation structure and LLM backbone parameters, optimizing only for the user embedding vector and alignment parameters. Through the soft prompt mechanism, gradients are transmitted while maintaining the LLM backbone frozen, bridging the modality gap and enabling collaborative signals to precisely map to the semantic space, complying with parameter efficiency fine-tuning (PEFT) [6, 9].

4.3 Semantic Alignment Interface

We use a multi-layer projection function $\phi(\cdot)$ composed of MLP to perform cross-modal transformation, bridging the modality gap between d-dimensional collaborative latent space and the frozen LLM high-dimensional embedding space. This multi-layer structure allows for non-linear alignment, mapping user-collaborative embedding vector $\mathbf{e}_u \in \mathbb{R}^d$ to LLM token space $\mathbb{R}^{L \times D_{LLM}}$ as defined: $\phi(\mathbf{e}_u) = \mathbf{W}_2 \cdot \sigma(\mathbf{W}_1 \cdot \mathbf{e}_u + \mathbf{b}_1) + \mathbf{b}_2$.

The transformation employs weight matrices $\mathbf{W}_1 \in \mathbb{R}^{d_h \times d}$, $\mathbf{W}_2 \in \mathbb{R}^{(L \cdot D_{LLM}) \times d_h}$, bias vectors $\mathbf{b}_1, \mathbf{b}_2$, and non-linear activation function $\sigma(\cdot)$ (ReLU) to map input $\mathbf{e}_u$ ($d = 64$) through latent space ($d_h = 256$) to intermediate features; these are reshaped into $L = 32$ continuous virtual tokens to align with LLM embedding dimension D_{LLM}. Collaborative signals are refined into soft prompts $\mathbf{P}_u \in \mathbb{R}^{L \times D_{LLM}}$, acting as personalized instructions that encode interaction history in semantic latent space [9, 30].

4.4 LLM-Integrated Recommendation

GLCSRec utilizes extensive world knowledge in pre-trained LLMs to conduct inductive inferences on cold-start items, bridging the collaborative gap. For cold-start item $j \in I_{cold}$, we retrieve textual metadata and convert it into token embeddings $\mathbf{E}_{item} \in \mathbb{R}^{T \times D_{LLM}}$. We adopt a prefix injection strategy, attaching soft prompt $\mathbf{P}_u$ before the semantic embedding sequence $\mathbf{E}_{item}$ to construct the unified input matrix $\mathbf{X}_{in} = [\mathbf{P}_u; \mathbf{E}_{item}] \in \mathbb{R}^{(L+T) \times D_{LLM}}$, where $[\cdot;\cdot]$ denotes concatenation along the sequence length dimension.

$\mathbf{X}_{in}$ conditions the causal attention mechanism, allowing the transformer backbone to attend back to virtual collaborative tokens while encoding item metadata, ensuring semantic reasoning is grounded in interaction history. Utilizing TinyLlama-1.1B [24] as the frozen backbone, we observe the hidden state of the final token $\mathbf{h}_{last} \in \mathbb{R}^{D_{LLM}}$ to extract relevance score $\hat{y}_{ui} = \mathbf{w}_{score}^{\top} \mathbf{h}_{last} + b_{score}$ via a linear score header. Trainable parameters $\mathbf{w}_{score} \in \mathbb{R}^{D_{LLM}}$ and $b_{score} \in \mathbb{R}$ are optimized using BPR loss [29] to maximize the margin between positive items and negative samples, mitigating computational costs by restricting updates to projectors and scoring heads.

4.5 Optimization Strategy

To reduce computational costs, we adopt parametric efficient fine-tuning (PEFT), freezing transformer backbone Θ_{LLM} and GNN propagation weights while limiting the trainable parameter set to $\Theta_{train} = \{\mathbf{E}_u, \Theta_\phi, \mathbf{w}_{score}, b_{score}\}$, where $\mathbf{E}_u$ denotes learnable user embedding vectors, Θ_ϕ denotes soft prompt projector parameters, and $\mathbf{w}_{score}, b_{score}$ denote scoring head parameters. We employ BPR loss [29] augmented with a regularization term: $L = L_{BPR}(\Theta_{train}) + \lambda \|\Theta_{train}\|^2$.

When Θ_{LLM} is frozen, gradients backpropagate through transformer layers to update the projector and user embeddings, ensuring end-to-end alignment between collaborative signals and semantic reasoning space [9]. PEFT enables memory efficiency by eliminating the necessity of storing gradients for the massive backbone, reducing memory cost from $O(\Theta_{LLM})$ to $O(\Theta_{train})$, where Θ_{train} constitutes $\approx$1.61% of total parameters. Freezing Θ_{LLM} enhances training efficiency by eliminating computational overhead for calculating backbone weight gradients, limiting optimization complexity to $O(\Theta_{train})$, as detailed in Table 2.

5 Experiments

This chapter explains the experimental evaluation frame-

Table 2: Parameter Configuration and Complexity Analysis (M denotes Million).

Component	Notation	Params (M)	Status
LLM Backbone	Θ_{LLM}	1,100	Frozen
User Embeddings	$\mathbf{E}_u$	0.39	Trainable
Projector	Θ_ϕ	17.36	Trainable
Scoring Head	$\mathbf{w}_{score}, b_{score}$	0.002	Trainable
Metric	**Full Fine-Tuning**	**GLCSRec (Ours)**	
Trainable Params	$\approx$ 1,118 M	$\approx$17.75 M (1.61%)	
Space Complexity	$O(\Theta_{LLM} + \Theta_{train})$	$O(\Theta_{train})$	
Time Complexity	$O(\Theta_{LLM} + \Theta_{train})$	$O(\Theta_{train})$	

work to confirm the effectiveness of the GLCSRec model. We will evaluate the performance of this framework in cold-start scenarios, where traditional collaborative filtering methods fail due to the lack of historical interaction data for new items. We recall the following RQs:

- **RQ1 (Overall Performance):** Does GLCSRec perform better than the baseline models in recommendation accuracy for cold-start items?
- **RQ2 (Ablation Study):** How much does the GNN-based collaborative signal contribute to the performance?
- **RQ3 (Robustness & Sensitivity):** How robust is the model to data sparsity, and how do hyperparameters like soft prompt length affect performance?

5.1 Experimental Settings

We utilize the widely used MovieLens-1M dataset. To ensure the quality of the graph structure and reduce interference from inactive nodes, we apply a 5-core filtering strategy, which removes users and items with fewer than 5 interactions. After preprocessing, the final dataset consists of 6,040 users and 3,706 active items, with a total of 1,000,209 interactions, yielding an interaction density of 4.47%.

Cold-Start Protocol. To simulate a realistic cold-start scenario, we adopt a item-side evaluation protocol:

1. Item split: Following the strategy in CGRC [21], we randomly select 70% of the items (2,593 items) to form the training set (warm items). The remaining 30% (1,113 items) are cold items. We split the interactions of these cold items into two parts: 15% for validation and 15% for testing. Because some items have very few interactions, they might not show up in one of the sets. That is why the actual item counts are 1,080 for validation and 1,087 for testing. The training set has 683,826 interactions, with a density of 4.37%.
2. Interaction masking: We masked interactions for all cold items in the validation and test sets ($y_{ui} = 0$) during training. This ensures that the GNN module cannot learn any structural ID embeddings for these items, forcing the model to rely on inductive reasoning via the LLM backbone.
3. Feature extraction: We employed a pre-trained SBERT model [11] to encode item metadata (i.e., titles and genres) into 384-dimensional vectors, serving the semantic context for the LLM.

Baselines. To evaluate the effectiveness of GLCSRec, we prepare reference models and cold-start baseline models, as described below:

1. Reference Models. We include two reference models to quantify the structural limitations of GNN and LLM in cold-start settings, the following models are not specifically designed to address the cold-start problem:

- LightGCN [3]: It is a representative collaborative filtering (CF) model. We set the embedding size to 64, learning rate to 1×10^{-3}, and batch size to 4096. Following the standard configuration in the RecBole framework, the model is trained for 30 epochs. Since LightGCN relies on learned ID embeddings, it is unable to generalize to unseen items, validating the failure of traditional CF under cold-start settings.
- Zero-Shot LLM: It is a content-only baseline utilizing the frozen TinyLlama-1.1B-Chat model [24]. The model is evaluated in half-precision (float16) without parameter updates. User preference is predicted via prompt-based inference (e.g., answer yes or no), where the probability of the affirmative response serves as the ranking score. This baseline represents the capability of pure LLM reasoning; however, it lacks collaborative signals from the user-item interaction graph, resulting in a "collaborative gap" that limits its personalization.

2. Cold-Start Baselines. These models are specifically designed to address the item cold-start problem:

- Cosine Similarity: It is a classic non-parametric content-based baseline. We employ the SBERT model [11] to encode item textual descriptions into 384-dimensional semantic vectors. User profiles are constructed by averaging the embeddings of their historical interactions. Recommendation scores are computed using cosine similarity between user and item representations.
- Hybrid GNN: It is a representative hybrid alignment baseline, inspired by hybrid recommendation models that combine collaborative and content signals [15, 31]. Specifically, user embeddings from GNN encoder are projected into the same semantic space as item text embeddings via a two-layer MLP ($64 \to 128 \to 384$) with ReLU activation function. The model is trained for 20

epochs using a learning rate of 1×10^{-3} and a cosine embedding loss. This baseline reflects a commonly adopted alignment paradigm that combines collaborative signals with content semantics.

- CLCRec [17]: It is a competitive cold-start recommendation method based on contrastive learning. We adopt the hyperparameters following the configurations suggested by the original authors, setting the learning rate to 0.001, regularization weight to 0.1, temperature $\tau = 2.0$, and contrastive loss weight $\lambda = 0.5$. An early stopping strategy based on validation recall is followed to select the best-performing checkpoint.

Implementation Details. We implemented GLCSRec using PyTorch 2.4.1 with Python 3.10. All model were trained on a single NVIDIA Tesla V100 GPU environment supporting CUDA 12.1. We adopt TinyLlama-1.1B as the frozen LLM backbone. For the collaborative encoder, the LightGCN embedding dimension is set to 64. The projector is a two-layer MLP with a hidden dimension of 256. As for the input settings, we set the soft prompt length to $L = 32$ and the maximum text sequence length to 128 tokens. The model is trained for 5 epochs using the Adam optimizer [32] with a learning rate of $1e-4$ and a batch size of 32.

5.2 Overall Performance (RQ1)

Table 3 summarizes the ranking performance comparison between GLCSRec and the baselines.

Table 3: Overall ranking performance comparison for cold-start items only. All models are evaluated using a fixed random seed (seed=42) to ensure identical data splitting.

Model	Recall@10	NDCG@10	Hit@10	MRR@10
LightGCN	0.0000	0.0000	0.0000	0.0000
Zero-Shot LLM	0.0067	0.0192	0.1675	0.0433
Cosine Similarity	0.0534	0.1093	0.5445	0.2449
Hybrid GNN	0.0463	0.1035	0.5156	0.2377
CLCRec	0.0200	0.0522	0.3156	0.1309
GLCSRec (proposed)	**0.0805**	**0.1724**	**0.7058**	**0.3484**

Performance Analysis. The results reveal several critical insights: The results reveal the following insights:

- LightGCN fails completely (metrics ≈ 0), confirming that pure GNNs cannot handle unseen nodes without learned ID embeddings, making it incapable of recommending cold-start items.
- Zero-shot LLM performs poorly (Recall 0.0067), worse than content-based methods. This indicates that the world knowledge of LLMs is insufficient to capture personalized cold-start recommendations.
- Cosine similarity is a content-based model, outperforms other models(Recall ≈ 0.0534), its direct semantic comparison strategy can reduce the risk of underfitting or overfitting in the cold-start setting.
- Hybrid GNN and CLCRec rely on interaction structures to align their latent spaces. In the cold-start setting, the mapping functions learned by these models on popular items cannot be effectively generalized to isolated nodes(cold-start item). This result shows the difficulty of mapping the semantic space to the collaborative space.
- GLCSRec achieves the best performance across all metrics. By integrating GNN-aggregated collaborative signals as soft prompts, our model successfully bridges the gap between collaborative signals and semantic reasoning for cold-start items.

5.3 Ablation Study (RQ2)

To validate the effectiveness of our core architectural choice, we analyze the contribution of the GNN-based collaborative signal.

Table 4: Ablation study on the collaborative signal. The variant "collaborative vs. semantic" refers to replacing structural GNN embeddings with semantic user profiles aggregated from item history.

Variant	Recall@10	NDCG@10	Hit@10	MRR@10
GLCSRec: Full	**0.0805**	**0.1724**	**0.7058**	**0.3484**
GLCSRec: Semantic-only	0.0545	0.1233	0.5798	0.2695

Impact of Collaborative Signal (collaborative vs. semantic). In our framework, the projector maps high-order user embeddings from LightGCN into the LLM's prompt space. To verify the necessity of these structural signals, we evaluated a variant, collaborative vs. semantic, where the GNN-based embeddings were replaced with semantic user profiles. These profiles were generated by averaging the embedding vectors of users' historical items encoded by LLM, representing a semantic of user preference without collaborative signal.

As shown in Table 4, removing the graph collaborative signal leads to a performance drop. This confirms that the LLM cannot effectively recommend items relying on historical semantic context. It requires collaborative signals from the GNN to capture interaction history that are not present in item descriptions, achieving more accurate cold-start recommendations.

5.4 Sensitivity and Robustness Analysis (RQ3)

5.4.1 Hyperparameter Sensitivity: Prompt Length

We explore the impact of the soft prompt length $L \in \{16, 32, 64\}$ on the recommendation performance of GLCSRec. The results are shown in Table 5, showing how different lengths of soft prompts affect the evaluation metrics.

Table 5: Performance sensitivity to Soft Prompt Length L (Seed 42).

Length (L)	Recall@10	NDCG@10	Hit@10	MRR@10
16	**0.0808**	0.1674	0.6969	0.3331
32	0.0805	0.1724	**0.7058**	0.3484
64	0.0806	**0.1760**	0.6925	**0.3635**

Increasing the prompt length L brings a slight improvement to the ranking precision metrics (NDCG@10, MRR@10). For instance, NDCG@10 rises from 0.1674 to 0.1760, indicating that longer virtual prompt sequences can provide LLMs with richer context for detailed item ordering. Other evaluation metrics (Recall@10 and Hit@10) perform consistently under different L settings, with a limited variation in overall performance. This indicates that the model has stability of prompt length, also shows that recommendation performance is mainly determined by the GNN structure rather than prompt length.

5. 4. 2 Training Stability and Convergence

Table 6 and Figure 2 summarize the computational cost and convergence of GLCSRec under different prompt lengths $L \in \{16, 32, 64\}$.

Table 6: Training convergence and time cost across different sequence lengths (L).

Config	Final Loss	Total Time	Speed (it/s)	Epoch
$L = 16$	0.0021	$\approx$ 8.8 Hours	3.37	5
$L = 32$	0.0360	$\approx$11.1 Hours	2.68	4
$L = 64$	0.3156	$\approx$16.1 Hours	1.84	3* (fluctuated)

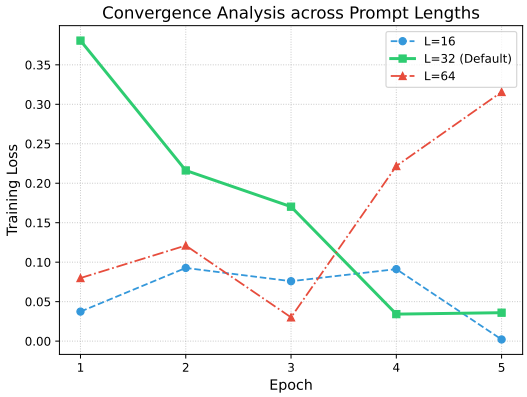

Figure 2: Training loss convergence and stability analysis over 5 epochs across different soft prompt Lengths L.

Training results indicate effective convergence; all configurations reach stable or minimal loss within 5 epochs. The default setting ($L = 32$) training loss decreases from 0.3807 to 0.0342 by the 4th epoch; negligible difference between the 4th (0.0342) and 5th epoch (0.0360) suggests stable convergence, indicating 5 epochs are sufficient. While $L = 16$ offers the highest training speed (3.37 it/s), its capacity to capture semantic collaborative signals is limited. Although the computational load of $L = 32$ increased approximately 2.3 hours more than $L = 16$, it provides stable optimization compared to $L = 64$.

The $L = 64$ shows unstable fluctuations; loss drops to 0.0301 in the 3rd epoch but rose to 0.3156 by the 5th epoch. These fluctuations, coupled with lower training speed (1.84 it/s), indicate excessively long soft prompts cause gradient noise and complicate alignment between GNN and LLM latent spaces.

5. 4. 3 Robustness to Random Sampling

To verify that the excellent performance of GLCSRec is not an accident of random sampling, we conducted experiments using 5 different random seeds and carried out statistical reliability tests, as shown in Table 7.

Table 7: Statistical Reliability of GLCSRec (Over 5 Random Seeds).

Metric	Recall@10	NDCG@10	Hit@10	MRR@10
Mean	0.0765	0.1666	0.6888	0.3497
Std Dev	$\pm$0.0052	$\pm$0.0108	$\pm$0.0224	$\pm$0.0241

The results show that all the evaluation metrics exhibit a high consistency. The low standard deviation (Recall@10 $\sigma = 0.0052$) indicates that the alignment of GNN collaborative signals with the LLM semantic space is robust and reproducible. These results indicate that GLCSRec can effectively capture user-item preferences and is not affected by random variations during the training process.

5. 4. 4 Robustness to Data Sparsity

We evaluated the robustness of GLCSRec by varying the cold-start ratio from 0.1 (Easy) to 0.5 (Hard). Table 8 summarizes the performance alongside the training data scale to provide a clear view of data scarcity.

Table 8: Robustness analysis under varying cold-start ratios. The term *train inter.* represents the total observed edges, while *density* reflects the sparsity of the training graph.

Ratio	Train Inter.	Density	Recall@10	NDCG@10	Hit@10	MRR@10
0.1 (Easy)	910,718	4.62%	**0.1569**	0.1521	0.5483	0.2611
0.3 (Default)	683,826	4.37%	0.0805	**0.1724**	**0.7058**	**0.3484**
0.5 (Hard)	485,541	4.33%	0.0471	0.1555	0.6783	**0.3484**

In contrast, at higher ratios (0.3 and 0.5), the increased data sparsity forces the projector to learn more robust translation rules from structural signals to semantic prompts. The MRR@10 remains consistently stable at 0.3484, indicating

that although the weakening GNN signal reduces the retrieval diversity, the stable reasoning ability of the LLM ensures that highly reliable semantic matching can rank accurately. In particular deployment environments, the cold-start ratio of items is often affected by external environmental factors, indicating the importance of this robustness.

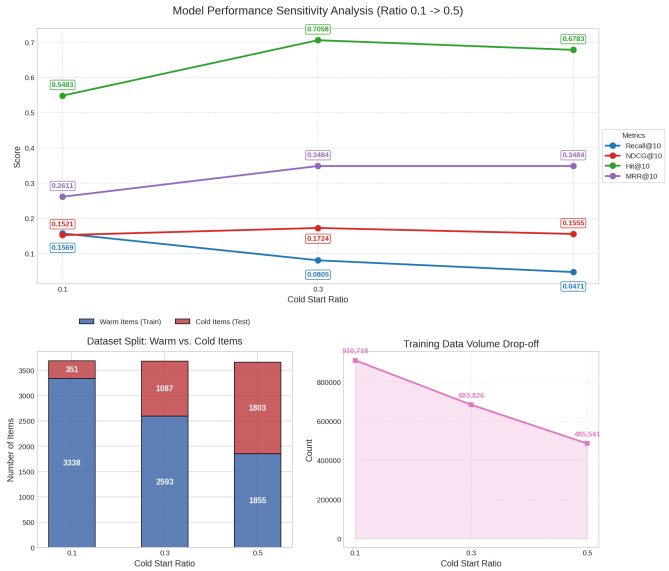

Figure 3: Model performance sensitivity analysis and data distribution between different cold-start ratios (0.1to 0.5). The above chart shows the results among various evaluation metrics as the cold-start ratio increases. The bottom-left chart shows the ratio of the warm item (training set) to the cold item (test set), and the bottom-right chart shows the decreasing trend of the total training interaction volume as the cold-start ratio changes.

6 Conclusion

In this paper, we introduced GLCSRec to address the item cold-start recommendation. This framework bridges the gap between collaborative filtering and semantic reasoning, converting the high-order interaction history captured by GNN into soft prompts. This enables the frozen LLM to inherit the knowledge of the language model while also basing the reasoning on the collaborative signals. We conduct an extensive experimental evaluation on the MovieLens-1M dataset. We demonstrated the effectiveness and robustness of the integration of GNN and LLM through soft prompt. Even when the training data was reduced by nearly 50%, the model still successfully maintain high ranking accuracy by leveraging the semantics of the backbone of LLM. This indicates that GLCSRec has achieved effective cross-modal alignment between graph structures and natural language.

Future Work. We plan to investigate knowledge distillation technology to further reduce inference delay. Additionally, integrating multi modal information (such as item images and video descriptions) can provide a more comprehensive representation for cold-start items. Finally, we aim to extend the current framework to cross-domain scenarios to assist in item recommendations in domains with extremely sparse interaction history.

References

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

[2] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web: methods and strategies of web personalization*, pages 291–324. Springer, 2007.

[3] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 639–648, 2020.

[4] Jyotirmoy Gope and Sanjay Kumar Jain. A survey on solving cold start problem in recommender systems. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pages 133–138. IEEE, 2017.

[5] Natalija Glisovic, Danica Kragic, and Martin Tegner. Item cold start in e-commerce recommender systems: A survey. volume 13, pages 164702–164722. IEEE, 2025.

[6] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. A survey on large language models for recommendation. volume 27, article number 60. Springer, 2024.

[7] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: A survey. volume 55, pages 1–37. ACM, 2022.

[8] Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V. Chawla. Can we soft prompt LLMs for graph learning tasks? In *Proceedings of the ACM Web Conference 2024 (WWW)*, pages 481–484, 2024.

[9] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3045–3059, 2021.

[10] Michael J Pazzani and Daniel Billsus. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*, pages 325–341. Springer, 2007.

[11] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982–3992, 2019.

[12] Peipei Xia, Li Zhang, and Fanzhang Li. Learning similarity with cosine similarity ensemble. volume 307, pages 39–52. Elsevier, 2015.

[13] Y. Ding, B. Wang, X. Cui, and M. Xu. Popularity prediction with semantic retrieval for news recommendation. volume 247,article number 123308, 2024.

[14] Huifeng Guo, Ruiming Tang, Yunming Forest Ye, Zhenguo Li, and Xiuqiang He. DeepFM: A factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1725–1731, 2017.

[15] Zeno Gantner, Lucas Drumond, Christoph Freudenthaler,

Steffen Rendle, and Lars Schmidt-Thieme. Learning attribute-to-feature mappings for cold-start recommendations. In *Proceedings of the 2010 IEEE International Conference on Data Mining (ICDM)*, pages 176–185. IEEE, 2010.

[16] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. MeLU: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1073–1082, 2019.

[17] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. Contrastive learning for cold-start recommendation. In *Proceedings of the 29th ACM international conference on multimedia*, pages 5382–5390, 2021.

[18] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 165–174, 2019.

[19] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[20] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, volume 30, pages 1025–1035, 2017.

[21] Jinri Kim, Eungi Kim, Kwangeun Yeo, Yujin Jeon, Chanwoo Kim, Sewon Lee, and Joonseok Lee. Content-based graph reconstruction for cold-start item recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1263–1273, 2024.

[22] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 66–74, 2020.

[23] 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). volume 36, pages 6889–6907. IEEE, 2024.

[24] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

[25] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, pages 364–381. Springer, 2024.

[26] 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 (RecSys)*, pages 1007–1014, 2023.

[27] Yangqin Jiang, Yuhao Yang, Lianghao Xia, Da Luo, Kangyi Lin, and Chao Huang. Reclm: Recommendation instruction tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15443–15459, 2025.

[28] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 4582–4597, 2021.

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 452–461, 2009.

[30] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. volume 55, pages 1–35. ACM New York, NY, 2023.

[31] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. DropoutNet: Addressing cold start in recommender systems. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, volume 30, pages 4964–4973, 2017.

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.