Content-based Graph Reconstruction for Cold-start Item Recommendation

Jinri Kim
ruth9811@snu.ac.kr
Seoul National Univ.
Seoul, Korea

Eungi Kim
kuman5262@snu.ac.kr
Seoul National Univ.
Seoul, Korea

Kwangeun Yeo
kwangeun.yeo@snu.ac.kr
Seoul National Univ.
Seoul, Korea

Yujin Jeon
jyj950309@snu.ac.kr
Seoul National Univ.
Seoul, Korea

Chanwoo Kim
chanwoo.kim@snu.ac.kr
Seoul National Univ.
Seoul, Korea

Sewon Lee
sewon0803@snu.ac.kr
Seoul National Univ.
Seoul, Korea

Joonseok Lee*
joonseok@snu.ac.kr
Seoul National Univ.
Seoul, Korea

ABSTRACT

Graph convolutions have been successfully applied to recommendation systems, utilizing high-order collaborative signals present in the user-item interaction graph. This idea, however, has not been applicable to the cold-start items, since cold nodes are isolated in the graph and thus do not take advantage of information exchange from neighboring nodes. Recently, there have been a few attempts to utilize graph convolutions on item-item or user-user attribute graphs to capture high-order collaborative signals for cold-start cases, but these approaches are still limited in that the item-item or user-user graph falls short in capturing the dynamics of user-item interactions, as their edges are constructed based on arbitrary and heuristic attribute similarity.

In this paper, we introduce Content-based Graph Reconstruction for Cold-start item recommendation (CGRC), employing a masked graph autoencoder structure and multimodal contents to directly incorporate interaction-based high-order connectivity, applicable even in cold-start scenarios. To address the cold-start items directly on the interaction graph, our approach trains the model to reconstruct plausible user-item interactions from masked edges of randomly chosen cold items, simulating fresh items without connection to users. This strategy enables the model to infer potential edges for unseen cold-start nodes. Extensive experiments on real-world datasets demonstrate the superiority of our model.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Cold-start Recommendation, Graph Neural Networks, Multi-modal, Masked Autoencoder

*Corresponding author

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07…$15.00
https://doi.org/10.1145/3626772.3657801

ACM Reference Format:
Jinri Kim, Eungi Kim, Kwangeun Yeo, Yujin Jeon, Chanwoo Kim, Sewon Lee, and Joonseok Lee. 2024. 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 '24), July 14–18, 2024, Washington, DC, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3626772.3657801

1 INTRODUCTION

Recommendation systems play an indispensable role in real-world applications such as online retail and video sharing platforms, where information overload poses a significant challenge. Collaborative Filtering (CF), the most successful technique for personalized recommender systems, predicts items tailored to specific users by capturing preference patterns commonly observed from user-item interactions. Matrix factorization (MF) [26, 40, 43], neural network-based models [22], and graph neural networks (GNNs) [21, 57, 65] have been adopted for the CF approaches.

Despite the effectiveness of CF approaches, they encounter challenges when dealing with new users or fresh content, commonly known as the *cold-start* problem–a persistent hurdle in recommender systems. Specifically, GNN-based methods usually rely on bipartite graphs formed by user-item interactions for recommendations. However, the introduction of a cold user or item lacking any interaction leads to zero information exchange with neighboring nodes due to the absence of connectivity. Consequently, recommendations become unfeasible for these cold entities.

To tackle the cold-start problem, side information has been employed [4, 25, 28, 44, 51] to represent the cold-start users or items. Despite differences in detailed methods, they commonly learn to represent users and items from their content signals in a common latent semantic space, where user tastes and item characteristics reside together. Specifically, most models first extract features from the given side information, *e.g.*, raw content or meta-data, either using pre-trained models or end-to-end training. Subsequently, as illustrated in Fig. 1, various methods have been employed to map the modality-specific content features ($\mathbf{x}$) to a common user-item embedding space, denoted by $\mathbf{z}$. Conventional approaches have adopted multi-layer perceptron (MLP) [38], autoencoders [67], or Generative Adversarial Networks (GANs) [8], as illustrated in Fig. 1(A,B). These methods are not capable of explicitly reflecting high-order collaborative signals, beyond the direct consumption, into item features.

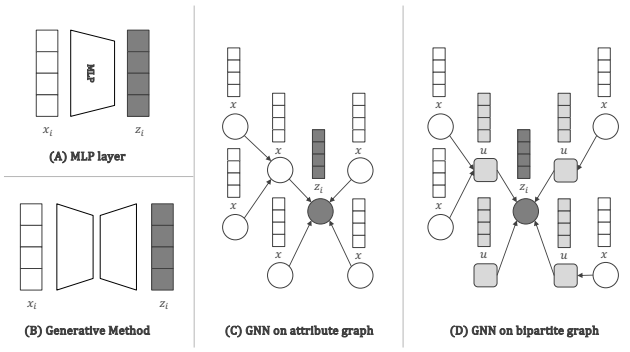


Figure 1: Various mapping methods from content features to CF representations

To capture high-order relationships among users and items more explicitly, recent efforts adopt GNNs, as illustrated in Fig. 1(C), to map content representation to CF schemes. AGNN [44] and HERS [25], for example, learn to represent items by applying message passing to adjacent nodes using graph convolution operations on an item-item attribute graph. That is, the graph consists of items as nodes and they are connected by an edge if they share common attributes, *e.g.*, same genre. As the node representations are learned on this graph by aggregating information from nearby nodes, the learned node (item) embeddings effectively capture unique characteristics of the item contents. Not just the directly adjacent nodes, however, this GNN-based approach allows access to information further away, up to the number of graph convolution layers.

The primary reason for adopting GNNs for recommendation is to leverage high-order connectivity of preference or consumption, allowing the model to consider neighbors multi-hops away from the target node [21, 57, 65]. The user-item interaction graph encodes the most primitive relationship of preference or consumption in recommendation; that is, a user A has interacted with an item B, and thus performing information exchange on this user-item graph, as illustrated in Fig. 1(D), would be the most natural.

Nevertheless, the previous works [25, 44] perform the information exchange on an item-item graph, instead of the raw user-item interactions. In fact, the item-item graph does not fully reflect the dynamics of the user-item interactions, since the edges have been constructed based on attribute similarity, which is often defined arbitrarily and heuristically. That is, the meaning of edges in the item-item attribute graph is *what we believe relevant*, relying on an assumption that items with similar attributes share comparable preferences, rather than an objective fact. The user-item interaction graph, on the other hand, conveys an obvious fact that a specific user has consumed a particular item, using an edge. Consequently, directly performing message passing on this graph would be less biased by human intuition.

Then, why have we been adopting the item-item graph? It is probably because the graph convolution is not directly applicable on the user-item interaction graph under the cold-start setting, since the cold-start user or item nodes are completely isolated from others, so no information exchange arises for them. For this reason, existing methods [25, 44] have adopted the item-item graph to connect isolated items based on the similarity of their attributes.

Then, would it never be possible to apply graph convolutions directly on the user-item interaction graph for cold-start recommendations? If this is possible, we will be able to perform cold-start recommendations without relying on arbitrary similarity measurements from item attributes. In this work, we propose a GNN-based recommendation model that is directly supervised on the user-item interaction signals in a cold-start situation, as shown in Fig. 1(D). To achieve this goal, we develop a novel framework called Content-based Graph Reconstruction for Cold-start item recommendation (CGRC). In order to connect isolated cold-start users and items, we propose a novel masked graph autoencoder model, inspired by the recent success of masked autoencoder model in different domains [12, 19, 24]. Specifically, our method learns to reconstruct plausible user-item interactions by masking a subset of existing user-item interaction edges in the training data and training the model to recover those hidden edges. Upon this simulated training, the model can infer plausible edges for an unseen cold-start node using the learned pattern. Through our masked graph autoencoder approach, our model learns to reconstruct potential user-item relationships (edges) in a self-supervised manner.

Our proposed method can be summarized as follows. First, our model chooses a subset of nodes to simulate a cold-start scenario and masks out all edges connected to them. This is in contrast to the conventional masked graph autoencoders, where several edges are randomly masked regardless of the nodes. Then, we employ an edge prediction module composed of multiple graph convolution layers to learn user preferences from their neighbors at various distances. Once trained, an edge decoder reconstructs the hidden edges, connecting the cold items to the most relevant users. Finally, on the reconstructed graph where cold items are linked to appropriate users, our model performs another set of graph convolutions to output user and item embeddings. Using these embeddings, the target user-item preference can be estimated.

Our contributions can be summarized as follows:

- To the best of our knowledge, CGRC is the first model capable of addressing the cold-start problem by directly leveraging interaction-based high-order connectivity.
- We exploit multimodal contents and graph masked autoencoder structure tailored for cold-start item recommendation to adaptively distill informative signals and facilitate self-supervised reconstruction of user-item edges.
- From extensive experiments on real-world datasets, we verify the exceptional performance of CGRC.

2 RELATED WORKS

Cold-start Recommendations. Collaborative Filtering (CF) has been proven effective in personalized recommendation systems, especially when abundant historical data is available. However, its persistent challenge lies in the cold-start problem, where there is no historical interaction data for users or items [9, 22, 26, 32, 39, 41, 43, 47, 50]. To address this problem, incorporating auxiliary information of items or users such as content features [36–38, 56], into recommendation models has been commonly employed. DropoutNet [54] randomly drops warm items during training to simulate cold-start, while diverse approaches including CB2CF [4],

Heater [70], CLCRec [61], and CCFCRec [68] integrate extra objective terms to align content information with collaborative signals. ALDI [27] employs the teacher-student method to mitigate the inherent gap between warm and cold items. Motivated by the remarkable achievements of Generative Adversarial Networks (GAN) [17] and variational autoencoder (VAE) [30], CVAR [67] has applied generative models to map content features to CF representations [2, 8, 51]. Moreover, meta-learning approaches [13, 35, 42, 69] also have been proposed to address cold-start recommendation problems.

Graph Neural Network (GNN) for Recommendations. GNNs have been successful in various graph-based learning tasks, such as node classification [6, 7, 48, 53] and link prediction [49, 71]. Inspired by the success of GNNs, many researchers have incorporated graph structures into recommendation models [10, 58, 63, 65]. GC-MC [5] introduces a graph autoencoder framework on a user-item bipartite interaction graph, treating the rating prediction task as the link prediction. NGCF [57] adopts GNN-based collaborative filtering to learn user and item embeddings. LightGCN [21] simplifies the Graph Convolution Network (GCN) by removing non-essential components for collaborative filtering.

Recent studies have successfully utilized graphs for cold- and cool-start recommendations. STAR-GCN [66] and MGL [59] employ the GCN model on user-item graphs with only a few interactions available for users and items. Strictly speaking, these approaches are applicable only to cool-start cases, but not to the complete cold-start scenarios. Our work, on the other hand, addresses a complete cold-start scenario where no interactions are available at all.

HERS [25] and AGNN [44] are two other previous methods that tackle the complete cold-start scenarios with graphs. HERS [25] models user-user and item-item relations. AGNN [44] constructs the user-user and item-item attribute graphs to learn the distribution of attributes with an extended variational auto-encoder (eVAE). Similarly to our method, they also utilize GNN-based approaches to tackle the cold-start problem. However, they apply message passing on user-user and item-item graphs rather than on the user-item interaction graph. Thus, both of them leverage high-order connectivity on the user-user or item-item graphs, which are constructed by manually and arbitrarily designed similarity metrics between users or items. Our method, on the other hand, learns high-order connectivity directly from the user-item interaction graph.

Graph Masked Autoencoder. Motivated by the recent success of generative self-supervised learning in both natural language processing [12, 45] and computer vision [3, 19, 62], similar ideas have been introduced to graph representational learning. For example, GraphMAE [24] enhances node and graph classification performance through masking and restoring the nodes. Subsequently, GraphMAE2 [23] improves its decoder to enhance the performance. These methods share commonalities with our method in that all employ masking and reconstruction strategy. However, GraphMAEs are devised for the graph classification task and mask the nodes. On the other hand, our CGRC tackles cold-start recommendation by masking and restoring the edges. Additionally, S2GAE [52] introduces a graph autoencoder structure for node classification, link prediction, and graph classification. This method also adopts masking and restoring, but it does not target the cold-start recommendation task. Recently, MAERec [64] incorporates graph masked autoencoder for sequential recommendations. Building upon this line, our approach devises a graph masked autoencoder structure, further advancing it with multimodal representation to address the comprehensive cold-start item recommendation scenario.

3 THE PROPOSED METHOD: CGRC

In this work, we focus on recommending completely new items without any interactions. Our proposed model architecture is comprised of three components: Node Encoder, Graph Reconstructor, and Cold-start Recommender, as illustrated in Fig. 2. Once users and items are encoded with their multimodal features, some items are randomly chosen, and all edges connected to them are masked out to simulate a cold-start scenario. The model learns to reconstruct those edges, so that they can generalize it to unseen cold-start items at testing. Then, within the cold-start recommender, user and item graph embedding is obtained through the reconstructed graph, exploiting interaction-based high-order connectivity.

3.1 Problem Formulation

Let $\mathcal{U}$ and $\mathcal{I}$ be a set of n users and a set of m items, respectively. Let $\mathbf{R} \in \{0,1\}^{n \times m}$ be a binary interaction matrix of n users and m items, where $\mathbf{R}_{ui} = 1$ if the user $u \in \mathcal{U}$ has interacted with the item $i \in \mathcal{I}$, and $\mathbf{R}_{ui} = 0$ otherwise. Each item i is accompanied with C content elements, *e.g.*, raw content features such as video, image, audio, and text, or meta-data like genre, creator, year, and so on. We construct a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of the user-item interactions, where $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ and $\mathcal{E} = \mathbf{R}$, the user-item interactions. We denote $\mathcal{N}_u$ as the set of items that the user u has interacted with. Likewise, $\mathcal{N}_i$ is the set of users who have interacted with the item i.

We aim to tackle the cold-start item recommendation task by estimating the preference score $\hat{\mathbf{R}}_{ui'}$ for a user $u \in \mathcal{U}$ and item $i' \notin \mathcal{I}$ (that is, the user is not a cold-start but only the target items are cold-start). Note that the proposed idea is equally applicable to the cold-start users as well, if informative side information for users is provided. We showcase only the cold-start item cases in this paper, mainly due to the lack of publicly available meaningful side information for users, concerning their privacy.

3.2 Node Encoder

3.2.1 User and Multi-modal Item Encoding Layer. In the user encoding layer, following the traditional collaborative filtering models, the user u is represented through an embedding layer, yielding $\mathbf{e}_u \in \mathbb{R}^d$, where d is the embedding size.

To address cold-start items, however, the item encoding layer leverages C content signals (or side information) to represent an item i. These signals are processed by modality-specific encoders, such as ViT [14] (visual), BERT [12] (text), and AST [16] (audio), followed by a learnable linear projection. Each modality representation is denoted as $\mathbf{f}_i^{(c)} \in \mathbb{R}^{d_c}$, where $c = 1, ..., C$ identifies each content feature, and d_c is the embedding dimensionality for the modality c.

Subsequently, these content features are aggregated (*i.e.*, concatenated and followed by a fully-connected (FC) layer) to produce

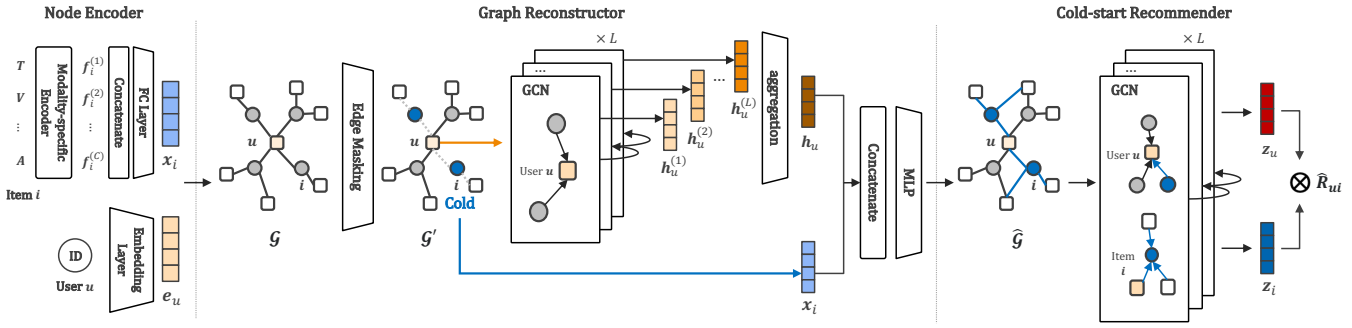


Figure 2: Overall Architecture

the final content-based item embedding $\mathbf{x}_i \in \mathbb{R}^d$, where d is the dimensionality of the final embedding:

$$\mathbf{x}_i = \mathbf{W}\left(\mathbf{f}_i^{(1)}; \mathbf{f}_i^{(2)}; \ldots; \mathbf{f}_i^{(C)}\right) + \mathbf{b}, \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{d \times \sum_c d_c}$ and $\mathbf{b} \in \mathbb{R}^d$.

3.2.2 Multi-modal Alignment Loss. In a simple approach, we expect that the projection layer in Eq. (1), which integrates diverse multimodal content features, will autonomously learn multimodal relationships within the same item. While this simple approach reasonably combines multi-modal signals, we delve deeper into leveraging multi-modal relationships through self-supervision. Specifically, we apply contrastive loss [11] on all item embeddings in each mini-batch, aiming to enhance the similarity between embeddings for the same item across different modalities, while minimizing similarities between all other combinations. Formally, the multi-modality loss $\mathcal{L}$ between two modalities a and b for the same item i is given by

$$\mathcal{L}_{\mathrm{M}} = -\sum_{a=1}^{C-1}\sum_{b=a+1}^{C} \log \frac{\exp(\mathrm{sim}(\mathbf{f}_i^{(a)}, \mathbf{f}_i^{(b)})/\tau_{ssl})}{\sum_j^B \exp(\mathrm{sim}(\mathbf{f}_i^{(a)}, \mathbf{f}_j^{(b)})/\tau_{ssl})}, \tag{2}$$

where $\mathrm{sim}(\cdot)$ computes the similarity between two vector representations of modalities using the inner product; τ_{ssl} represents a temperature hyperparameter, and B denotes the number of examples within a batch.

3.3 Graph Reconstructor

The key innovation of this work lies in this graph reconstruction step. Building upon the success of masked autoencoding in the fields of CV [19], NLP [12] and GNN [24], our approach predicts potentially relevant users to a cold-start item, simulating cold-start situations with the given user-item interaction graph $\mathcal{G}$ by masking out all edges for a subset of randomly chosen item nodes. The model is trained to estimate the existence of missing edges to other remaining nodes, based on the content signals of the nodes. In this way, the model is expected to learn the underlying relationship between the content signals and the connectivity within the graph, which corresponds to the collaborative signals in recommendation systems.

3.3.1 Masked Graph $\mathcal{G}'$. From the user-item bipartite graph $\mathcal{G}$, we construct a masked graph, denoted by $\mathcal{G}'$, to simulate the cold-start scenarios. Specifically, we first randomly choose some item nodes on $\mathcal{G}$ as cold items, $\mathcal{I}_{\text{cold}} \subset \mathcal{I}$, with the probability of ρ. Then, we mask out all the edges associated with these items in $\mathcal{I}_{\text{cold}}$:

$$\mathcal{E}_{\text{masked}} = \{(u,i) \in \mathcal{E} \mid u \in \mathcal{U}, i \in \mathcal{I}_{\text{cold}}\}. \tag{3}$$

The masked graph is formed as $\mathcal{G}' = (\mathcal{V}, \mathcal{E} - \mathcal{E}_{\text{masked}})$. (Note that the masked items $\mathcal{I}_{\text{cold}}$ still exist in the graph $\mathcal{G}'$; just being isolated from other nodes to simulate cold-start.) From the recommendation perspective, these masked edges indicate *hidden relevance* between a cold item and a user, and we train the model to relate these hidden relevance using the content signals, which are available for the cold items. Once trained, the model is expected to discover potentially relevant users for an unseen cold-start item, generalizing the learned patterns from the contents to the user-item preference.

3.3.2 Graph Convolution Layers. Once the masked graph $\mathcal{G}'$ is constructed, we apply graph convolution layers (*e.g.* LightGCN [21], NGCF [57]) to learn users and items representations by incorporating both their own representations and those of their neighboring nodes. This enables us to capture user preferences at multi-hop neighbors. For instance, to generate the user representation, the first layer aggregates all the items preferred by the user (user-item), and the second layer integrates other users who shared preferences for commonly preferred items (user-item-user). Extending the number of layers allows a more comprehensive understanding of the user preferences, by drawing insights from more distant neighbors.

At each layer, following LightGCN [21], a graph convolution layer updates node embeddings by

$$\mathbf{H}^{(\ell)} = (\mathbf{D}'^{-\frac{1}{2}}\mathbf{A}'\mathbf{D}'^{-\frac{1}{2}})\mathbf{H}^{(\ell-1)}, \tag{4}$$

where $\mathbf{A}' = \begin{pmatrix} \mathbf{0} & \mathbf{R}' \\ \mathbf{R}'^{\top} & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{(n+m)\times(n+m)}$ is the adjacency matrix of the graph $\mathcal{G}'$ and $\mathbf{D}' \in \mathbb{R}^{(n+m)\times(n+m)}$ denotes the diagonal degree matrix, where each diagonal element $\mathbf{D}'_{ii}$ represents the number of non-zero entries in the i-th row of the adjacency matrix $\mathbf{A}'$. Here, $\mathbf{R}' \in \mathbb{R}^{n\times m}$ is the user-item interaction matrix corresponding to the masked graph $\mathcal{G}'$. Note that our method is not confined to the specific method of LightGCN; other GCN variants can be applied as well, *e.g.*, NGCF [57]. Stacking L graph convolution layers, we produce L node representations $\{\mathbf{H}^{(1)}, \mathbf{H}^{(2)}, \ldots, \mathbf{H}^{(L)}\}$, where each $\mathbf{H}^{(\ell)} \in \mathbb{R}^{(n+m)\times d}$ captures the neighborhood structure within ℓ hops for each node.

Similarly to the conventional GNN encoders, each user node $u \in \mathcal{U}$ is initiated with $\mathbf{h}_u^{(0)} = \mathbf{e}_u$ and each item node $i \in \mathcal{I}$ is initiated with $\mathbf{h}_i^{(0)} = \mathbf{x}_i$. However, since we perform the graph convolution operations on the masked graph $\mathcal{G}'$, not on the original graph $\mathcal{G}$, the cold item nodes ($\mathcal{I}_{\text{cold}}$) are frozen to their initialized values, as they are disconnected from other nodes in $\mathcal{G}'$ and thus no information exchange arises with their neighbors. Other warm nodes, including users and warm items ($\mathcal{I} - \mathcal{I}_{\text{cold}}$), are updated by performing graph convolution operations in the conventional way, excluding the edges connected to a cold-start item node. At each layer ℓ, the node embedding for a user u and an item i is denoted by $\mathbf{h}_u^{(\ell)}$ and $\mathbf{h}_i^{(\ell)}$, respectively, for $\ell = 1, ..., L$. In the end, the GNN encoder generates $\left\{\mathbf{h}_u^{(\ell)}, \mathbf{h}_i^{(\ell)}\right\}_{\ell=1}^{L}$ for all $u \in \mathcal{U}$ and $i \in \mathcal{I}$, but the cold items in $\mathcal{I}_{\text{cold}}$ are not updated and keep the initalized value, $\mathbf{x}_i$.

3.3.3 Edge Predictor. The edge predictor, depicted in Fig. 2, estimates the probability of a connection between a user $u \in \mathcal{U}$ and each cold item $i \in \mathcal{I}_{\text{cold}}$, determining whether they should be connected or not, leveraging the user and item embeddings learned at various levels, $\left\{\mathbf{h}_u^{(\ell)}, \mathbf{h}_i^{(\ell)}\right\}_{\ell=1}^{L}$.

Specifically, the edge predictor is a binary classifier estimating the score $p(u, i)$ of an edge to exist between a user u and an item i. There are multiple options to build this predictor [31, 52, 55]. As its input, we take the L embeddings, $\mathbf{h}_u^{(\ell)}$ and $\mathbf{h}_i^{(\ell)}$ for $\ell = 1, ..., L$, and take their mean to aggregate them:

$$\mathbf{h}_u = \frac{1}{L}\sum_{\ell=1}^{L}\mathbf{h}_u^{(\ell)}, \quad \mathbf{h}_i = \frac{1}{L}\sum_{\ell=1}^{L}\mathbf{h}_i^{(\ell)} \overset{\forall i \in \mathcal{I}_{\text{cold}}}{=} \mathbf{x}_i. \tag{5}$$

Note that we do not need to take a mean for the item features, since this edge predictor takes only a cold item as its input, and cold item features $\mathbf{h}_i^{(\ell)}$ are always the same regardless of $\ell = 1, ..., L$. There are multiple options for the classifier as well; *e.g.*, an inner-product between the user and item embeddings, $p(u, i) = \langle \mathbf{h}_u, \mathbf{h}_i \rangle$, or a feed-forward network on top of the concatenation of the user and item embeddings, $p(u, i) = \text{MLP}([\mathbf{h}_u; \mathbf{h}_i])$, where $\mathbf{h}_u$ and $\mathbf{h}_i$ stand for the final embeddings of the user u and item i. We empirically validate aforementioned design choices in our ablation study section 4.3.4.

3.3.4 Reconstruction Loss. At training, we predict $p(u, i)$ for all $u \in \mathcal{U}$ and $i \in \mathcal{I}_{\text{cold}}$, and for each cold item, we connect the top-K user nodes with the highest predicted scores, where K is a hyperparameter. As a result, we build a *reconstructed graph*, denoted by $\hat{\mathcal{G}} = (\mathcal{V}, \mathcal{E}')$, where $\mathcal{E}' = (\mathcal{E} - \mathcal{E}_{\text{masked}}) \cup \mathcal{E}_{\text{recon}}$, where $\mathcal{E}_{\text{recon}}$ is the set of reconstructed edges between cold items and user. The training loss for reconstructing masked edges is

$$\mathcal{L}_{\text{E}} = -\frac{1}{|\mathcal{E}_{\text{masked}}|}\sum_{(u,i)\in\mathcal{E}_{\text{masked}}} \log\frac{\exp(p(u,i))}{\sum_{j\in\mathcal{I}_{\text{cold}}\setminus\mathcal{N}_u}\exp(p(u,j))}, \tag{6}$$

where $p(u, i)$ is the estimated link score between the user u and a positive cold item i, while $p(u, j)$ is the score between the user u and a cold item j selected with negative sampling. In contrast to conventional graph autoencoders, which typically focus on reconstructing edges for homogeneous nodes [52, 64], our proposed model is designed to reconstruct masked user-item interaction edges.

3.4 Cold-start Recommender

3.4.1 Graph Convolution Layers on the Reconstructed Graph $\hat{\mathcal{G}}$. After reconstructing the edges for the simulated cold items, we employ another set of graph convolution layers, similar to the one used in Sec. 3.3.2, on the reconstructed graph $\hat{\mathcal{G}}$ to obtain the final user and item embeddings, $\mathbf{z}_u$ and $\mathbf{z}_i$, respectively.

Specifically, we iteratively perform message-passing on the graph $\hat{\mathcal{G}}$ using Eq. (4), for L times. This process generates user and item embeddings at each layer, denoted by $\mathbf{z}_u^{(\ell)}$ and $\mathbf{z}_i^{(\ell)}$ for $\ell = 0, ..., L$. Then, the final user and item representations $\mathbf{z}_u$, $\mathbf{z}_i$ are produced either by concatenating all $L+1$ embeddings (NGCF [57]) or by taking an average of them (LightGCN [21]). Subsequently, we predict the preference score $\hat{\mathbf{R}}_{ui}$ by

$$\hat{\mathbf{R}}_{ui} = \mathbf{z}_u^{\top}\mathbf{z}_i. \tag{7}$$

The rationale behind employing a GNN on the reconstructed graph $\hat{\mathcal{G}}$ lies in enhancing the information flow from cold item embeddings to multi-hop neighbors. This effectively utilizes interaction-based high-order connectivity, which is the most advantageous feature of GNN in recommender system [21, 57]. The detailed illustration of this process can be found in Fig. 2. This approach enables the creation of more fine-grained final user and item embeddings, thereby enabling high-quality recommendations using cold items. The empirical validation of the effects of applying GNN on the reconstructed graph is presented in our ablation study 4.3.3.

Note that this step is performed on the reconstructed graph $\hat{\mathcal{G}}$ only at testing. This message-passing step is trained on the original complete graph $\mathcal{G}$ due to the instability of the reconstruction step. Since a reconstructed edge does not necessarily indicate actually relevant user-item interaction at the early stage of training, message-passing on this unstable reconstructed graph would lead to significantly noisy node embeddings. Thus, similarly to the teacher forcing [33], we train the graph convolution itself on the original graph during training.

3.4.2 Rating Ranking Loss. Utilizing the user $\mathbf{z}_u$ and item $\mathbf{z}_i$ embeddings, we subsequently train the model to assign higher scores to preferred items and lower scores for non-preferred ones. We employ the contrastive loss [20, 34], a prevalent approach in representation learning. Specifically, for each user, the items they prefer are identified as positives, and conversely, all other items in the minibatch are considered negatives. The model is optimized to maximize the positive pairs and to minimize the negative pairs. Rating Ranking Loss $\mathcal{L}_{\text{R}}$ for each mini-batch is defined by

$$\mathcal{L}_{\text{R}} = -\sum_{i\in\mathcal{N}_u}\log\frac{\exp(\text{sim}(\mathbf{z}_u, \mathbf{z}_i)/\tau)}{\sum_{j\notin\mathcal{N}_u}^{B}\exp(\text{sim}(\mathbf{z}_u, z_j)/\tau)} \tag{8}$$

where $\text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^{\top}\mathbf{b}$, τ is a temperature hyperparameter, and B denotes the number of examples within a mini-batch.

3.5 Overall Loss Function

The overall loss function linearly combines the three losses presented above:

$$\mathcal{L} = \lambda_{\text{M}}\mathcal{L}_{\text{M}} + \lambda_{\text{E}}\mathcal{L}_{\text{E}} + \mathcal{L}_{\text{R}}, \tag{9}$$

where $\mathcal{L}_{\text{M}}$ is the multi-modal alignment loss within the Node Encoder (Sec. 3.2), $\mathcal{L}_{\text{E}}$ is the reconstruction loss within the Graph

Table 1: Statistics of the datasets with multi-modal item Visual (V), Acoustic (A), Textual (T) contents

Dataset	TikTok			ML-1M			Yahoo Movie		
Modality	V	A	T	V	A	T	V	A	T
Emb. Dim	128	128	768	768	768	768	768	768	768
User	9,308			6,039			5,353		
Item	6,710			2,819			2,739		
Interactions	68,722			730,012			70,244		
Density	0.11%			4.29%			0.48%		

Reconstructor (Sec. 3.3), and $\mathcal{L}_{\mathrm{R}}$ signifies the recommendation loss in the Recommender (Sec. 3.4). $\lambda_{\{\mathrm{M,\,E}\}}$ are hyperparameters controlling the relative importance among the losses.

4 EXPERIMENTS

We conduct extensive experiments to verify the efficacy of CGRC on multiple cold-start recommendation datasets. Our experiments aim to answer the following research questions:

- **RQ1**: How does CGRC perform compared to the state-of-the-art cold-start item recommendation models?
- **RQ2**: How do different components contribute to the performance of CGRC?
- **RQ3**: How do the hyperparameters affect the performance of CGRC?
- **RQ4**: How does our CGRC model perform qualitatively?

4.1 Experimental Settings

4.1.1 Datasets. We conduct experiments on three widely used video domain datasets: MovieLens-1M [18], Tiktok [60], and Yahoo Movie[1]. These datasets are chosen as they contain not only user-item interaction records but also rich content signals, *e.g.*, visual scenes, textual plot, metadata like genre, or audio. The statistics of these datasets are detailed in Table 1.

For the TikTok dataset, the interaction data is provided without explicit ratings (*i.e.*, click), so we assume all the interactions to be implicit. On the other hand, both MovieLens and Yahoo Movie provide explicit ratings ranging from 1 (least preferred) to 5 (most preferred). We convert these ratings to implicit ones using 3 as the threshold, following previous works [4].

The items are randomly split into training, cold validation, and cold test sets in a 70:15:15 ratio, following CCFCRec [68]. We use the cold validation set to determine hyperparameters and the cold-test set to evaluate the final performance.

4.1.2 Content Features. In our experiments, we use visual, text, and audio features. The Tiktok dataset provides all three types of features, so we take the preprocessed version provided by MMSSL[2]. According to MMSSL, the visual and acoustic features of micro-videos are extracted and published without providing the raw data (probably due to copyright), while the textual embeddings are encoded with Sentence-BERT [46].

On the other hand, the MovieLens and Yahoo Movie datasets offer limited content information, requiring additional data collection. For visual content of these movie datasets, we use movie trailers provided by MovieLens [1] and MovieNet[3] due to copyright issues. From each video, frames of dimensions 224×224 are sampled at 2 fps. To exclude potentially irrelevant content such as age rating screens or ending credits, the first and last 10% of sampled frames are omitted. We extract frame-level features from a pre-trained Vision Transformer (ViT) [14], then subsequently average these features to obtain video-level representations. Regarding textual content in the movie datasets, we utilize movie synopses obtained from `imdb.com` for MovieLens. The Yahoo Movie dataset inherently includes synopses. These synopses, typically comprising 2-3 sentences summarizing the movie overview, are tokenized using uncased $\mathrm{BERT}_{\mathrm{BASE}}$ [12]. For audio content in the movie datasets, we extract audio segments lasting 32 seconds from the movie trailers. Subsequently, the Audio Spectrogram Transformer (AST) [16] is employed to derive audio features.

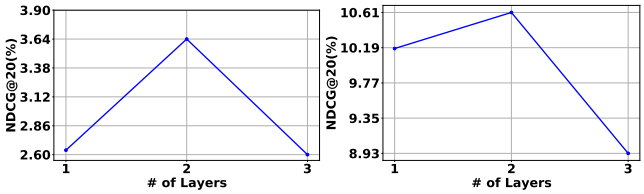
4.1.3 Baseline Methods. We compare CGRC with the following cold-start item recommendation methods: **DropoutNet [54]**, which adapts to cold-start by intermittently omitting warm items at training, indirectly converting the content of cold-start items into warm embeddings, **Heater [70]**, which jointly trains both warm and cold embeddings without suffering from error superimposition problem through randomized training and mixture-of-experts transformation, **AGNN [44]**, which exploits the attribute graph, **CLCRec [61]**, which incorporates collaborative signals in the content representations for both warm and cold-start items with contrastive learning, **CVAR [67]**, which employs conditional variational autoencoders to warm up cold-start item embeddings, and **CCFCRec [68]** exploiting co-occurrence collaborative signals and item attributes.

4.1.4 Evaluation Protocols. We evaluate the recommendation performance by ranking unseen items for each user in a held-out test set, then comparing the top-k items from the ranked list with the items that the user actually gave positive feedback to. We adopt three widely used metrics {Precision, Recall, NDCG}@k with $k \in \{5, 10, 20\}$.

4.1.5 Implementation Details. To train CGRC, we use Xavier initialization [15] and Adam optimizer [29]. We use a batch size of 4096, and initial learning rates are set to 10^{-3} for TikTok and ML-1M, and to 10^{-4} for Yahoo Movie. We vary the embedding dimensionality $d \in \{8, 16, 32, 64, 128, 256, 512\}$. To create the masked graph $\mathcal{G}'$, we try a mask ratio of {0.1, 0.3, 0.5, 0.7, 0.9} across all datasets. For the loss balancing factors $\lambda_{\{\mathrm{M,E}\}}$, we vary from {0, 0.5, 1.0, 2.0}, respectively. For the number of edges K to recover for each cold item from $\hat{\mathcal{G}}$, we cross-validate within {5, 10, 20, 30, 40, 50}. We explore the number of graph convolution layers L within {0, 1, 2, 3, 4, 5}. This search is conducted on both the masked graph $\mathcal{G}'$ and the reconstructed graph $\hat{\mathcal{G}}$ to identify the ideal number of layers. The best choices are empirically discovered in Sec. 4.3.2 and 4.3.3.

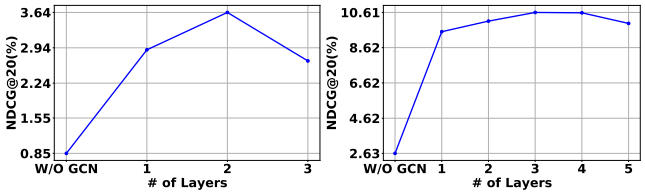
Among many other GNN encoders, we select LightGCN [21] for our graph convolution method due to its concise and simple structure. Both τ_{ssl} and τ are uniformly set at 0.5 for all datasets. To ensure fairness, we configure the baseline methods' hyper-parameters to their optimal settings, determined by cross-validation.

[1] https://webscope.sandbox.yahoo.com/
[2] https://github.com/HKUDS/MMSSL
[3] https://movienet.github.io/



(a) TikTok (b) Yahoo Movie

Figure 3: Effect of Number of Layers on $\mathcal{G}'$



(a) TikTok (b) Yahoo Movie

Figure 4: Effect of Prediction Method

4.2 Overall Performance (RQ1)

We compare our proposed model with six other state-of-the-art cold-start recommendation methods, presenting the results in Table 2. CGRC consistently outperforms the baselines in most metrics across all three datasets, confirming the superiority of CGRC in cold-start item recommendation attributed to its utilization of interaction-based high-order connectivity.

Notably, CGRC achieves even stronger performance on TikTok and Yahoo Movie, showing higher sparsity (see Table 1). This observation demonstrates that our method is particularly stronger in cold or cool start cases. On a denser dataset, MovieLens, CVAR demonstrates relatively comparable performance with ours especially when k is small. However, CGRC still maintains competitive performance even on this denser case.

This superiority stems from the fact that, compared with contrastive learning-based methods (CLCRec and CCFCRec) and attribute graph-based methods (AGNN), CGRC explicitly incorporates high-order collaborative signals into item features.

4.3 Ablation Study (RQ2)

We conduct ablation studies to explore how each component of our CGRC contributes to its overall performance. We utilize two datasets, TikTok and Yahoo Movies, for ablation studies.

4.3.1 Effect of Masked Graph Autoencoder Structure. We first investigate the impact of the masked graph autoencoder (MGAE) architecture on recommendation performance.

In Table 3, the Graph Encoder refers to the utilization of graph convolution layers, *e.g.*, LightGCN [21], to encode the masked graph $\mathcal{G}'$. Without the Graph Encoder, the user representation is directly obtained from the user embedding e_u without exchanging information with neighboring nodes. The MLP decoder denotes the MLP layer used to calculate the edge probability in Fig. 2. Without the MLP decoder, the edge probability is determined by the inner product of the user and item representations.

According to the results in Table 3, the best performance on both datasets is achieved when both the graph encoder and the MLP decoder are used. This combination appears to optimize the effectiveness of the MGAE structure in the context of cold-start recommendations, highlighting the benefits of a comprehensive encoding and decoding process.

4.3.2 Effect of Number of Layers on Masked Graph $\mathcal{G}'$. CGRC employs L graph convolution layers on the masked graph $\mathcal{G}'$ to derive the node representations (Sec. 3.3.2). We investigate the optimal number of these layers on the overall recommendation performance. Fig. 3 illustrates the performance with a varied number of GCN layers on the two datasets. The best performance is achieved with 2 layers for both TikTok and Yahoo Movie datasets, aligning with the findings in the LightGCN paper [21].

4.3.3 Effect of Prediction Method on Reconstructed Graph $\hat{\mathcal{G}}$. Recall that CGRC becomes capable of recommending isolated cold items by generating the reconstructed graph $\hat{\mathcal{G}}$. With the newly discovered K edges from these cold items, they become a part of the graph and eventually are recommended to the connected users. In order to further leverage high-order connectivity with the newly connected cold items, we apply additional graph convolution layers on the reconstructed graph $\hat{\mathcal{G}}$, as mentioned in Sec. 3.4.1.

Fig. 4 illustrates the effect of different numbers of additional GCN layers on the two datasets. Consistent with the findings in the LightGCN paper [21], the best recommendation performance is achieved with 2 layers for the TikTok dataset and 3 layers for the Yahoo Movie dataset. In this experiment, we confirm that having at least one layer of the graph convolutions on the reconstructed graph indeed helps.

4.3.4 Effect of Aggregation Methods. Selecting the optimal decoder for a given task is a critical aspect of the masked graph autoencoder framework [23, 24]. Therefore, we investigate multiple design choices for our masked graph autoencoder framework to obtain a user and edge representation. Specifically, we try a couple of approaches to aggregate L user representations that are produced at each graph convolution layer, in our edge predictor (Sec. 3.3.3): 1) concatenation of the L representations (Concat), and 2) taking an average of them (Mean). Similarly, we also try two aggregation approaches to combine the final user and item representations, $\mathbf{h}_u$ and $\mathbf{h}_i$: 1) concatenation (Concat), and 2) element-wise multiplication (Multiply). Once they are aggregated, the combined embedding goes through an MLP layer.

The results in Table 4 highlight the superior performance of the mean pooling across the L user node representations, followed by concatenation with the item representation to form an edge representation on both datasets. Despite its simplicity, the mean pooling presents stronger robustness to noisy features across different representation layers. Additionally, the concatenation approach preserves diverse information, making it well-suited for representing edge features involving both user and item representations.

4.3.5 Ablation on Content Features. In our study, we leverage multimodal features to represent items, incorporating video (V), audio (A), and text (T) modalities. This naturally raises the question: which modality has the most significant impact on recommendations? To explore this, we conduct a modality ablation study on the two datasets in Table 5.

Table 2: Overall performance (%) comparison with cold-start recommendation models on three datasets. The best performance is boldfaced, and the second best is underlined.

Dataset	Method	Precision (↑)			Recall (↑)			NDCG (↑)		
		@5	@10	@20	@5	@10	@20	@5	@10	@20
TikTok	DropoutNet [54]	0.49	0.42	0.46	0.80	1.48	2.68	0.81	0.98	1.28
	Heater [70]	0.44	0.41	0.39	0.83	1.45	2.72	0.78	0.87	1.20
	AGNN [44]	0.57	0.65	0.53	1.12	3.64	4.60	1.15	1.94	2.15
	CLCRec [61]	0.44	0.41	0.34	1.34	2.08	2.90	1.01	1.27	1.49
	CVAR [67]	0.51	0.46	0.47	0.81	1.50	2.89	0.73	0.93	1.28
	CCFCRec [68]	0.61	0.55	0.49	1.17	1.89	3.40	1.11	1.33	1.67
	CGRC (Ours)	**0.99**	**0.94**	**0.80**	**2.87**	**4.75**	**7.83**	**2.22**	**2.88**	**3.64**
MovieLens	DropoutNet [54]	6.01	5.74	5.35	1.80	3.48	6.21	5.65	5.94	6.63
	Heater [70]	4.85	5.38	5.59	1.33	3.21	6.77	5.46	6.07	6.81
	AGNN [44]	9.96	9.47	7.33	3.15	5.77	8.57	9.17	9.97	9.70
	CLCRec [61]	10.44	9.61	8.65	3.72	6.75	11.50	10.51	10.72	11.76
	CVAR [67]	14.70	12.71	10.95	4.61	7.49	12.78	**16.25**	15.23	15.50
	CCFCRec [68]	11.95	11.15	10.34	3.46	6.38	11.76	12.42	12.31	13.26
	CGRC (Ours)	**15.21**	**13.98**	**12.36**	**5.38**	**9.73**	**16.60**	16.01	**16.04**	**17.22**
Yahoo Movie	DropoutNet [54]	1.19	1.08	1.09	1.93	3.30	6.49	1.76	2.24	3.23
	Heater [70]	0.88	0.92	0.93	1.40	2.74	5.59	1.21	1.69	2.72
	AGNN [44]	2.78	2.03	1.55	5.25	6.94	9.99	5.65	5.91	6.76
	CLCRec [61]	2.81	2.25	1.82	4.22	6.85	10.92	4.38	5.13	6.35
	CVAR [67]	1.89	1.88	1.50	2.04	4.20	7.16	2.31	3.05	3.91
	CCFCRec [68]	2.49	1.95	1.64	3.00	4.94	8.60	3.68	4.11	5.19
	CGRC (Ours)	**4.20**	**3.71**	**2.96**	**7.04**	**12.37**	**19.35**	**6.63**	**8.45**	**10.61**

Table 3: Effect of MGAE Structure

Graph Enc	MLP Dec	TikTok		Yahoo Movie	
		N@10	N@20	N@10	N@20
✗	✗	2.04	2.68	5.77	7.60
✓	✗	1.86	2.37	7.45	9.50
✗	✓	2.52	3.15	6.88	8.99
✓	✓	**2.88**	**3.64**	**8.45**	**10.61**

Table 4: Comparison for user and edge aggregation methods

User Agg	Edge Agg	TikTok		Yahoo Movie	
		N@10	N@20	N@10	N@20
Concat	Multiply	1.60	2.08	7.46	9.43
Concat	Concat	1.45	1.96	8.19	10.47
Mean	Multiply	1.58	2.13	7.43	9.37
Mean	**Concat**	**2.88**	**3.64**	**8.45**	**10.61**

Table 5: Effect of input feature modalities

Modalities			TikTok		Yahoo Movie	
V	A	T	N@10	N@20	N@10	N@20
✓			1.77	2.26	6.76	8.59
	✓		1.20	1.53	2.74	3.96
		✓	1.13	1.45	4.76	6.48
✓	✓		2.47	3.15	6.54	8.53
✓		✓	1.95	2.45	7.79	10.18
	✓	✓	1.65	2.16	6.09	8.00
✓	✓	✓	**2.88**	**3.64**	**8.45**	**10.61**

The findings reveal that the optimal performance on both datasets is obtained when all modalities (V, A, T) are utilized together. Specifically, on the TikTok dataset, the combination of video and audio modalities achieves the second-highest performance, followed by video and text, and then video alone. For the Yahoo Movie dataset, the video and text combination ranks second in performance, succeeded by video and audio, and then video alone.

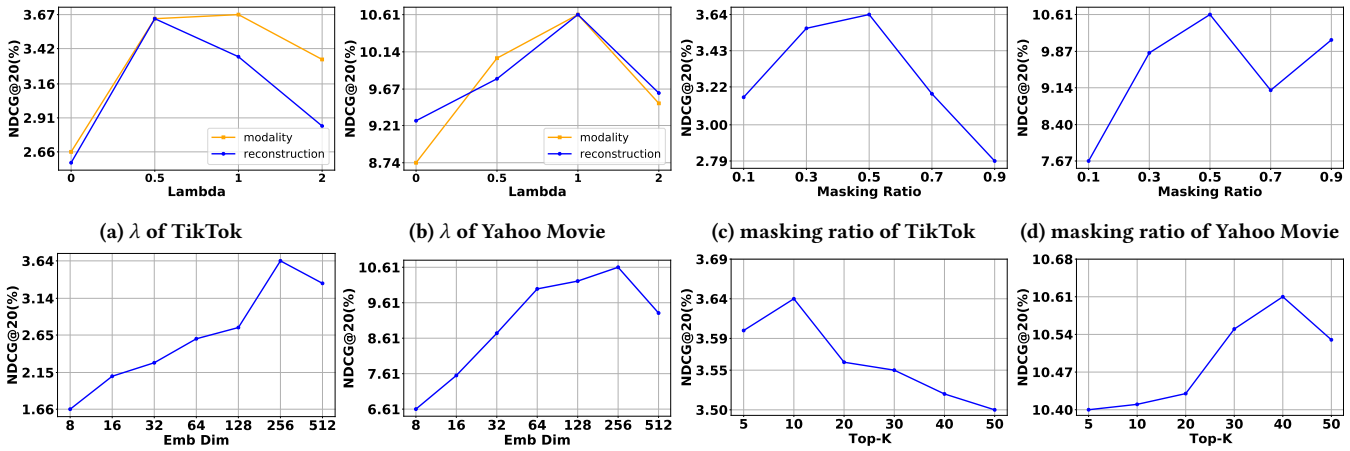
Overall, we conclude that 1) all three modalities play their own roles under our video/movie recommendation settings, 2) particular modalities may play a stronger role depending on the data, *e.g.*, with more detailed text descriptions, text features may play a stronger role in the case of Yahoo Movies, and 3) in general, video features are the most essential with rich content signals.

4.4 Effect of Hyperparameters (RQ3)

We investigate the impact of 5 key hyperparameters of CGRC: the embedding dimensionality d, the masking ratio ρ, the loss weights $\lambda_{\{E,M\}}$, and the number of edges K to be connected. We demonstrate the results on two datasets, TikTok and Yahoo Movie, but a similar trend is also observed in other datasets.

The results are depicted in Fig. 5. The performance with different loss balancing factors, λ_{M} and λ_{E}, is illustrated in Fig. 5(a,b), respectively, where the default value is set to 1.0 for both λ_{E} and λ_{M}. We observe that the best performance is achieved with $\lambda_{\mathrm{M}} = 1.0$ and $\lambda_{\mathrm{E}} = 0.5$ on the TikTok dataset, while with $\lambda_{\mathrm{M}} = 1.0$ and $\lambda_{\mathrm{E}} = 1.0$ on the Yahoo Movie dataset. Overall, we conclude that each loss plays its role with equal importance.

The masking ratio ρ determines the number of cold items $\mathcal{I}_{\mathrm{cold}}$ to be masked during training. From Fig. 5(c,d), we observe that the best performance is achieved at the masking ratio of 0.5 on both datasets. We see that masking too many items hurts the performance since the available information to learn about the dataset gets too sparse. On the other hand, masking too few items also degrades the performance, taking little advantage of our proposed method. In Fig. 5(e,f), we observe that the performance consistently



(a) λ of TikTok (b) λ of Yahoo Movie (c) masking ratio of TikTok (d) masking ratio of Yahoo Movie

(e) d of TikTok (f) d of Yahoo Movie (g) Number of edges K of TikTok (h) Number of edges K of Y-Movie

Figure 5: Effect of Hyperparameters

Table 6: Case Studies of CGRC

User ID	Watch history	Recommended items
3315	E.T. The Extra-Terrestrial Back to the Future Jurassic Park Jaws Armageddon The Lost World: Jurassic Park Gladiator	***Saving Private Ryan** Spartacus The Man Who Knew Too Much The Maltese Falcon Kidnapped
784	Rocky Casino Bringing Out the Dead The Last Temptation of Christ	Crossfire ***Midnight Run** ***The Godfather Part II** Last Man Standing Midnight Cowboy

peaks at $d = 256$ on both datasets. When $d > 256$, the performance slightly drops, indicating that the model starts to overfit.

Finally, we investigate the impact of the value of K, the number of edges to reconstruct for each cold item. As depicted in Fig. 5(g,h), connecting $\mathcal{I}_{\text{cold}}$ to the top $K = 10$ users yields the highest performance on the TikTok dataset, aligning with the dataset's characteristic of approximately 10 average interactions per item. On the Yahoo Movie dataset, the best result is achieved with $K = 40$, which is reasonable, considering that the average degree of an item for the Yahoo Movie dataset is about 25. Note that an excessive number of edge connections degrades the performance, indicating that too many edges introduce irrelevant information. This emphasizes the importance of a proper value of K to ensure meaningful and relevant connections for recommendation.

4.5 Qualitative Analysis (RQ4)

We demonstrate the results of our proposed model, CGRC, through qualitative case studies. Table 6 presents examples of two distinct users with their watch histories and recommended items from our model. The movies in bold and marked with an asterisk indicate the ground truth, *i.e.*, the movies that the user actually watched.

The first case is the user 3315, who has shown a preference for science fiction and adventure movies. Notably, this user has watched several Steven Spielberg films, including "E.T.", "Jurassic Park", "Jaws", and "The Lost World: Jurassic Park". In response, CGRC recommends movies such as "Spartacus" and "Saving Private Ryan". "Spartacus" is a reasonable recommendation since it shares similar contents with films like "Armageddon" and "Gladiator". On the other hand, "Saving Private Ryan", differs significantly in content from the user's viewing history, although it is a Spielberg film. This example shows that CGRC is capable of retrieving relevant movies that are not directly similar content-wise but aligned with the user's detailed preference considering CF signals.

Another example is user 784, who is interested in movies "Casino" and "Bringing Out the Dead", featuring actor Robert De Niro. Our model, CGRC, successfully captures this interest, recommending films such as "Midnight Run", "The Godfather Part 2", and "Last Man Standing". Specifically, "The Godfather Part 2", featuring De Niro, directly aligns with the user's interests, as does "Last Man Standing" in terms of content. Interestingly, although "Midnight Run" differs from the user's usual choices, it includes De Niro, leading to its recommendation by CGRC. This demonstrates that the model's recommendations go beyond basic content matching, effectively grasping user preferences.

5 CONCLUSION

In this work, we present a novel model called Content-based Graph Reconstruction for Cold-start item recommendation (CGRC). By leveraging interaction-based high-order connectivity, CGRC effectively tackles the challenging cold-start problem. Notably, our approach incorporates a sophisticated graph masked autoencoding structure, enhanced with multimodal representations to fully exploit rich content information in a cold-start scenario. Thus, our approach not only distills informative signals but also facilitates the self-supervised reconstruction of user-item edges. The superiority of our model is demonstrated through extensive experiments conducted on real datasets. As a future work, it would be worthwhile to explore the effectiveness of CGRC in the warm-start setting.

Acknowledgements. This work was supported by the New Faculty Startup Fund from Seoul Nat'l Univ., by Youlchon Foundation (Nongshim Corp.), and by NRF grants (No. 2021H1D3A2A03038607/50%, RS-2024-00336576/10%, RS- 2023-00222663/5%) and IITP grants (No. RS-2024-00353131/25%, 2022-0-00264/10%) by the government of Korea.

REFERENCES

[1] Sami Abu-El-Haija, Joonseok Lee, Max Harper, and Joseph Konstan. 2018. MovieLens 20M YouTube trailers dataset. *MovieLens* (2018).

[2] Haoyue Bai, Min Hou, Le Wu, Yonghui Yang, Kun Zhang, Richang Hong, and Meng Wang. 2023. GoRec: A Generative Cold-Start Recommendation Framework. In *Proc. of the ACM International Conference on Multimedia (MM)*.

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2021. BEiT: BERT Pre-Training of Image Transformers. In *Proc. of the International Conference on Learning Representations (ICLR)*.

[4] Oren Barkan, Noam Koenigstein, Eylon Yogev, and Ori Katz. 2019. CB2CF: a neural multiview content-to-collaborative filtering model for completely cold item recommendations. In *Proc. of the ACM Conference on Recommender Systems (RecSys)*.

[5] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2018. Graph convolutional matrix completion. *KDD Deep Learning Day* (2018).

[6] Hao Chen, Zengde Deng, Yue Xu, and Zhoujun Li. 2021. Non-recursive graph convolutional networks. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[7] Hao Chen, Zhong Huang, Yue Xu, Zengde Deng, Feiran Huang, Peng He, and Zhoujun Li. 2022. Neighbor enhanced graph convolutional networks for node classification and recommendation. *Knowledge-Based Systems* 246 (2022), 108594.

[8] Hao Chen, Zefan Wang, Feiran Huang, Xiao Huang, Yue Xu, Yishi Lin, Peng He, and Zhoujun Li. 2022. Generative adversarial framework for cold-start item recommendation. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[9] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[10] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proc. of the AAAI Conference on Artificial Intelligence*.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proc. of the International Conference on Machine Learning (ICML)*.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

[13] Manqing Dong, Feng Yuan, Lina Yao, Xiwei Xu, and Liming Zhu. 2020. MAMO: Memory-augmented meta-optimization for cold-start recommendation. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of the International Conference on Learning Representations (ICLR)*.

[15] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[16] Yuan Gong, Yu-An Chung, and James Glass. 2021. AST: Audio spectrogram transformer. In *Proc. Interspeech*.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*.

[18] F Maxwell Harper and Joseph A Konstan. 2015. The Movielens datasets: History and context. *Acm transactions on interactive intelligent systems (TIIS)* 5, 4 (2015), 1–19.

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[21] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[22] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proc. of the ACM International Conference on World Wide Web (WWW)*.

[23] Zhenyu Hou, Yufei He, Yukuo Cen, Xiao Liu, Yuxiao Dong, Evgeny Kharlamov, and Jie Tang. 2023. GraphMAE2: A Decoding-Enhanced Masked Self-Supervised Graph Learner. In *Proc. of the ACM International Conference on World Wide Web (WWW)*.

[24] Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. GraphMAE: Self-supervised masked graph autoencoders. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[25] Liang Hu, Songlei Jian, Longbing Cao, Zhiping Gu, Qingkui Chen, and Artak Amirbekyan. 2019. HERS: Modeling influential contexts with heterogeneous relations for sparse and cold-start recommendation. In *Proc. of the AAAI Conference on Artificial Intelligence*.

[26] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proc. of the IEEE International Conference on Data Mining (ICDM)*.

[27] Feiran Huang, Zefan Wang, Xiao Huang, Yufeng Qian, Zhetao Li, and Hao Chen. 2023. Aligning distillation for cold-start item recommendation. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[28] Jooeun Kim, Jinri Kim, Kwangeun Yeo, Eungi Kim, Kyoung-Woon On, Jonghwan Mun, and Joonseok Lee. 2024. General Item Representation Learning for Cold-start Content Recommendations. *arXiv:2404.13808* (2024).

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

[30] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proc. of the International Conference on Learning Representations (ICLR)*.

[31] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning* (2016).

[32] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[33] Alex M Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. *Advances in Neural Information Processing Systems (NIPS)* 29 (2016).

[34] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. 2020. Contrastive representation learning: A framework and review. *IEEE Access* 8 (2020), 193907–193934.

[35] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-learned user preference estimator for cold-start recommendation. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[36] Hyodong Lee, Joonseok Lee, Joe Yue-Hei Ng, and Paul Natsev. 2020. Large scale video representation learning via relational graph clustering. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[37] Joonseok Lee and Sami Abu-El-Haija. 2017. Large-scale content-only video recommendation. In *Proc. of the IEEE/CVF Conference on International Conference on Computer Vision (ICCV)*.

[38] Joonseok Lee, Sami Abu-El-Haija, Balakrishnan Varadarajan, and Apostol Natsev. 2018. Collaborative deep metric learning for video understanding. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[39] Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer. 2013. Local low-rank matrix approximation. In *Proc. of the International Conference on Machine Learning (ICML)*.

[40] Joonseok Lee, Seungyeon Kim, Guy Lebanon, Yoram Singer, and Samy Bengio. 2016. LLORMA: Local low-rank matrix approximation. *Journal of Machine Learning Research (JMLR)* 17, 15 (2016), 1–24.

[41] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proc. of the ACM International Conference on World Wide Web (WWW)*.

[42] Yuanfu Lu, Yuan Fang, and Chuan Shi. 2020. Meta-learning on heterogeneous information networks for cold-start recommendation. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[43] Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*.

[44] Tieyun Qian, Yile Liang, Qing Li, and Hui Xiong. 2020. Attribute graph neural networks for strict cold start recommendation. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2020), 3597–3610.

[45] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. (2018).

[46] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

[47] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*.

[48] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. DropEdge: Towards deep graph convolutional networks on node classification. In *Proc. of the International Conference on Learning Representations (ICLR)*.

[49] Paolo Rosso, Dingqi Yang, and Philippe Cudré-Mauroux. 2020. Beyond triplets: hyper-relational knowledge graph embedding for link prediction. In *Proc. of the ACM International Conference on World Wide Web (WWW)*.

[50] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. 2020. RecVAE: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proc. of the International Conference on Web Search and Data Mining (WSDM)*.

[51] Changfeng Sun, Han Liu, Meng Liu, Zhaochun Ren, Tian Gan, and Liqiang Nie. 2020. LARA: Attribute-to-feature adversarial learning for new-item recommendation. In *Proc. of the International Conference on Web Search and Data Mining (WSDM)*.

[52] Qiaoyu Tan, Ninghao Liu, Xiao Huang, Soo-Hyun Choi, Li Li, Rui Chen, and Xia Hu. 2023. S2GAE: Self-Supervised Graph Autoencoders are Generalizable Learners with Graph Masking. In *Proc. of the International Conference on Web Search and Data Mining (WSDM)*.

[53] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proc. of the International Conference on Learning Representations (ICLR)*.

[54] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing cold start in recommender systems. *Advances in Neural Information Processing Systems (NIPS)* 30 (2017).

[55] Chun Wang, Shirui Pan, Guodong Long, Xingquan Zhu, and Jing Jiang. 2017. MGAE: Marginalized graph autoencoder for graph clustering. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*.

[56] Wenjie Wang, Xinyu Lin, Liuhui Wang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. 2023. Equivariant Learning for Out-of-Distribution Cold-start Recommendation. In *Proc. of the ACM International Conference on Multimedia (MM)*.

[57] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[58] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[59] Chunyu Wei, Jian Liang, Di Liu, Zehui Dai, Mang Li, and Fei Wang. 2023. Meta Graph Learning for Long-tail Recommendation. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[60] Wei Wei, Chao Huang, Lianghao Xia, and Chuxu Zhang. 2023. Multi-Modal Self-Supervised Learning for Recommendation. In *Proc. of the ACM International Conference on World Wide Web (WWW)*.

[61] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive learning for cold-start recommendation. In *Proc. of the ACM International Conference on Multimedia (MM)*.

[62] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. 2022. SimMIM: A simple framework for masked image modeling. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[63] Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. 2018. HOP-rec: high-order proximity for implicit recommendation. In *Proc. of the ACM Conference on Recommender Systems (RecSys)*.

[64] Yaowen Ye, Lianghao Xia, and Chao Huang. 2023. Graph masked autoencoder for sequential recommendation. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[65] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

[66] Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. STAR-GCN: Stacked and Reconstructed Graph Convolutional Networks for Recommender Systems. In *IJCAI International Joint Conference on Artificial Intelligence*.

[67] Xu Zhao, Yi Ren, Ying Du, Shenzheng Zhang, and Nian Wang. 2022. Improving item cold-start recommendation via model-agnostic conditional variational autoencoder. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[68] Zhihui Zhou, Lilin Zhang, and Ning Yang. 2023. Contrastive Collaborative Filtering for Cold-Start Item Recommendation. In *Proc. of the ACM International Conference on World Wide Web (WWW)*.

[69] Yongchun Zhu, Ruobing Xie, Fuzhen Zhuang, Kaikai Ge, Ying Sun, Xu Zhang, Leyu Lin, and Juan Cao. 2021. Learning to warm up cold item embeddings for cold-start recommendation with meta scaling and shifting networks. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[70] Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. 2020. Recommendation for new users and new items via randomized training and mixture-of-experts transformation. In *Proc. of the International ACM SIGIR Conference on Research and Development in Information Retrieval*.

[71] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. 2021. Neural bellman-ford networks: A general graph neural network framework for link prediction. In *Advances in Neural Information Processing Systems (NeurIPS)*.