Automated Self-Supervised Learning for Recommendation

Lianghao Xia
The University of Hong Kong
aka_xia@foxmail.com

Chao Huang*
The University of Hong Kong
chaohuang75@gmail.com

Chunzhen Huang
Tencent
chunzhuang@tencent.com

Kangyi Lin
Tencent
plancklin@tencent.com

Tao Yu
The University of Hong Kong
tyu@cs.hku.hk

Ben Kao
The University of Hong Kong
kao@cs.hku.hk

ABSTRACT

Graph neural networks (GNNs) have emerged as the state-of-the-art paradigm for collaborative filtering (CF). To improve the representation quality over limited labeled data, contrastive learning has attracted attention in recommendation and benefited graph-based CF model recently. However, the success of most contrastive methods heavily relies on manually generating effective contrastive views for heuristic-based data augmentation. This does not generalize across different datasets and downstream recommendation tasks, which is difficult to be adaptive for data augmentation and robust to noise perturbation. To fill this crucial gap, this work proposes a unified Automated Collaborative Filtering (AutoCF) to automatically perform data augmentation for recommendation. Specifically, we focus on the generative self-supervised learning framework with a learnable augmentation paradigm that benefits the automated distillation of important self-supervised signals. To enhance the representation discrimination ability, our masked graph autoencoder is designed to aggregate global information during the augmentation via reconstructing the masked subgraph structures. Experiments and ablation studies are performed on several public datasets for recommending products, venues, and locations. Results demonstrate the superiority of AutoCF against various baseline methods. We release the model implementation at https://github.com/HKUDS/AutoCF.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Self-Supervised Learning, Graph Neural Networks, Masked Autoencoder, Automated Machine Learning, Collaborative Filtering

*Chao Huang is the corresponding author.

1 INTRODUCTION

Recommender systems, which aim to suggest items to users by learning their personalized interests, have provided essential web services (*e.g.*, E-commerce sites, online reviews, and advertising platforms) to alleviate the information overload problem [40]. The core part of recommenders lies in effective modeling of user preference on various items based on observed historical interactions.

To date, various types of collaborative filtering (CF) techniques have been proposed to project users and items into latent embedding space, such as matrix factorization [22], autoencoder [28], attention mechanism [4]. In the context of graph learning, graph neural networks (GNNs) have emerged as the state-of-the-art frameworks for collaborative filtering (CF) with the modeling of high-order connectivity among users and items, such as GCMC [1], NGCF [35], LightGCN [15] and GCCF [5]. These recommender systems are proposed to perform the recursive message passing over the generated user-item interaction graph. However, the success of such methods largely relies on sufficient high quality labels, and cannot produce accurate user and item representations when the observed labeled data is scarce and noisy [39]. Recently, contrastive self-supervised learning (SSL) has achieved promising results in generating representations with small labeled data with yielding auxiliary self-supervision signals in Computer Vision [13] and Nature Language Processing [27]. Motivated by this, recent recommendation studies propose to address the limitation of heavy label reliance in current GNN-based recommendation models based on various contrastive learning techniques for data augmentation.

According to the ways in which contrastive views are generated for collaborative filtering, recent contrastive learning models can be broadly categorized into different aspects: i) *Structure-Level Augmentation*: Models such as SGL [39] and DCL [25] employ random node and edge dropout to create contrastive views based on graph topology augmentation. ii) *Feature-Level Augmentation*: SLRec [47] generates contrasts between feature vectors that have been corrupted by random noise for augmentation purposes. iii) *Local-Global Augmentation*: This line aims to reconcile local user and item embeddings with global information by performing local-global contrastive learning. To achieve this goal, various information aggregators are used to produce global-level embeddings, including hypergraph-enhanced message fusion in HCCF [42] and EM algorithm-based node clustering in NCL [24].

However, current contrastive learning recommenders heavily rely on manually generated contrastive views for heuristic-based
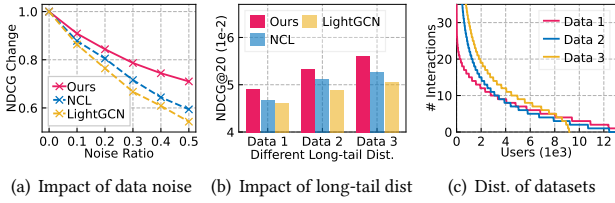
(a) Impact of data noise (b) Impact of long-tail dist (c) Dist. of datasets

Figure 1: The influence of data noise and long-tail distributions on the prediction accuracy of different methods.

data augmentation. The effectiveness of these approaches is guaranteed only when the contrastive views are properly specified for different datasets and tasks. In diverse recommendation scenarios, accurate generation of contrastive representation views is very challenging and manually performing data augmentation may unavoidably involve noisy and irrelevant information for SSL.

From the perspective of graph structure-level and local-global augmentation, the self-discrinimation through random node/edge dropout operations may lead to i) losing important structural information (*e.g.*, limited interactions of inactive users); and ii) keeping the noisy data (*e.g.*, misclick behaviors or popularity bias) for contrasting samples. To have a better understanding of the aforementioned limitations in current methods, Figure 1 presents the performance comparison of different methods against data noise and long-tail distributions. Specifically, we contaminate the training set with different ratios of adversarial user-item interactions as noisy examples. In addition, we randomly sample three datasets from Gowalla with different long-tail distributions (shown in Figure 1 (c)). The superior performance of our AutoCF indicates that existing state-of-the-art SSL recommendation models can hardly learn high-quality user (item) representations when faced with data noise and long-tail issues. The performance of compared current contrastive methods (*e.g.*, NCL [24]) vary greatly with different recommendation data distributions.

Therefore, current contrastive recommender systems are still vulnerable to the quality of the augmented supervision signals, due to their handcrafted strategies of non-adaptive contrastive view generation. With the consideration of limitations in existing work, we believe it is essential to design a unified SSL-based recommendation model, which not only can distill self-supervised signals for effective data augmentation, but also relieve human efforts to manually in generating self-supervision signals or defining augmentation strategies. Towards this end, an interesting question may be raised: *is there a principled way to automatically distill the important self-supervision signals for adaptive augmentation?*

Inspired by the emerging success of generative self-supervised learning in vision learner [12] with the reconstruction objectives, we propose an automated framework for self-supervised augmentation in graph-based CF paradigm via a masked graph auto-encoder architecture, to explore the following questions for model design.

- **Q1**: How to automatically distill self-supervised signals which are more beneficial for the recommendation objective?
- **Q2**: How to enable the learnable graph augmentation by well preserving informative collaborative relationships?
- **Q3**: How to design the graph auto-encoder framework in global information aggregation for the adaptive graph reconstruction.

Contributions. To tackle these challenges, we propose a new Automated Collaborative Flitering (AutoCF) which is capable of distilling the graph structure-adaptive self-supervised signals for advancing the graph-based CF framework. In particular, we design a learnable masking function to automatically identify the important centric nodes for reconstruction-based data augmentation. In the mask learning stage, the node-specific subgraph semantic relatedness will be considered to accurately preserve the graph-based collaborative relationships. Additionally, we propose a new masked graph autoencoder in which the key ingredient is a graph neural encoder that captures the global collaborative relationships for reconstructing the masked user-item subgraph structures.

In summary, our work makes the following contributions:

- Investigate the drawbacks of existing contrastive GNN-based recommendation methods with the non-adaptive self-supervised augmentation and weak robustness against noise perturbation.
- Propose an automated self-supervised learning model AutoCF, in which a learning to mask paradigm is designed to perform data augmentation with structure-adaptive self-supervision. In addition, the automated mask generator is integrated with graph masked autoencoder to enhance user representations with SSL.
- Demonstrate the significant improvements that AutoCF achieves over state-of-the-art GNN-based and SSL-enhanced recommenders, including some recent strong baselines NCL and HCCF.

2 GRAPH CONTRASTIVE LEARNING FOR COLLABORATIVE FILTERING

Recap of Graph-based Collaborative Filtering. In general, the input data for recommender systems involve an user set $\mathcal{U} = \{u\}$, an item set $\mathcal{I} = \{i\}$ and observed user-item relations represented by an interaction matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$. In $\mathbf{A}$, each element $a_{u,i} = 1$ if user u has adopted item i before, $a_{u,i} = 0$ given the non-interaction between user u and item i. To improve the user-item interaction modeling with high-order connectivity, graph-based collaborative filtering models have shown their effectiveness through the recursive message propagation over the constructed interaction graph $\mathcal{G} = \{\mathcal{U}, \mathcal{I}, \mathcal{E}\}$ between u and i for embedding refinement :

$$\mathbf{h}_u^l \leftarrow \mathbf{Aggre}_{i \in \mathcal{N}_u}\Big(\mathbf{Const}(\mathbf{h}_i^{l-1}, \mathbf{h}_u^{l-1}, e_{u,i})\Big)$$
$$\mathbf{h}_i^l \leftarrow \mathbf{Aggre}_{u \in \mathcal{N}_i}\Big(\mathbf{Const}(\mathbf{h}_u^{l-1}, \mathbf{h}_i^{l-1}, e_{i,u})\Big) \quad (1)$$

where in $\mathcal{G}$, users and items are connected through their interaction edges ($e_{u,i}$) based on the interaction matrix $\mathbf{A}$. $\mathbf{h}_u^{l-1}$ and $\mathbf{h}_i^{l-1}$ denote the embedding of user u and item i at the $(l-1)$-th graph neural layer. The core of graph-based CF paradigm contains: i) the message construction function $\mathbf{Const}(\cdot)$ which extracts the feature information from the connected users or items; ii) the information aggregation function $\mathbf{Aggre}(\cdot)$ for gathering the embeddings from the neigboring nodes $i \in \mathcal{N}_u$ or $u \in \mathcal{N}_i$, using different operators, such as mean-pooling, summation, or attentive combination.

Contrastive Augmentation on Interaction Graph. Inspired by the recent success of contrastive learning [20, 31], several recent studies propose to perform graph contrastive learning (GCL) over the user-item interaction data with various data augmentation strategies. In general, contrastive learning-enhanced recommender

systems aim to reach the alignment between the generated contrasting representation views, so as to inject the auxiliary self-supervised objective $\mathcal{L}_{ssl}$ into the recommendation loss $\mathcal{L}_{rec}$ with the joint model optimization for estimating the $u - i$ interaction likelihood:

$$\hat{y}_{u,i} = f(\mathcal{G}; \Theta)[u, i]; \quad \min_{\Theta} \mathcal{L}_{rec}(f, \mathcal{G}) + \mathcal{L}_{ssl}(f, \mathcal{G}; \varphi) \quad (2)$$

However, the success of most state-of-the-art GCL-based recommendation solutions largely relies on the careful design of contrastive views and handcrafted augmentation strategies φ. To address this limitation, this work aims to design graph augmentation scheme that enables the automated self-supervision signal generation for adaptive data augmentation in recommendation.

3 METHODOLOGY

This section elaborates the technical details of adaptive masked graph autoencoder–AutoCF–to enhance user representation with the automated distillation of self-supervision signals based on graph neural networks. We present the architecture of AutoCF in Figure 2.

3.1 Automated Graph Augmentation

3.1.1 **Learning to Mask Paradigm.** To automatically distill the reconstructed self-supervised signals over the graph-structured interaction data $\mathcal{G}$, we propose to reconsutrct the masked user-item interaction edges adaptively for benefiting the modeling of high-order graph collobrative relationships. The core idea of our learnable mask scheme is to first identify the centric nodes in $\mathcal{G}$ and then mask the informative interaction edges based on their subgraph structual information. In general, the learnable graph-based interaction mask function $\varphi(\cdot)$ in AutoCF can be formalized:

$$\varphi(\mathcal{G}, \mathcal{V}; k) = \left\{ \mathcal{U}, \mathcal{I}, \mathcal{V}, \mathcal{E} \setminus \{(v_1, v_2) | v_1, v_2 \in \mathcal{N}_v^k, v \in \mathcal{V}\} \right\} \quad (3)$$

where $\mathcal{V}$ is the set of centric nodes in $\varphi(\cdot)$. $\mathcal{N}_v^k$ denotes the set of neighborhood nodes of v within k hops. Here, k is a hyperparameter to control the injection of high-order connectivity during the reconstruction-based self-augmentation. (v_1, v_2) denotes the existing edge masked through the set subtraction operation "\", in the sampled subgraph structure given the centric node $v \in \mathcal{V}$.

3.1.2 **Infomax-based Semantic Relatedness.** Towards this end, motivated by the mutual information encoding over graphs [29, 32], we leverage the mutual information to measure the semantic relatedness between the node-level embedding and subgraph-level representation, based on the high-order graph collaborative relations among users and items. Formally, the subgraph semantic relatedness score s_v of node v can be derived as follows:

$$s_v = \psi(v; k) = \text{sigm}(\mathbf{h}_v^\top \sum_{v' \in \mathcal{N}_v^k} \mathbf{h}_{v'} / (|\mathcal{N}_v^k| \cdot \|\mathbf{h}_v\| \cdot \|\mathbf{h}_{v'}\|)) \quad (4)$$

where $\mathbf{h}_v, \mathbf{h}_{v'} \in \mathbb{R}^d$ denote randomly-initialized user/item embeddings. sigm$(\cdot)$ dentoes the sigmoid activation. The subgraph-level representation is generated by aggregating the embeddings of all nodes $(v' \in \mathcal{N}_v^k)$ contained in this k-order subgraph except the centric node v, using readout functions, such as mean-pooling or summation [32]. The larger semantic relatedness score indicates not only the higher structural consistency between the target user and his/her graph dependent nodes (users and items), but also the lower percentage of topological information noise in the sampled

subgraph. For example, an outlier user with many item misclicks will lead to lower structural consistency with others with respect to their collaborative relationships of interaction data.

3.1.3 **Learning to Mask with Gumbel Distribution.** Given the designed learnable masking function $\varphi(\mathcal{G}, \mathcal{V}; k)$, our AutoCF is capable of automatically generating the self-supervised reconstruction signals of user-item interactions, which offers adaptive data augmentation. To improve the robutness of our learning to mask paradigm $\varphi(\mathcal{G}, \mathcal{V}; k)$, we inject the Gumbel-distributed noises [19] into the deriviation of node-specific mask probability $\psi(v; k)$:

$$\psi'(v; k) = \log \psi(v; k) - \log(-\log(\mu)); \quad \mu \sim \text{Uniform}(0, 1) \quad (5)$$

Based on the estimated mask probabilities $\psi'(v; k)$ of all nodes in $v \in \mathcal{V}$, we generate a set of S centric nodes by selecting the top-ranked user and item nodes in terms of the learned mask probabilities. To supercharge our AutoCF with learnable data augmentation, we further inject the SSL signals via the subgraph mutual information maximization using the following infomax-based opimized objective $\mathcal{L}_{\text{InfoM}} = -\sum_{v \in \mathcal{U} \cup \mathcal{I}} \psi(v; k)$.

3.2 Masked Graph Autoencoder

The goal of AutoCF is to augment the graph-based collaborative filtering with a reconstruction learning task over the masked user-item interaction edges in graph $\mathcal{G}$. After applying our learning to mask paradigm, we feed the augmented graph with the masked edges into our developed graph autoencoder framework. In particular, AutoCF proposes to leverage graph convolutional network, which is widely used in previous recommender systems [15], as the encoder to incorporate graph structural information into user and item node embeddings. To alleviate the over-smoothing issue in GNNs and supercharge AutoCF with the global information aggregation, we adopt graph self-attention as the decoder to bridge the encoder and the auxiliary self-supervised reconstruction task.

3.2.1 **Graph Convolution-based Structure Encoder.** Recent studies (*e.g.*, SGL [39] and LightGCN [15]) have demontrated the effectiveness of lightweight graph convolution-based message passing for encoding the structual information. Motivated by this, given the corrupted interaction graph $\mathcal{G}' = \varphi(\mathcal{G}, \mathcal{V}; k)$ output from our mask function $\varphi(\mathcal{G}, \mathcal{V}; k)$, our encoder in AutoCF is built over the following embedding propagation schema:

$$\mathbf{h}_v^{l+1} = \alpha_{v,v} \cdot \mathbf{h}_v^l + \sum_{v' \in \mathcal{N}_v'} \alpha_{v,v'} \cdot \mathbf{h}_{v'}^l; \quad \alpha_{v,v'} = 1/\sqrt{|\mathcal{N}_v'||\mathcal{N}_{v'}'|} \quad (6)$$

where $\mathbf{h}_v^l, \mathbf{h}_v^{l+1} \in \mathbb{R}^d$ denotes the node embeddings of node v in the l-th and the $(l+1)$-th graph neural layers, respectively. The scalar $\alpha_{v,v'}$ serves as the normalization weight for node pair (v, v'), which is calculated based on the node degree $|\mathcal{N}_v'|, |\mathcal{N}_{v'}'|$. $\mathcal{N}_v'$ denotes the neighborhood set of node v in the augmented graph $\mathcal{G}'$. During our embedding passing process, AutoCF applies the residual connections to enable the self-propagation in the last graph layer, in order to alleviate the gradient vanishing issue [14].

3.2.2 **Graph Self-Attention Decoder.** Although the graph convolutional encoder allows us to capture the user-item interaction graph structure, the over-smoothing problem will become a natural effect with increasing the graph propagation layers [3, 6]. To alleviate this limitation, we design the graph self-attention module
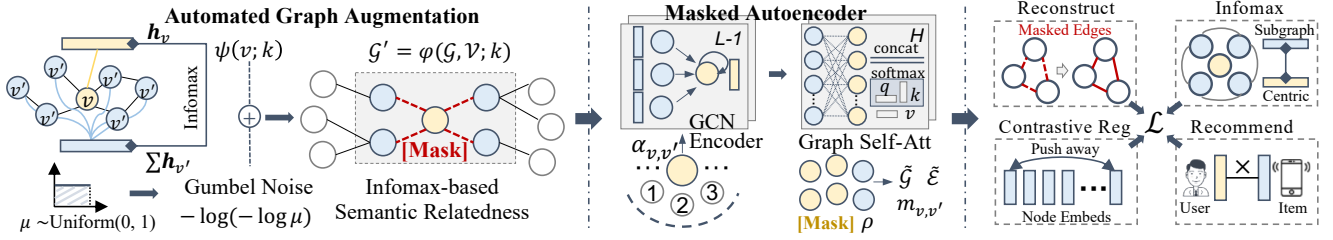
Figure 2: The framework of AutoCF is composed of: i) the adaptive graph structure augmentation based on infomax relatedness; ii) the masked graph autoencoder based on the graph self-attention architecture; iii) the self-augmented training paradigm including graph structure reconstruction, node-subgraph mutual information maximization, and contrastive learning.

as the decoder in AutoCF for the self-supervised structure reconstruction, with the global self-attention for long-range information aggregation, rather than the localized convolutional fusion.

While the attention-based information aggregation addresses the limitation of localized GCN with limited receptive field, the high computational complexity limits its feasibility to perform message passing over the large number of graph user and item nodes. Inspired by the efficient transformer design in encoding long sequences [21] and high-resolution images [50], we propose to conduct the pairwise relation learning over a subset of nodes with the emphasis on the masked subgraph structure. By doing so, we can not only improve the efficiency of our graph self-attention decoder, but also further capture the high-order structural information of centric nodes $\mathcal{V}$ with high subgraph semantic relatedness. Specifically, we firstly define a vertex set $\bar{\mathcal{V}}$ to include the vertices from all the masked subgraphs. Given $\bar{\mathcal{V}}$, a subset $\tilde{\mathcal{V}}$ of nodes will be added from the remaining nodes $((\mathcal{U} \cup \mathcal{I}) \setminus \bar{\mathcal{V}})$. Then, node pairs $\bar{\mathcal{E}}$ are then selected from the union node set, and combined with the edges in $\mathcal{G}'$ using the following formulas:

$$\tilde{\mathcal{G}} = \{\mathcal{U}, \mathcal{V}, \tilde{\mathcal{E}} = \bar{\mathcal{E}} \cup \mathcal{E}'\}; \quad \bar{\mathcal{E}} = \{(v_1, v_2) | v_1, v_2 \in \bar{\mathcal{V}} \cup \tilde{\mathcal{V}}\}$$
$$\text{s.t. } |\bar{\mathcal{E}}| = |\mathcal{E}'|, \ |\bar{\mathcal{V}} \cup \tilde{\mathcal{V}}| = \rho \cdot (|\mathcal{U}| + |\mathcal{I}|) \tag{7}$$

where $\mathcal{E}'$ denotes the edge set of the augmented graph $\mathcal{G}' = \varphi(\mathcal{G}, \mathcal{V}; k)$. ρ is a hyperparameter to control the ratio of the node set. Given the constructed node pairs for global self-attention aggregation, the graph attention-based message passing is presented:

$$\mathbf{h}_v^{l+1} = \sum_{v'} \overset{H}{\underset{h=1}{\|}} m_{v,v'} \beta_{v,v'}^h \mathbf{W}_V^h \mathbf{h}_{v'}^l; \quad m_{v,v'} = \begin{cases} 1 & \text{if } (v, v') \in \tilde{\mathcal{E}} \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_{v,v'}^h = \frac{\exp \bar{\beta}_{v,v'}^h}{\sum_{v'} \exp \bar{\beta}_{v,v'}^h}; \quad \bar{\beta}_{v,v'}^h = \frac{(\mathbf{W}_Q^h \cdot \mathbf{h}_v^l)^\top \cdot (\mathbf{W}_K^h \cdot \mathbf{h}_{v'}^l)}{\sqrt{d/H}} \tag{8}$$

where H denotes the number of attention heads (indexed by h). $m_{v,v'}$ is the binary indicator to decide whether to calculate the attentive relations between node v and v'. $\beta_{v,v'}^h$ denotes the attention weight for node pair (v, v') *w.r.t* the h-th head representation space. $\mathbf{W}_Q^h, \mathbf{W}_K^h, \mathbf{W}_V^h \in \mathbb{R}^{d/H \times d}$ denotes the query, the key, and the value embedding projection for the h-th head, respectively.

Different from conventional graph auto-encoders [11, 30], our AutoCF aims to recover the masked interaction graph structure by learning to discover the masked user-item edges. With the encoded layer-specific user/item representations $\mathbf{h}_u^l$ and $\mathbf{h}_i^l$, the overall embeddings are generated through layer-wise aggregation. Formally,

the reconstruction phase over the masked graph structures is:

$$\mathcal{L}_{\text{recon}} = -\sum_{(v,v') \in \mathcal{E} \setminus \mathcal{E}'} \hat{\mathbf{h}}_v^\top \cdot \hat{\mathbf{h}}_{v'}; \quad \hat{\mathbf{h}}_v = \sum_{l=0}^{L} \mathbf{h}_v^l \tag{9}$$

3.3 Model Training

In our model training stage, we further introduce a contrastive training strategy to enhance the representation discrimination ability with uniformly-distributed user/item embeddings, so as to better preserve the unique preference of users in latent space. Inspired by the InfoNCE-based augmentation in [39], in the learning process of AutoCF, we propose to generate more uniform user and item embeddings with the regularization for user-item, user-user, item-item pairs, for improving the embedding discrimination ability and further alleviating the over-smoothing effect. The loss function $\mathcal{L}_{\text{ssl}}$ with the augmented self-supervised learning objectives is:

$$\mathcal{L}_{\text{ssl}} = \sum_{u \in \mathcal{U}} \log \sum_{i \in \mathcal{I}} \exp \hat{\mathbf{h}}_u^\top \hat{\mathbf{h}}_i + \sum_{u \in \mathcal{U}} \log \sum_{u' \in \mathcal{U}} \exp \hat{\mathbf{h}}_u^\top \hat{\mathbf{h}}_{u'}$$
$$+ \sum_{i \in \mathcal{I}} \log \sum_{i' \in \mathcal{I}} \exp \hat{\mathbf{h}}_i^\top \hat{\mathbf{h}}_{i'} + \mathcal{L}_{\text{InfoM}} + \mathcal{L}_{\text{recon}} \tag{10}$$

To perform the learning process with the main recommendation task and augmented SSL optimized objectives, we define our joint loss function for model optimization as follows:

$$\mathcal{L} = -\sum_{(u,i) \in \mathcal{E}} \hat{\mathbf{h}}_u^\top \cdot \hat{\mathbf{h}}_i + \lambda_1 \cdot \mathcal{L}_{\text{ssl}} + \lambda_2 \cdot \|\Theta\|_F^2 \tag{11}$$

λ_1 and λ_2 control the regularization strength of the distilled self-supervised signals and weight-decay constrain term.

3.4 Theoretical and Complexity Analysis

3.4.1 Noise Filtering via Infomax. In this section, we give a theoretical discussion on the benefits of our infomax-based semantic relatedness in alleviating noisy signals for SSL-based augmentation. By avoiding involving noise in our adaptive graph augmentation using infomax, the masked autoencoding task better benefits the downstream recommendation task with noise-less gradients. Detailed derivations are presented in Section A.6. In brief, we first analyze the source of noises by giving the gradient of $\mathcal{L}_{\text{recon}}$ over the embedding $\mathbf{h}_{v_1}$ of a node v_1 in the masked subgraph as:

$$\frac{\partial \mathcal{L}_{\text{recon}}}{\partial \mathbf{h}_{v_1}} = -\sum_{(v_1,v_2) \in \mathcal{E} \setminus \mathcal{E}'} \mathbf{h}_{v_2} + \sum_{m_{v_2,v'}=1} \beta'_{v_2,v'} \mathbf{W}'_V \mathbf{h}_{v'} \tag{12}$$

Table 1: Statistics of the experimental datasets.

Dataset	# Users	# Items	# Interactions	Interaction Density
Gowalla	25,557	19,747	294,983	5.85×10^{-4}
Yelp	42,712	26,822	182,357	1.59×10^{-4}
Amazon	76,469	83,761	966,680	1.51×10^{-4}

where node v_2 is adjacent to v_1, v' is an arbitrary node, most-probably from the masked subgraph due to our mask-dependent sampling strategy (Section 3.2.2). By inspecting the gradient in Eq 12, we find that the second term introduces noise if v' is semantically less-relevant to v_1. Specifically, although the weight $\beta'_{v_2,v'}$ can reduce the influence of irrelevant v', it is normalized using softmax with high temperature, which in practical learning process, is prone to assigning non-neligable weights to less-relevant v' when a large portion of nodes in the masked subgraph are less-relevant.

To avoid the above situation, AutoCF learns to mask noise-less subgraphs by referring to the infomax-based relatedness s_v. For a centric node v, s_v not only measures the relatedness between v and its neighboring nodes, but also restricts the lowerbound of semantic relatedness between any pair of nodes in the subgraph, as follows:

$$\cos(v_1, v') > \cos(v, v_1) + \cos(v, v') - 1 \qquad (13)$$

As subgraphs with larger s_v have larger center-neighbor similarity $\cos(v, v_1)$ and $\cos(v, v')$, larger s_v also indicates higher lowerbound for the semantic relatededness between any nodes (v_1, v') in the subgraph. In other words, by masking subgraphs with larger infomax-based relatedness, AutoCF reduces the likelihood of introducing noisy gradient in the self-augmented reconstruction task.

3.4.2 **Model Complexity Analysis.** We analyze the time complexity of our AutoCF from different components. i) In the learning to mask paradigm, the subgraph-level embedding generation process takes $O(k \times |\mathcal{E}| \times d)$ complexity for maintainaning a multi-order node intersection set. Here, $|\mathcal{E}|$ denotes the size of the edge set and d is the latent embedding dimensionality. Additionally, the masked edge detection takes $O(|\mathcal{E}| \times |\mathcal{V}_{k'}|)$ time, where $\mathcal{V}_{k'}$ denotes the set of centric nodes for the k'-th iteration ($\mathcal{V}_0 = \mathcal{V}$, $1 \leq k' \leq k$). ii) In the masked graph autoencoder component, the most time-consuming part is the graph self-attention which takes $O(|\mathcal{E}| \times d^2)$ complexity. iii) In the model training phase, contrastive augmentation takes $O(B \times (|\mathcal{U}| + |\mathcal{I}|) \times d)$ with B denoting the batch size. In summary, our AutoCF can achieve comparable overall complexity compared with state-of-the-art graph contrastive recommendation models, such as HCCF [42] and SGL [39].

4 EVALUATION

In this section, experiments are performed to validate AutoCF's advantage and answer the following key research questions:

- **RQ1**: How effective is our proposed AutoCF method perform against various recommendation techniques?
- **RQ2**: How effective are the main components of our AutoCF?
- **RQ3**: How is the sensitivity of the hyperparameters in AutoCF?
- **RQ4**: How efficient is our AutoCF compared with baselines?
- **RQ5**: How is the interpretation ability of the proposed AutoCF?

4.1 Experimental Settings

4.1.1 **Datasets and Evaluation Protocols.** We use three widely-used datasets, including Gowalla, Yelp and Amazon to evaluate

the performance of different recommendation tasks for locations, venues and online products. **Gowalla**. This dataset is collected from a location-based service to record the check-in behaviors between users and different locations from Jan, 2010 to Jun, 2010. **Yelp**. This dataset contains user-venue rating interaction from Jan, 2018 to Jun, 2018 on Yelp platform. **Amazon**. This is another benchmark dataset in evaluating recommender systems. We use the version of user implicit feedback over the book category of items.

In our experiments, 70% of the observed interactions are randomly sampled to generate the training set. In the remaining 30% data, 5% and 25% percentage of interactions are used for validation and testing. Following the same settings in [39, 42], we adopt the all-rank evaluation protocol [23] to measure the item recommendation accuracy for each user. Two representative evaluation metrics *Recall@N* and *NDCG@N* are used to evaluate all methods.

4.1.2 **Baseline Methods.** For comprehensive performance comparision, our baseline set contains 14 recommendation methods.
1) Conventional Collaborative Filtering Methods
- **BiasMF** [22]: This is a widely-adopted matrix factorization baseline which projects users and items into latent vectors.
- **NCF** [16]: It is a representative neural CF method which replaces the inner-product with the non-linear feature projection.

2) Autoencoder-based Recommender System
- **AutoR** [28]: It employs the autoencoder to generate the embeddings of users and items with the reconstruction loss.

3) GNN-based Collaborative Filtering
- **PinSage** [48]: It leverages the graph convolution network to model the user-item interaction through graph structures.
- **STGCN** [51]: It design the graph autoencoding for high-order interaction learning with the reconstruction regularization.
- **GCMC** [1]: It is one of the pioneering CF models that enhances the matrix completion with GNN-based message passing.
- **NGCF** [35]: It is one of the state-of-the-art graph CF models that incorporates high-order connectivity for recommendation.
- **GCCF** [5] and **LightGCN** [15]: To simplify the message passing scheme, recent studies propose to omit the non-linear feature transformation and activiation, which achieves better results.

4) Disentangled Representation-enhanced GNN Model
- **DGCF** [36]: This method disentangles latent factors behind user-item interactions under the graph neural network architecture, to capture the fine-grained user-item relationships.

5) SOTA Self-Supervised Recommendation Methods
- **SLRec** [47]: This self-supervised learning method conducts data augmentation with the learning of latent feature correlations.
- **NCL** [24]: It enhances the graph contrastive learning with the augmented structural and semantic-relevant training pairs.
- **SGL** [39]: This method applies random structure augmentation such as node/edge drop and random walk, to generate contrastive views for graph augmentation in collaborative filtering.
- **HCCF** [42]: This method conducts local-global cross-view contrastive learning with hypergraph-based structure learning.

4.1.3 **Hyperparameter Settings.** We implement AutoCF with PyTorch. The Adam optimizer is used for parameter inference, with learning rate of $1e^{-3}$ and batch size of 4096. By default, the

Table 2: Performance comparison on Gowalla, Yelp, and Amazon datasets in terms of *Recall* and *NDCG*.

| Data	Metric	BiasMF	NCF	AutoR	PinSage	STGCN	GCMC	NGCF	GCCF	LightGCN	DGCF	SLRec	NCL	SGL	HCCF	*AutoCF*	p-val.
Gowalla	Recall@20	0.0867	0.1019	0.1477	0.1235	0.1574	0.1863	0.1757	0.2012	0.2230	0.2055	0.2001	0.2283	0.2332	0.2293	**0.2538**	$1.3e^{-10}$
	NDCG@20	0.0579	0.0674	0.0690	0.0809	0.1042	0.1151	0.1135	0.1282	0.1433	0.1312	0.1298	0.1478	0.1509	0.1482	**0.1645**	$4.9e^{-12}$
	Recall@40	0.1269	0.1563	0.2511	0.1882	0.2318	0.2627	0.2586	0.2903	0.3181	0.2929	0.2863	0.3232	0.3251	0.3258	**0.3441**	$9.3e^{-10}$
	NDCG@40	0.0695	0.0833	0.0985	0.0994	0.1252	0.1390	0.1367	0.1532	0.1670	0.1555	0.1540	0.1745	0.1780	0.1751	**0.1898**	$1.0e^{-9}$
Yelp	Recall@20	0.0198	0.0304	0.0491	0.0510	0.0562	0.0584	0.0681	0.0742	0.0761	0.0700	0.0665	0.0806	0.0803	0.0789	**0.0869**	$6.2e^{-7}$
	NDCG@20	0.0094	0.0143	0.0222	0.0245	0.0282	0.0280	0.0336	0.0365	0.0373	0.0347	0.0327	0.0402	0.0398	0.0391	**0.0437**	$1.0e^{-6}$
	Recall@40	0.0307	0.0487	0.0692	0.0743	0.0856	0.0891	0.1019	0.1151	0.1175	0.1072	0.1032	0.1230	0.1226	0.1210	**0.1273**	$1.2e^{-4}$
	NDCG@40	0.0120	0.0187	0.0268	0.0315	0.0355	0.0360	0.0419	0.0466	0.0474	0.0437	0.0418	0.0505	0.0502	0.0492	**0.0533**	$1.1e^{-5}$
Amazon	Recall@20	0.0324	0.0367	0.0525	0.0486	0.0583	0.0837	0.0551	0.0772	0.0868	0.0617	0.0742	0.0955	0.0874	0.0885	**0.1277**	$5.1e^{-13}$
	NDCG@20	0.0211	0.0234	0.0318	0.0317	0.0377	0.0579	0.0353	0.0501	0.0571	0.0372	0.0480	0.0623	0.5690	0.0578	**0.0879**	$8.0e^{-13}$
	Recall@40	0.0578	0.0600	0.0826	0.0773	0.0908	0.1196	0.0876	0.1175	0.1285	0.0912	0.1123	0.1409	0.1312	0.1335	**0.1782**	$7.3e^{-13}$
	NDCG@40	0.0293	0.0306	0.0415	0.0402	0.0478	0.0692	0.0454	0.0625	0.0697	0.0468	0.0598	0.0764	0.0704	0.0716	**0.1048**	$1.5e^{-13}$

(a) Gowalla-Recall (b) Gowalla-NDCG
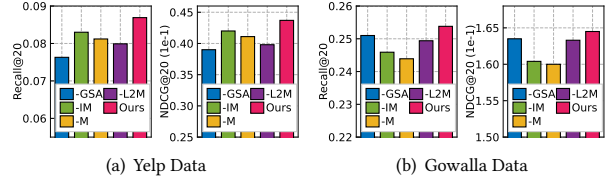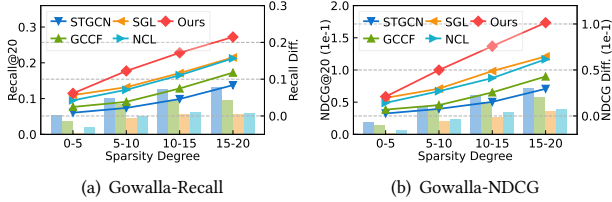
Figure 3: Performance against data sparsity. Left-side y-axis: performance curve of each method. Right-side y-axis: improvement ratio between AutoCF and each baseline.

(a) Yelp Data (b) Gowalla Data

Figure 4: Module ablation study on Yelp and Gowalla.

hidden dimensionality is set as 32, and the number of graph neural iterations is tuned from {1,2,3}. The graph self-attention employs 4 attention heads. We conduct subgraph sampling for every 10, 30, or 60 steps for further improving the model efficiency. In our learning to mask paradigm, the size of centric nodes is selected from {200, 400, ..., 1800, 2000}. The ratio of sampled nodes ρ is set as 0.2. We select the number of graph hops for subgraph masking from {1,2,3} to reflect the high-order semantic relatedness. In the model training phase, λ_1 and λ_2 are respectively chosen from $\{1e^{-k} | -1 \leq k \leq 4\}$ and $\{1e^{-k} | 3 \leq k \leq 8\}$ to control the regularization strength of weight-decay factor and the augmented self-supervised signals.

4.2 Overall Performance Comparison (RQ1)

Table 2 lists the overall performance of all compared methods on three datasets. From the results, we have the observations below:

- **Obs.1**: **Superiority over SOTA SSL approaches**. AutoCF consistently achieves the best performance compared with SOTA recommendation approaches by a large margin (measured by calculated *p*-values), including strong self-supervised approaches in all datasets. By incorporating the adaptive data augmentation into the graph SSL framework through our learning to mask paradigm, AutoCF surpasses existing solutions with the automated self-supervised signal generation. The augmentation strategy in current SSL methods (*e.g.*, SGL, HCCF) with self-discrimination over all nodes, which make them vulnerable to the perturbation, such as popularity bias and interaction noise. That is, with the careful design of our graph augmentation scheme adaptive for the infomax-based collaborative relation learning, self-supervised signals with more beneficial instances will be automately identified for data augmentation.

- **Obs.2**: **Benefits with SSL augmentation**. Comparing various baselines, we notice that SSL methods (*i.e.*, HCCF, SGL, NCL) outperforms pure GNN-based CF models (*e.g.*, LightGCN, NGCF,

GCCF) in most cases. This verifies the benefits of self-supervised data augmentation under the graph-based CF scenario. Such performance gap sheds light on the limitation of GNN-based CF model: i) easily aggregating noisy information for downstream recommendation task; ii) the over-fitting issue over sparse data, which weakens the model representation quality.

- **Obs.3**: **Performance against data sparsity**. We further investigate the recommendation performance of our AutoCF and several representative baselines with respect to different data sparsity degrees. In particular, users are partitioned into four groups in terms of their interaction frequencies, *i.e.*, $[0, 5), [5, 10), [10, 15), [15, 20)$. Bars in this figure indicate the performance improvement between our AutoCF and each compared baseline corresponding to the ratio in the right side y-axis. The performance curves of all methods are shown in Figure 3 in terms of *Recall@20* and *NDCG@20*. The evaluation results demonstrate that our AutoCF owns a consistent superiority over strong self-supervised learning baselines, which again confirms the effectiveness of our adaptive data augmentation in accurately learning user preference with small labeled interaction data. The data augmentation with stochastic operations, such as the random node/edge dropout in SGL and corrupt embedding with random noise perturbation in SLRec, may drop some important information for inactive users, making the data sparsity issue even worse.

4.3 Module Ablation Study (RQ2)

We conduct ablation study to investigate the individual contribution of different sub-modules of the proposed AutoCF to the superior recommendation results. The results are reported in Figure 4.

Effect of graph self-attention. We study the benefit of graph self-attention decoder for the reconstruction-based SSL task with the variant **-GSA**. In this variant, a symmetry encoder-decoder network is adopted to replace graph self-attention with the graph convolutional network. The results manifest that our AutoCF has an obvious improvement over **-GSA**, which demontrates that aggregation with global self-attention over the augmented graph improves
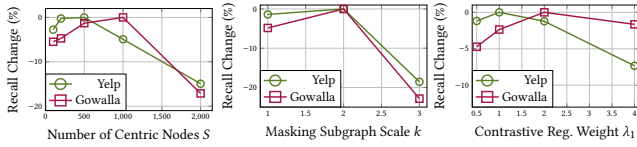
Figure 5: Hyperparameter study for the proposed AutoCF in terms of Recall@20 changes on Yelp and Gowalla datasets.

performance against the over-smoothing issue in GNNs.

Effect of reconstruction SSL signals. We study the influence of our augmented generative self-supervised signals with the masked reconstruction objective. From the results of AutoCF and variant -**M** (without the $\mathcal{L}_{\text{InfoM}}$ and $\mathcal{L}_{\text{recon}}$ regularization), the performance gain reveals that the design generative SSL task is beneficial for enhancing graph-based CF paradigm, by injecting auxiliary self-supervised signals to reflect true underlying patterns.

Effect of infomax-based semantic relatedness. The learnable mask generation function $\varphi(\mathcal{G}, \mathcal{V}; k)$ plays a critical role in the automated augmentation of our AutoCF model by considering the infomax-based subgraph semantic relatedness s_v. Instead, the variant -**IM** removes the infomax-based loss $\mathcal{L}_{\text{InfoM}}$ in the mask generator to discard the guidance of local-global consistency. The performance degradation of -**IM** variant indicates that our AutoCF can provide meaningful gradients to guide the model training in recovering importnat and noise-resistent collaborative relations.

Effect of learning to mask paradigm. The adaptive centric node-based masking strategy $\psi'(v; k)$ is replaced with the randomly edge masking in the variant -**L2M**. While the SSL reconstruction task is still added for data augmentation, the non-adaptive augmention scheme may contain noisy information for suboptimal performance.

4.4 Hyperparameter Investigation (RQ3)

This section investigates the effect of several important hyperparameters on the recommendation performance of AutoCF. The evalution results in terms of *Recall@20* and *NDCG@20* are presented in Figure 5. Due to space limit, the y-axis represents the relative performance degradation ratio compared with the best accuracy. We summarize the results with the following observations:

- **Number of Centric Nodes** S: This hyperparameter determines the size of the identified centric nodes in our learning to mask paradigm. We vary parameter S in the range of $200 \leq S \leq 2000$. As shown in the results, we first observe the performance improvement trend by increasing the size of centric node set. This observation points out the positive effect of incorporating more self-supervised signals into our recommendation task. However, further increasing the number of centric nodes results in more masked interaction edges, which might lead to damaging to the graph structure and distort the user (item) representations.

- **Masking Subgraph Scale** k: In our graph masked autoencoder, k represents the maximum hops included in the masked subgraph. The larger k value indicates higher-order node dependency modeling with respect to user (item) semantic relatedness. The best performance is achieved with 2 hops. By comparing with the adaptive masking within 1-hop subgraph, the incorporation of 2-order may bring benefits to the consideration of high-order
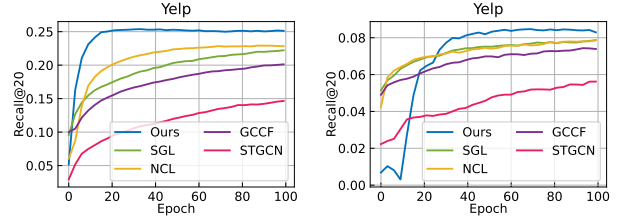

Figure 6: Convergence analysis *w.r.t* epochs while training.

Table 3: Model efficiency study in terms of per-epoch training time (seconds) on Gowalla, Yelp, and Amazon datasets.

Model	DGCF	NCL	HCCF	SGL	Ours
Gowalla	12.03s	5.38s	6.00s	8.07s	7.09s
Yelp	11.47s	3.33s	4.07s	4.88s	4.06s
Amazon	85.54s	25.62s	48.28s	49.87s	59.81s

semantic relatedness in our self-supervised augmentation.

- **Contrastive Regularization Strength** λ_1: In our model training phase, λ_1 determines the strength of contrastive regularization to enhance the discrimination ability of encoded representations. It can be seen that an approriate regularization weight improves the model representation ability against graph over-smoothing issue with better user preference discrimination.

4.5 Model Efficiency Study (RQ4)

4.5.1 **Model Convergence Analysis.** This section investigates the convergence of our AutoCF and the results are depicted in Figure 6. We observe that AutoCF benefits from the incorporation of adaptive self-supervision signals for faster convergence to 20 and 40 training epochs on Gowalla and Yelp, respectively. While self-supervision is performed for data augmentation in state-of-the-art contrastive recommender systems (*i.e.*, SGL, NCL), the weak robustness of those methods against noisy collaborative relations in graphs , leads to their slower convergence compared with our AutoCF model. This observation validates the training efficiency of AutoCF, meanwhile maintaining superior performance. Such model advantage can be ascribed to that our offered SSL task with infomax-based semantic relatedness is helpful for guiding the optimization with better gradients.

4.5.2 **Computational Cost Evaluation.** We further perform the model efficiency study and report the evaluation results in Table 3. While we enable the automated self-supervision signal distillation for data augmentation in recommender systems, we can still achieve competitive efficiency compared with state-of-the-art methods. Compared with our method, SGL performs the contrastive learning by constructing dual-view representation space for self-supervision, which requires more time for extra branch-based augmentation and graph-based embedding propagations.

We further evaluate the scalability of our method in handling large-scale datasets collected from an online retailer site. The evaluation results are presented in Supplementray Section A.5.

4.6 Model Case Study (RQ5)

Semantic Relatedness Interpretation. In Figure 7, we sample three users (u_{3580}, u_{9510}, u_{4628}) with their learned subgraph semantic relatedness scores s_v. We see that u_{4628} is observed to be more
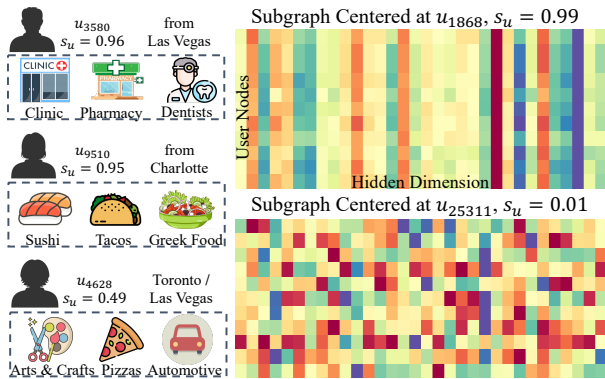
Figure 7: Case study for i) model interpretation in learning semantic relatednes of user interactions; ii) heatmaps of encoded user embeddings from two user-centric subgraphs.

likely with interaction inconsistency compared with u_{3580} and u_{9510}, because some of his/her interacted venues within a short time period are geographically located at distant cities (*e.g.*, Las Vegas and Toronto). In contrast, u_{3580}'s interacted venues (*e.g.*, Clinic, Dentists and Pharmacy) show high semantic relatedness in the same city. Additionally, we sample two subgraph from one identified centeric node (u_{1868}) and other node (u_{25311}). Each row represents the embedding of a user in the subgraph. By comparing the visualized node embeddings of these two subgraphs, we observe the strong homophily of the identified subgraph structure, which is consistent with high semantic relatedness for generative data augmentation.

Visualization of Representation Distributions. We further show the distribution of user representations learned by different methods in 2-d space using tSNE and Gaussian kernel density estimation (KDE) in Figure 8. The embeddings encoded by AutoCF exhibit better uniformity compared with SOTA contrastive approaches, which is indicative of better ability for preserving unique user preference.

5 RELATED WORK

GNN-based Recommender Systems. Due to the strength of GNNs, a line of research for recommendation focuses on enhancing the relational modeling with graph neural architectures in various recommendation scenarios [10, 41]. For example, i) *Social Recommendation*. GNN-based social recommender systems are developed to jointly perform message passing over the user-user and user-item connections, including GraphRec [9], KCGN [18] and ESRF [49]. ii) *Sequential/Session-based Recommendation*. To improve the embedding quality, many recent attempts leverage GNNs to encode repreentations over the item sequences and capture the dynamic user preference, *e.g.*, LESSR [7], GCE-GNN [37], and SURGE [2]. iii) *Knowledge graph-enhanced recommendation*. GNNs have also been adopted for improve the representation over the knowledge graphs to consider item semantic dependencies for recommendation, like KGAT [34], KGNN-LS [33] and KGCL [46]. Towards this research line, our AutoCF is built over GNN and advances the graph-based interaction modeling with automated data augmentation, which is less explored in existing solutions of recommender systems.

Self-Supervised Learning for Recommendation. Recent studies bring the benefits of SSL into the recommender systems to
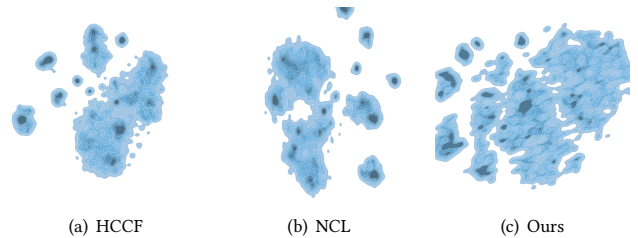

Figure 8: User embedding distribution of different methods with Gaussian kernel density estimation (KDE) on Yelp data.

address the data sparsity and noise challenges [8, 38, 39, 43, 44]. For example, contrastive SSL models have become the state-of-the-art recommendation paradigm by performing the data augmentation with their generated handcrafted or random contrastive views over graph structures, such as SGL [39], HCCF [42] and NCL [24]. In addition, some sequential models have also been improved with contrastive learning strategies to boost the recommendation performance, *e.g.*, ICL [8] and CL4SRec [45]. However, the success of most contrastive learning-based recommender systems relies on manually designing effective contrastive views for reaching embedding agreement, which limits the model generalization ability. To fill this gap, this proposed AutoCF method can offer great potential benefits for automated SSL signal distillation in recommendation.

Graph Autoencoders. Autoencoder has become an effective neural network which is composed of the encoder to map input data into latent dimensions and the decoder to return the reconstruction of original input. It has inspired a wide range of graph learning applications, such as vertex feature reconstruction for node and graph classification [17], embedding reconstruction for graph-based recommendation [51], and structure reconstruction for molecular graph [26]. Despite their effectiveness, most of existing graph autoencoders rely on the feature/structure reconstruction without masking or randomly masking, which can hardly be robust and adaptive to the target downstream graph learning tasks.

6 CONCLUSION

The authors of this work have identified key limitations in existing self-supervised recommendation models, and propose a novel solution called the AutoCF model. By incorporating self-supervised learning signals through infomax-based subgraph semantic relatedness, AutoCF offers a graph augmentation scheme to distill informative self-supervision information in an automated manner. To evaluate the effectiveness of AutoCF, the authors conducted extensive experiments on benchmark datasets and compared its performance to other types of recommender systems. The results demonstrate the effectiveness of our new approach compared with existing recommendation methods. This work highlights the potential benefits of automated self-supervised learning and provides an effective solution for enhancing recommender systems.

REFERENCES

[1] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. In *KDD*.

[2] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *SIGIR*. 378–387.

[3] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *AAAI*. 3438–3445.

[4] 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 *SIGIR*. 335–344.

[5] 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 *AAAI*, Vol. 34. 27–34.

[6] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. 2020. Simple and deep graph convolutional networks. In *ICML*. PMLR, 1725–1735.

[7] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In *KDD*. 1172–1180.

[8] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *WWW*. 2172–2182.

[9] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *WWW*. 417–426.

[10] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, et al. 2021. Graph neural networks for recommender systems: challenges, methods, and directions. *TOIS* (2021).

[11] Arman Hasanzadeh, Ehsan Hajiramezanali, Krishna Narayanan, Nick Duffield, Mingyuan Zhou, and Xiaoning Qian. 2019. Semi-implicit graph variational auto-encoders. *NIPS* 32 (2019).

[12] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *CVPR*. 16000–16009.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[15] Xiangnan He, Kuan Deng, Xiang Wang, et al. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[16] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. 173–182.

[17] Zhenyu Hou, Xiao Liu, Yuxiao Dong, Chunjie Wang, Jie Tang, et al. 2022. GraphMAE: Self-Supervised Masked Graph Autoencoders. In *KDD*.

[18] Chao Huang, Huance Xu, Yong Xu, Peng Dai, Lianghao Xia, Mengyin Lu, Liefeng Bo, Hao Xing, Xiaoping Lai, and Yanfang Ye. 2021. Knowledge-aware coupled graph neural network for social recommendation. In *AAAI*. 4115–4122.

[19] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparametrization with Gumble-Softmax. In *ICLR*. OpenReview. net.

[20] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Neurips* 33 (2020), 18661–18673.

[21] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *ICLR*.

[22] Yehuda Koren, Robert Bell, et al. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[23] Walid Krichene and Steffen Rendle. 2022. On sampled metrics for item recommendation. *Commun. ACM* 65, 7 (2022), 75–83.

[24] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *WWW*. 2320–2329.

[25] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, and Zhang Xiong. 2021. Contrastive learning for recommender system. *arXiv preprint arXiv:2101.01317* (2021).

[26] Changsheng Ma and Xiangliang Zhang. 2021. GF-VAE: A Flow-based Variational Autoencoder for Molecule Generation. In *CIKM*. 1181–1190.

[27] Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural machine translation. *ACL* (2021).

[28] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. 2015. Autorec: Autoencoders meet collaborative filtering. In *WWW*. 111–112.

[29] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2020. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *ICLR*.

[30] Da Tang, Dawen Liang, Tony Jebara, and Nicholas Ruozzi. 2019. Correlated variational auto-encoders. In *ICML*. PMLR, 6135–6144.

[31] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? *Neurips* 33 (2020), 6827–6839.

[32] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax.. In *ICLR*.

[33] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *KDD*. 968–977.

[34] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *KDD*. 950–958.

[35] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*.

[36] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *SIGIR*. 1001–1010.

[37] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *SIGIR*. 169–178.

[38] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive meta learning with behavior multiplicity for recommendation. In *WSDM*. 1120–1128.

[39] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[40] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *TKDE* (2022).

[41] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2020. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys (CSUR)* (2020).

[42] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *SIGIR*. 70–79.

[43] Lianghao Xia, Chao Huang, and Chuxu Zhang. 2022. Self-supervised hypergraph transformer for recommender systems. In *KDD*. 2100–2109.

[44] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. 2021. Self-supervised graph co-training for session-based recommendation. In *CIKM*. 2180–2190.

[45] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *ICDE*. IEEE, 1259–1273.

[46] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *SIGIR*. 1434–1443.

[47] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, et al. 2021. Self-supervised Learning for Large-scale Item Recommendations. In *CIKM*. 4321–4330.

[48] Rex Ying, Ruining He, Kaifeng Chen, et al. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*. 974–983.

[49] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. 2020. Enhance social recommendation with adversarial graph convolutional networks. *TKDE* (2020).

[50] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. 2022. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*. 5728–5739.

[51] Jiani Zhang, Xingjian Shi, Shenglin Zhao, et al. 2019. Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. In *IJCAI*.

A APPENDIX

In our provided appendix, we summarize the learning process of our AutoCF recommender in Section A.1. Then, we present the implementation details of baseline methods in Section A.2. Additionally, we present the model robustness study against data noise and impact study of model depth in Section A.3 and Section A.4, respectively. Additionally, the model scalability investigation is presented in Section A.5. Finally, the derivation details of our AutoCF model with therotical analysis are presented in Section A.6.

A.1 Learning Algorithm Description

We present the learning steps of our AutoCF in Algorithm 1.

Algorithm 1: Learning Process of AutoCF

Input: User-item interaction graph $\mathcal{G}$, number of centric nodes in masking S, masking depth k, learning rate η, training epochs E, number of graph iterations L, regularization weights λ_1, λ_2.

Output: Trained model parameters Θ.

1 Initialize model parameters Θ
2 **for** $e = 1$ *to* E **do**
3   Initialize the maintained average difference of training loss by $\bar{\nabla}\mathcal{L}'_{\text{rec}} = 0$
4   **for** *mini-batch* $\{(u, i)\}$ *drawn from* $\mathcal{E}$ **do**
5   Calculate the infomax-based semantic relatedness s_v for each node v (Eq 4)
6   Sample centric nodes $\mathcal{V}$ with Gumbel noise (Eq 5)
7   Conduct masking by $\mathcal{G}' = \varphi(\mathcal{G}, \mathcal{V}; k)$ (Eq 3)
8   Calculate the infomax-based loss $\mathcal{L}_{\text{InfoM}}$
9   **for** $l = 1$ *to* L **do**
10   Apply GCN encoding on $\mathcal{G}'$ (Eq 6)
11   **end**
12   Sample edges $\tilde{\mathcal{G}}$ for graph self-attention (Eq 7)
13   Conduct graph self-attention decoding (Eq 8)
14   Calculating the reconstruction loss $\mathcal{L}_{\text{recon}}$ (Eq 9)
15   Calculate the self-contrastive SSL loss $\mathcal{L}_{\text{ssl}}$ of the current batch (Eq 10)
16   Calculate the recommendation loss and weight-decay regularization and combine them with the SSL loss to obtain $\mathcal{L}$ (Eq 11)
17   **for** *each parameter* θ *in* Θ **do**
18           $\theta = \theta - \eta \cdot \frac{\partial \mathcal{L}}{\partial \theta}$;
19   **end**
20   **end**
21 **end**
22 **return** all parameters Θ

A.2 Baseline Implementation Details

For fair performance comparison, we present the parameter setting details of some baseline methods. Specifically, the weight for weight-decay regularization is tuned from $\{1e^{-k}|3 \leq k \leq 8\}$ for all baseline methods. The regularization strength of self-supervised learning task to supplement the main recommendation objective, is tuned

from the suggested value range in compared SSL-based recommender systems. For example, the regularization weight for loss balance in NCL, HCCF, SGL, SLRec is tuned from $\{1e^{-k}| - 1 \leq k \leq 6\}$. The temperature parameter in the infoNCE loss for the contrastive learning-based baselines is tuned from $\{0.01, 0.03, 0.1, 0.3, 1, 3\}$. For baseline models that conduct edge or node dropout (*e.g.* LightGCN, SGL), the dropout rate is tuned from $\{0.1, 0.2, 0.3, 0.5, 0.8, 0.9\}$. For methods that were proposed for recommendation with explicit feedbacks (*e.g.* AutoRec, ST-GCN, GCMC), the models are trained using BPR ranking loss. For NCF, we evaluate the results given by the final NeuMF version. For ST-GCN, the reconstruction loss is tuned from $\{1e^{-k}|2 \leq k \leq 6\}$. The number of intents in DGCF is tuned from $\{2, 4, 8\}$. For NCL, K-Means clustering is conducted in each n epochs, $1 \leq n \leq 5$.

(a) Comparasion of different methods gainst noise data on Gowalla dataset

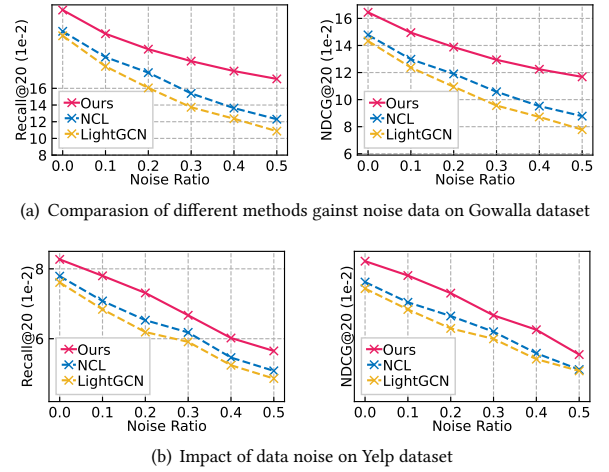(b) Impact of data noise on Yelp dataset

Figure 9: Full version of performance comparison in alleivating interaction noise in terms of Recall@20 and NDCG@20. x-axis represents the ratios of contaminated adversarial noisy user-item interactions in the training set. y-axis shows the performance of different methods against noise data.

A.3 Model Robustness Study

In addition to the presented results in Introduction section, we show the full version of performance comparison of different methods in Figure 9 to study the impact of noisy data on performance. In particular, we corrupt the training set by randomly adding user-item interaction noisy examples with various percentages (*i.e.*, from 10% to 50%). We keep the test set unchanged for all compared methods. The evaluation results show that our AutoCF consistently outperforms state-of-the-art SSL recommender systems under different noise data ratios. It can be seen that our automated generative data augmentation is more beneficial for alleviating noise issue, which brings larger performance gain as the noise percentage increases. Even though current contrastive learning-based methods construct contrasting views for data augmentation, they may introduce noisy auxiliary self-supervision signals. Such noisy signals might mislead the data augmentation process. The superior performance of AutoCF over compared methods, indicates the effectiveness of the designed learnable mask scheme to avoid involving noise in self-supervised augmentation, by considering the infomax-based semantic relatedness among collaborative relationships.

Table 4: Model performance and per-epoch model training time of representative methods on large-scale Tmall dataset.

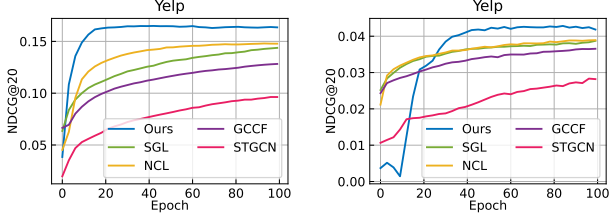| # Edges	Metric	DGCF	SGL	HCCF	NCL	Ours
	R@20	0.0221	0.0258	0.0272	0.0286	**0.0299**
1.6M	N@20	0.0258	0.0296	0.0309	0.0337	**0.0354**
	Time	3569.5s	1009.4s	782.3s	848.0s	892.4s
	R@20	0.0253	0.0278	0.0283	0.0294	**0.0306**
2.9M	N@20	0.0279	0.0311	0.0319	0.0334	**0.0358**
	Time	4828.6s	1187.0s	1040.5s	1015.3s	1158.8s

Figure 12: Convergence analysis *w.r.t* epochs while training on Gowalla and Yelp datasets, in terms of NDCG@20.
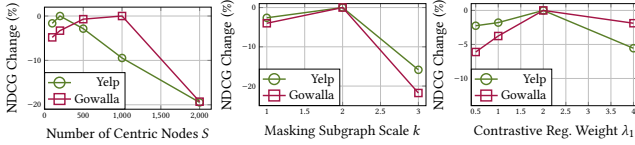

Figure 10: Hyperparameter study for AutoCF in terms of NDCG@20 changes on Yelp and Gowalla datasets.
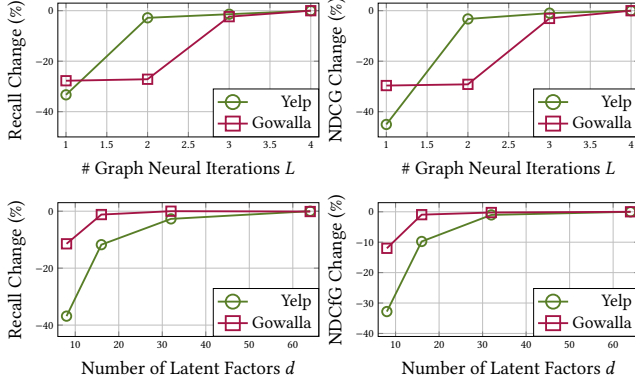

Figure 11: Impact of model depth and dimensionality on the performance of AutoCF, in terms of Recall@20 and NDCG@20 on both Yelp and Gowalla datasets.

A.4 Imapct of Depth and Dimensionality

In this section, we present the results of impact study for model depth investigation and embedding dimensionality. From evaluation results in Figure 11, we have the following observations:

- **Number of Graph Neural Iterations** L: This hyperparameter determines how many iterative graph neural network layers are applied in the masked autoencoder with graph self-attention. As shown in results, too few graph iterations result in insufficient graph connectivity modeling and damage the graph autoencoding learning. However, further increasing the graph iterations yields limited improvements due to oversmoothing.

- **Embedding Dimensionality** d: This is another general hyperparameter for model-based collaborative filtering. It controls the representation capacity of our AutoCF. From the evaluation results, we observe typical underfitting-to-overfitting curves, where the performance increases dramatically with the increase of d at the beginning (from 8 to 16 or 32), and then the improvements become marginal when further increasing d to 64.

Figure 10 presents the supplementary results for Section 4.4.

A.5 Model Scalability Study

In this section, we report the running time of several methods in handling large-scale datasts (with millions of interactions) from an online retail platform. The system configuration for evaluation is Intel Xeon W-2133 CPU, NVIDIA TITAN RTX, with 64GB RAM. From results listed in Table 4, we can still observe that AutoCF is able to achieve competitive efficiency compared with baselines, meanwhile maintaing much better recommendation performance. Convergence study in terms of NDCG@20 are presented in Figure 12 to supplement the results reported in Section 4.5.

A.6 Theoretical Analysis

We first give derivations on the gradients related to $\mathcal{L}_{\text{recon}}$. The reconstruction loss can be decomposed as follows:

$$-\mathcal{L}_{\text{recon}} = \sum_{v_1, v_2} \hat{\mathbf{h}}_{v_1}^\top \cdot \hat{\mathbf{h}}_{v_2} = \sum_{v_1, v_2} (\mathbf{h}_{v_1} + \tilde{\mathbf{h}}_{v_1})^\top (\mathbf{h}_{v_2} + \tilde{\mathbf{h}}_{v_2})$$

$$= \sum_{v_1, v_2} \mathbf{h}_{v_1}^\top \mathbf{h}_{v_2} + \mathbf{h}_{v_2}^\top \tilde{\mathbf{h}}_{v_1} + \mathbf{h}_{v_1}^\top \tilde{\mathbf{h}}_{v_2} + \tilde{\mathbf{h}}_{v_1}^\top \tilde{\mathbf{h}}_{v_2} \qquad (14)$$

For simplicity, here we only consider v_1, v_2 from the masked subgraph, which have no edges in the augmented graph and are thus not influenced by the graph convolutions. $\mathbf{h}_{v_1}, \mathbf{h}_{v_2}$ are randomly-initialized ego embeddings for node v_1, v_2, and $\tilde{\mathbf{h}}_{v_1}, \tilde{\mathbf{h}}_{v_2}$ are graph attention-based embeddings, which can be represented as:

$$\tilde{\mathbf{h}}_v = \sum_{v'} \beta'_{v,v'} \mathbf{W}'_V \mathbf{h}_{v'} \qquad (15)$$

where $\beta'_{v,v'}, \mathbf{W}'_V$ denote the learned attention weights and the value transformation. Here we omit the multi-head setting for simplification. Then, the gradient of $\partial \mathcal{L}_{\text{recon}}/\partial \mathbf{h}_{v_1}$ in Eq 12 is obtained.

Next, we show the derivations for the lower bound of the relatedness between a random pair of nodes v_1 and v'. Assuming the embedding vectors $\mathbf{h}_v, \mathbf{h}_{v_1}, \mathbf{h}_{v'}$ are unit vectors, the pairwise distance can be calculated using a geometric trick, as follows:

$$d_{v,v_1} = 2\sin\frac{\theta_{v,v_1}}{2}, \; d_{v,v'} = 2\sin\frac{\theta_{v,v'}}{2}, \; d_{v_1,v'} = 2\sin\frac{\theta_{v_1,v'}}{2} \quad (16)$$

where $\theta_{v,v_1}, \theta_{v,v'}, \theta_{v_1,v'} \in [0, \pi]$ denote the angles between the corresponding vector pairs. As v, v', v_1 form a triangle (or in a line), the pairwise distances meet the following ristriction:

$$d_{v_1,v'} \le d_{v,v_1} + d_{v,v'}; \quad \sin\frac{\theta_{v_1,v'}}{2} \le \sin\frac{\theta_{v,v_1}}{2} + \sin\frac{\theta_{v,v'}}{2}$$

$$\sin^2\frac{\theta_{v_1,v'}}{2} \le (\sin\frac{\theta_{v,v_1}}{2} + \sin\frac{\theta_{v,v'}}{2})^2 < \sin^2\frac{\theta_{v,v_1}}{2} + \sin^2\frac{\theta_{v,v'}}{2}$$

$$1 - \cos\theta_{v_1,v'} < 2 - \cos\theta_{v,v_1} - \cos\theta_{v,v'}$$

$$\cos\theta_{v_1,v'} > \cos\theta_{v,v_1} + \cos\theta_{v,v'} - 1$$

$$\cos(v_1, v') > \cos(v, v_1) + \cos(v, v') - 1 \qquad (17)$$