Adaptive Fusion Multi-View Dual Contrastive Learning for Recommendation

Kunpeng Li, Dan Yang, Xi Gong

Abstract—**Current recommendation methods built upon graph contrastive learning encounter substantial challenges in modeling heterogeneous graph structures. Specifically, current methods struggle to integrate diverse information from various node types and relationships within heterogeneous graphs. To overcome these limitations, we propose Adaptive Fusion Multi-View Dual Contrastive Learning for Recommendation (AFVCL), a novel recommendation framework. The framework incorporates a dynamic weighting mechanism that adaptively adjusts the importance of each view based on its contribution to the recommendation task, enabling effective integration of diverse heterogeneous data and improving recommendation performance. Moreover, AFVCL applies data augmentation via random node and edge dropout to simulate diverse user behaviors, which improves the model's robustness against data sparsity. To capture complex user-item relationships in heterogeneous graphs more effectively, the proposed framework leverages meta-paths to construct multi-view representations. Meta-paths enhance the graph's contextual information by capturing composite semantic relationships between nodes, offering additional perspectives for modeling user preferences and item characteristics. Within the contrastive learning paradigm, AFVCL integrates multi-view embeddings derived from meta-paths to formulate multi-task optimization objectives. Furthermore, we adopt a dual contrastive learning approach to align semantic representations and enhance the consistency of user-item modeling. Comprehensive evaluations on three real-world datasets demonstrate the efficiency and applicability of AFVCL, underscoring its superiority over existing baseline methods.**

Index Terms—**Recommendation Algorithm, Graph Neural Network, Adaptive Fusion, Contrastive Learning**

I. INTRODUCTION

Recommender systems [1] have become pivotal in delivering personalized services across major digital sectors such as online shopping sites, social platforms, and multimedia streaming applications [2]. Despite their widespread adoption, current recommendation methodologies face significant challenges in modeling intricate user-item interaction patterns, addressing data sparsity limitations, and improving generalization capabilities within dynamic environments characterized by evolving user preferences and complex data ecosystems [3-4]. While conventional collaborative filtering approaches have achieved empirical success, their dependence on historical interaction data restricts adaptability to dynamic interest diversification and hinders the discovery of latent preference patterns [5], particularly in scenarios requiring long-tail item recommendations.

The advent of graph neural networks (GNNs) [6-7] has introduced transformation capabilities for recommender system development through sophisticated modeling of graph-structured relational data. By constructing explicit graph representations of user-item relationships, GNNs facilitate the discovery of higher-order interaction dynamics. However, critical limitations persist when applying these architectures to heterogeneous graph environments, notably in achieving effective cross-perspective information integration, robust latent preference modeling, and accurate user intent representation. Recent advancements in self-supervised learning frameworks, particularly contrastive learning methodologies [8-9], offer promising avenues for addressing data sparsity constraints through representation consistency optimization. While these approaches demonstrate improved recommendation accuracy via semantic invariance preservation, current implementations exhibit notable deficiencies in fully harnessing latent data diversity and capturing intricate cross-feature dependencies. The concurrent objectives of effective data diversity enhancement and robust model training under sparse-data conditions continue to present unresolved challenges in the field.

Although existing graph contrastive learning-based models have achieved notable success in recommender systems, they still encounter the following challenges:

- Effectively integrating information from various types of nodes and relationships in heterogeneous graphs remains a significant challenge. Existing approaches often fail to fully leverage the multi-dimensional information within graphs, leading to suboptimal recommendation performance.
- Contemporary recommendation models typically depend on a single data augmentation strategy, which imposes inherent limitations. When the selected strategy is mismatched with specific graph data, it can cause the omission of key features, thereby weakening the precise representation of user-item interactions and ultimately degrading the quality of recommendations.

To address the challenges outlined above, we put forward an Adaptive Fusion Multi-View Dual Contrastive Learning framework for Recommendation (AFVCL). The framework introduces meta-paths, which capture multi-hop connections among various types of nodes and relationships in

Manuscript received January 8, 2025; revised May 16, 2025.

Kunpeng Li is a postgraduate student at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: 18937861953@163.com).

Dan Yang is a professor at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (corresponding author to provide e-mail: asyangdan@163. com).

Xi Gong is an associate professor at School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China (e-mail: askdjy05gx@163.com).

heterogeneous graphs. We utilize meta-paths to build diversified semantic perspectives, thereby enriching the relational information between users and items. Within the user-item interaction graph, AFVCL employs data augmentation strategies that randomly drop nodes and edges to simulate potentially missing information. This approach enhances data diversity and strengthens the model's robustness. Subsequently, an adaptive fusion strategy is employed to integrate multi-level information from user-item perspectives. Through a dynamic weighting mechanism, the framework adjusts the importance of different perspectives based on their contributions to the recommendation task, thus improving the representation quality of both users and items. Finally, the framework leverages a contrastive learning-based approach to ensure consistency across multi-view representations, thereby enhancing the alignment between user interests and item features. This design further enhances the model's generalization capability. The learned node embeddings are ultimately employed to generate recommendation lists.

The primary contributions of this paper are as follows:

- Two data augmentation strategies, namely node dropout and edge dropout, are introduced to enhance the diversity of training data by graph structural perturbation. These strategies effectively address the issue of data sparsity by simulating perturbations in user behaviors, thereby ensuring system stability of recommender systems in sparse data scenarios and enhancing the model's robustness to missing information.
- A dynamic weighting mechanism is proposed to better integrate information from multiple views. This mechanism automatically adjusts the weights of each view based on its contribution to the recommendation task, thus significantly improving model performance. Particularly in multi-view and heterogeneous data environments, this mechanism enhances system robustness and generalization capabilities.
- A new recommendation algorithm, named AFVCL and based on graph contrastive learning, is proposed to overcome the limitations of existing methods and enhance recommendation performance.
- Extensive experiments on three public datasets—Amazon, Yelp, and Douban-book—demonstrate that AFVCL achieves significant improvements across various performance metrics, surpassing baseline models and effectively addressing challenges such as complex user behaviors and data sparsity.

II. RELATED WORK

A. *Graph Neural Networks*

Graph Neural Networks (GNNs) have become a fundamental technique in recommender systems due to their capability of modeling graph-structured data. By representing user-item interactions as graphs, GNNs effectively capture higher-order interactions. Prior studies [10] proposed a mechanism for learning node representations through message passing between nodes and their neighbors. The Graph Convolutional Network (GCN) [11] employs graph convolution operations within node neighborhoods to model user-item relationships. Further advancements include the Relational Graph Convolutional Network (RGCN) [12], which incorporates specialized convolution operations to process heterogeneous information across different edge types. Based on these foundations, researchers have developed more advanced GNN architectures, such as integrating graph convolution with self-attention to capture long-range dependencies. For instance, NGCF [5] improves recommendation accuracy by modeling high-order connections between users and items, addressing the limitations of traditional collaborative filtering in signal propagation. GraphRec [13] utilizes a hierarchical GNN structure to propagate information, effectively capturing users' long-term preferences and refining user-item similarity. GraphSAGE [14] employs a neighborhood sampling strategy to enhance computational efficiency on large-scale graphs while enabling inductive learning for unseen nodes.

B. *Multi-View Learning in Recommendation*

Multi-view learning is essential in recommender systems as it integrates diverse information sources to enhance representation learning. Conventional recommendation methods often depend on single-view inputs—for example, user-item interaction data—which can overlook rich contextual signals. In contrast, multi-view learning integrates multiple perspectives to enhance recommendation performance. Existing multi-view learning approaches can be classified into feature-level fusion, representation-level fusion, and contrastive learning-based fusion. Feature-level fusion directly integrates raw features from multiple views. For instance, DeepCoNN [15] combines user reviews and ratings to improve representation learning. Representation-level fusion generates distinct embeddings for each view and aggregates them using attention mechanisms or graph-based approaches, as exemplified by MVGCN [16]. Recently, contrastive learning has been utilized to align multi-view representations while preserving their unique characteristics. HeCo [17] and CLCRec [18] improve inter-view consistency by employing contrastive learning objectives. Despite their effectiveness, these approaches often depend on fixed fusion strategies or predefined augmentation techniques. To overcome these limitations, the proposed framework incorporates an adaptive integration strategy that flexibly reweighs each view according to its significance, ensuring optimal information fusion. By leveraging contrastive learning, the framework enforces consistency across views while preserving crucial semantic distinctions, ultimately enhancing recommendation accuracy and robustness.

C. *Recommendation Algorithms Based on Contrastive Learning*

Contrastive learning is a self-supervised technique designed to learn high-quality feature representations by maximizing intra-view similarity and minimizing inter-view differences. Prior research [19] constructs positive and negative sample pairs to train embeddings, improving representation quality without requiring manual labels. The method proposed in [20] encodes users and items, generates positive and negative pairs, and optimizes a contrastive loss function. By aligning similar user-item pairs, this approach enhances recommendation accuracy and generalization. NCF integrates network modeling of user-item interactions with contrastive learning to refine embedding representations, thereby improving recommendation accuracy. SGL [21] captures user interests from graph-structured data, thereby enhancing the performance of collaborative filtering models on sparse datasets. KGCL [22] leverages structural information from knowledge graphs to optimize user-item relationship representations through contrastive learning. By generating contrasting pairs of samples labeled as similar or dissimilar, KGCL enhances the alignment between user preferences and item attributes, which in turn improves both the precision and robustness of recommendations. CLCRec [18] combines collaborative filtering with contrastive objectives by learning representations for users and items, while minimizing the discrepancy between similar and dissimilar pairs through contrastive loss. This method significantly boosts recommendation effectiveness and strengthens the generalization capability of models, especially under sparse data conditions.

III. Preliminaries

To better understand the proposed recommendation algorithm, the following key concepts and notations are first defined.

Definition 1. User-Item Interaction Graph. The user-item interaction graph $G_{\mathrm{ui}} = (U, I, E)$ represents the interaction relationships between users and items. The set of users is denoted as $U = \{u_1, u_2, \cdots, u_m\}$, the set of items as $I = \{i_1, i_2, \cdots, i_n\}$, and E indicates whether an interaction exists between a user and an item, where $E = 1$ if an interaction exists and $E = 0$ otherwise. In this paper, the adjacency matrix $A_{ui} \in R^{m*n}$, which corresponds to the graph G_{ui}, is defined such that m denotes the number of users and n denotes the number of items.

Definition 2. Meta-Path. A meta-path ρ is a sequence of nodes and edges of different types, used to represent composite relationships between nodes. In a heterogeneous graph, different meta-paths reveal distinct dependency information. For example, a meta-path ρ can represent a composite connection from node type X_l to X_{l+1}, expressed as: $X_1 \xrightarrow{T_1} X_2 \xrightarrow{T_2} \dots \xrightarrow{T_l} X_{l+1}$. Through meta-paths, connections between nodes in the graph can be established based on specific relationships, thereby enriching the interaction information between nodes.

Definition 3. Subgraph Based on Meta-Path. Given a specific meta-path ρ and a set of nodes V, the subgraph induced by the meta-path ρ, denoted as $G_{\rho,V}$, consists of all nodes in V and the edges connecting them via ρ. Specifically, for a node set $v \in V$, all neighboring nodes connected via the meta-path ρ, along with their associated edge set $E_\rho(v)$, form part of this subgraph. By traversing all nodes in $v \in V$, the complete edge set E can be obtained, thereby constructing a complete subgraph based on the meta-path. Formally, the subgraph $G_{\rho,V} = (V_\rho, E_\rho)$ based on the meta-path ρ is defined such that V_ρ represents the set of all nodes connected via ρ, and E_ρ represents the set of edges between these nodes.

TABLE I
SYMBOLS AND THEIR MEANINGS

Symbol	Meaning
G_{ui}	user-item interaction graph
U,I,E	user set, item set, interaction set
R	the rating matrix of users on items
u,i	user node and item node
m,n	number of users, number of items
X_l, T_l	Node type and edge type sets
ρ	meta-path
$G_{\rho,V}$	subgraph based on meta-math

The definitions of the notations used in this paper are summarized in Table I.

IV. Recommendation Algorithm Framework

This part offers an in-depth explanation of the proposed recommendation algorithm framework, AFVCL, with its specific structure illustrated in Fig.1. AFVCL aims to boost recommendation performance in heterogeneous information contexts by leveraging contrastive learning and multi-view integration. The core of the framework consists of the primary user-item graph and auxiliary heterogeneous information graphs. The auxiliary graphs are constructed using meta-paths to enable multi-view enhancement, capturing diverse features of user preferences and item attributes. The framework integrates data augmentation techniques and adaptive fusion strategies to extract features from both the direct user-item graph and higher-order neighbor structures. It further applies contrastive objectives to ensure alignment and consistency among multiple views. In particular, contrastive learning introduces dual contrastive learning(DCL): intent-to-intent and intent-to-interaction, further improving the semantic expressiveness of user and item embeddings. Finally, through the joint optimization of multi-task losses, including recommendation loss, contrastive loss, and regularization terms, AFVCL achieves precise modeling of user interests and item features, ultimately generating a Top-N recommendation list.

A. Model Input

The model takes as input the central user-item interaction graph along with supplementary heterogeneous information graphs. These inputs are employed to construct a multi-view representation space, which enriches the understanding of user interests and item characteristics.

1) User-Item Graph

At the heart of the model lies the user-item interaction graph, which encodes behavioral relationships between users and items. The embedding matrices for users and items are defined as $E_u \in R^{m \times d}$ and $E_i \in R^{n \times d}$, respectively, where m and n denote the numbers of users and items, and d is the embedding dimension. E_u and E_i are used as globally shared embedding parameters during parallel training.

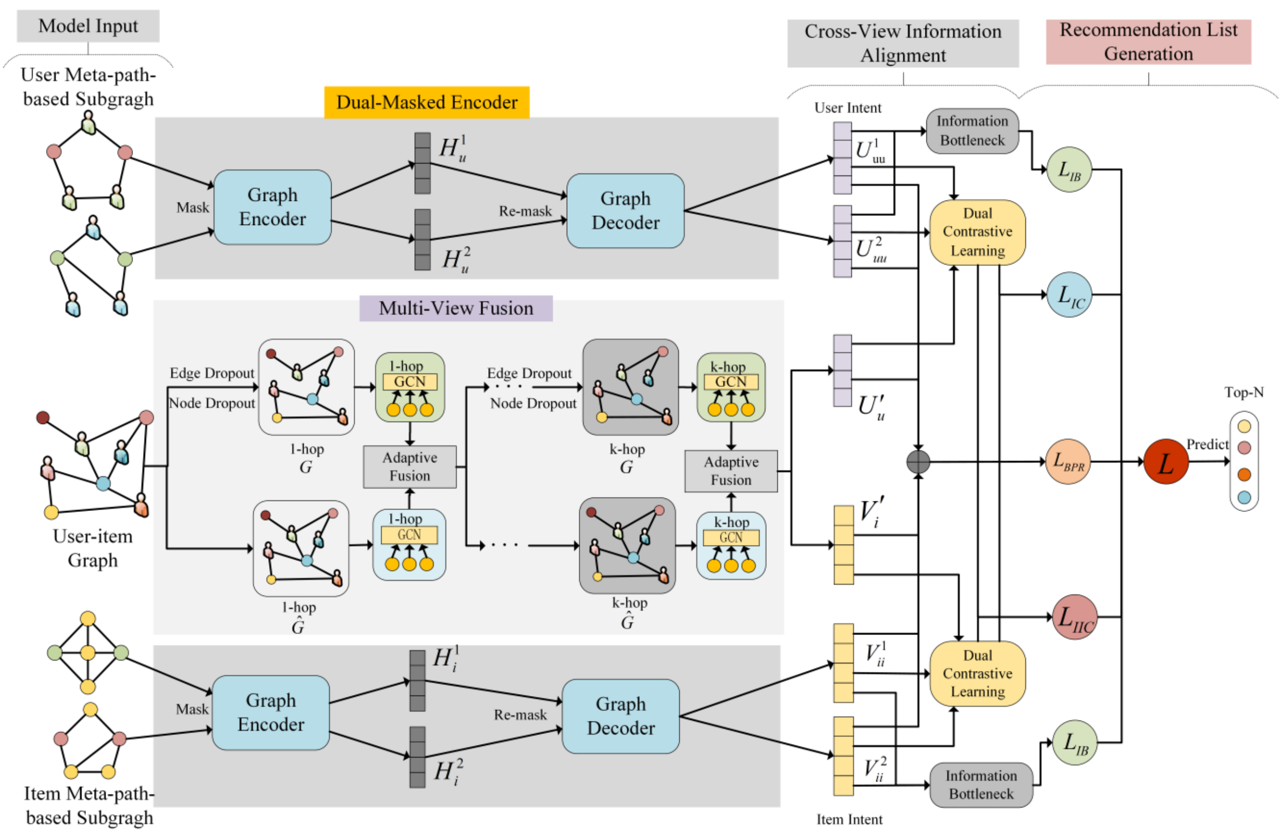

Fig.1. The overall architecture of AFVCL

2) Heterogeneous Information Graph

To construct multi-view augmentations, the model leverages meta-paths to build heterogeneous information graphs, where each meta-path reveals the semantic associations between users and items. Two meta-paths are selected respectively from the perspectives of users and items to generate the corresponding subgraphs:

$$\text{User: } G_u^{\rho_u^1}, G_u^{\rho_u^2} \quad \text{Item: } G_i^{\rho_i^1}, G_i^{\rho_i^2}$$

Here, ρ_u^k denotes the definition of the k-th meta-path related to users, and the meta-paths for items are defined in a similar manner.

B. *Dual-Masked Information Bottleneck Encoder*

This study proposes the Dual-Masked Information Bottleneck Encoder (DMIB) to model the rich intents in heterogeneous information while mitigating noise interference. By learning the global data structure, the encoder can better reconstruct features affected by noise. Additionally, incorporating an information bottleneck strategy helps improve the fidelity of reconstruction by preserving only the essential data necessary for accurate recommendations.

Multi-view subgraphs for users and items are generated through meta-path guidance, aiming to reflect the diverse characteristics of user interests and item properties. Studies [23] suggest that the similarity structure represented by each meta-path can function as an individual perspective. For users and items, two subgraphs are defined for each, generated using the selected meta-paths:

$$G_u^{\rho_u^k} = \left(V_u, X_u^{\rho_u^k}, E_u\right) \quad G_i^{\rho_i^k} = \left(V_i, X_i^{\rho_i^k}, E_i\right) \tag{1}$$

Here, $k=1,2$ represents different meta-paths, $X_u^{\rho_u^k}$ and $X_i^{\rho_i^k}$ refer to the corresponding adjacency matrices for users and items under each meta-path. Similarly, E_u and E_i represent the respective embedding representations for users and items.

To enhance the robustness of the model, random masking is first applied to the nodes in the meta-path subgraphs. A random sampling strategy without replacement is employed to select the set of masked nodes:

$$V_u^{mask} = \left\{v \in V_u \middle| r_v \le p\right\} \tag{2}$$

$$V_i^{mask} = \left\{v \in V_i \middle| r_v \le p\right\} \tag{3}$$

Here, $r_v \sim Uniform(0,1)$ and p denote the masking ratios. When a node's entire neighborhood is either entirely hidden or entirely retained, a uniform random selection strategy is employed. This sampling strategy avoids complete reliance on the neighborhood information of specific nodes, thereby enhancing the model's generalization capability.

The embeddings of all nodes in set V^{mask} are masked to obtain $\hat{E}_u = \{e_u^1, e_u^2, \dots, e_u^m\}$ and $\hat{E}_i = \{e_i^1, e_i^2, \dots, e_i^n\}$. A lightweight graph neural network (LightGCN) is then employed as the encoder to process the node embeddings, performing feature extraction and information propagation. The user and item embedding propagation process is formulated as follows:

$$h_u^{(l_e)} = \sum\nolimits_{j \in N(u)} \frac{1}{\sqrt{|N(u)|}\sqrt{|N(v)|}} h_j^{(l_e-1)} \tag{4}$$

$$h_i^{(l_e)} = \sum\nolimits_{v \in N(i)} \frac{1}{\sqrt{|N(i)|}\sqrt{|N(v)|}} h_v^{(l_e-1)} \tag{5}$$

Here, l_e denotes the total number of layers used in the encoder, h_j^0 and h_v^0 denote the embeddings of nodes $\hat{E}_u$ and $\hat{E}_i$, respectively, and $N(u)$ and $N(i)$ represent the neighborhood sets of users and items. The resulting initial embeddings after reconstruction are $H_u = \{h_u^1, h_u^2, \ldots, h_u^m\}$ and $H_i = \{h_i^1, h_i^2, \ldots, h_i^n\}$. During the process of information aggregation, nodes typically gather information from their neighbors, effectively capturing local structural information. Nevertheless, such an operation can inadvertently increase reliance on certain nodes. In scenarios with a low masking rate, some node representations may retain noise inherited from raw features, which undermines the benefits of data augmentation. To mitigate this problem, a consistent sampling mechanism is applied to the encoded representations, generating a newly masked set V^{mask}. This set is used to mask the reconstructed embeddings, producing $\hat{H}_u$ and $\hat{H}_i$ to further denoise the data. These denoised embeddings are then passed through a decoder for reconstruction. The formula for reconstructing the embeddings is as follows:

$$U_{uu} = \left(D_u^{-0.5} X_u D_u^{-0.5}\right)^{l_d} \hat{H}_u \tag{6}$$

$$V_{ii} = \left(D_i^{-0.5} X_i D_i^{-0.5}\right)^{l_d} \hat{H}_i \tag{7}$$

Here, l_d indicates the total number of layers within the decoder, while D_u and D_i correspond to the degree matrices. The final output embeddings are denoted by U_{uu} and V_{ii}. When incorporating multiple auxiliary heterogeneous views derived from different input sources, the outputs can be expressed as U_{uu}^1, U_{uu}^2, V_{ii}^1 and V_{ii}^2.

To enhance the model's focus on critical information, information bottleneck regularization is introduced. This regularization limits the shared information captured by the embeddings and the graph structure, while enhancing their alignment with the recommendation objective. Let $(U_{uu} + H_u)$ and $(V_{ii} + H_i)$ be defined as S_U and S_I, respectively. The mutual information loss for users and items is defined as:

$$L_{uib} = -I\left(S_U^k; Y_{\text{Rec}}\right) + \beta \cdot I\left(X_u^k; S_U^k\right) \tag{8}$$

$$L_{iib} = -I\left(S_I^k; Y_{\text{Rec}}\right) + \beta \cdot I\left(X_i^k; S_I^k\right) \tag{9}$$

Here, $I(\cdot;\cdot)$ denotes the mutual information function, β controls the strength of regularization, and Y_{Rec} represents the recommendation signals corresponding to BPR (Bayesian Personalized Ranking) interaction pairs in the recommendation task. $k = 1, 2$ indicates the different meta-paths.

This module effectively mitigates noise interference while preserving core semantic information, laying a solid foundation for subsequent contrastive learning and recommendation tasks.

C. Adaptive View Fusion

1) Assisted Neighbor Graph Construction

To better extract high-order collaborative signals from the underlying interaction graph G, this paper introduces a collaborative neighbor view $\hat{G}$, which encodes the structural resemblance among users and among items. This view is derived from the user-item interaction graph G by analyzing shared neighbor relationships in the interaction records. Specifically, collaborative similar users are defined as those sharing analogous preferences, whereas items are considered collaboratively similar if they have been co-interacted with by identical users. The collaborative similarity is computed using the Jaccard similarity coefficient, as formulated below:

$$sim_{i,j} = \frac{\left|N_G(i) \cap N_G(j)\right|}{\left|N_G(i) \cup N_G(j)\right|} \tag{10}$$

Here, $N_G(i)$ and $N_G(j)$ represent the first-order neighbor sets of i and j, respectively. Based on the collaborative similarity, the adjacency matrix $\hat{A}$ of the collaborative neighbor view is defined as follows:

$$\hat{A}_{i,j} = \begin{cases} sim_{i,j} & sim_{i,j} \geq \eta \text{ or Top-K values for node i} \\ 0 & \text{others} \end{cases} \tag{11}$$

The specific values of the threshold NNN and Top-K are determined based on the characteristics of the dataset. This definition effectively filters out irrelevant collaborative information while preserving high-quality higher-order neighbor information, thereby mitigating the issue of data sparsity.

2) Data Augmentation

During the training process, a data augmentation strategy is applied to improve model robustness, particularly by introducing perturbations to the graph structure. In this process, the graph's adjacency matrix A is subjected to node dropout. Node dropout involves the random removal of a subset of nodes, thereby generating a new adjacency matrix $\bar{A}$. The procedure is as follows:

$$\bar{A} = node\_dropout(A, p) \tag{12}$$

Here, p represents the probability of retaining a node. This operation introduces noise into the data by randomly removing nodes from the graph, which in turn improves the model's robustness against input variations. Additionally, edge dropout further perturbs the connectivity between nodes by selectively removing edges, based on the retained nodes. The procedure is as follows:

$$\tilde{A} = edge\_dropout(\bar{A}, q) \tag{13}$$

Here, q denotes the dropout probability of the edges. This operation weakens the influence of highly connected nodes by randomly removing a subset of edges, thereby increasing the diversity of the graph structure.

3) Adaptive Fusion Strategy

After acquiring the representations from the user-item view $\tilde{G}$ and the collaborative neighbor view $\hat{G}$, a flexible fusion mechanism is applied to selectively combine the feature representations from both views, enabling the capture of diverse aspects of user preferences and item properties.

At each layer of the Graph Neural Network (GNN), the node embeddings are updated through the main view $\tilde{G}$ and the collaborative neighbor structure $\hat{G}$, denoted as $h_i^{G,l}$ and $h_i^{\hat{G},l}$, respectively. Subsequently, the embeddings from both views are fused using the fusion strategy to generate the final embedding h_i^l at the l-th layer, which serves as input to the subsequent GNN layer for continued refinement. Similar to LightGCN, the message passing process in this study discards non-linear activations and feature transformations, performing efficient and straightforward embedding updates solely based on the graph structure. The procedure is as follows:

$$h_i^{G,i} = \sum_{j \in N_G(i)} \frac{1}{\sqrt{|N_G(i)|}\sqrt{|N_G(j)|}} h_j^{l-1} \tag{14}$$

$$h_i^{\hat{G},l} = \sum_{j \in N_{\hat{G}}(i)} \frac{1}{|N_{\hat{G}}(i)|} h_j^{l-1} \tag{15}$$

For the main view $\tilde{G}$ and the collaborative neighbor view $\hat{G}$, distinct normalization strategies are applied. To jointly learn both direct interactions in the user-item graph $\tilde{G}$ and indirect, higher-level associations such as user-user or item-item relations in $\hat{G}$, the model employs an adaptive fusion strategy that dynamically combines $h_i^{G,l}$ and $h_i^{\hat{G},l}$ to generate the final embedding h_i^l. Unlike attention-based fusion methods, this strategy does not rely on parameterized weights, thus avoiding performance degradation due to premature convergence. The adaptive fusion strategy dynamically adjusts the contribution of higher-order information based on the layer number, the importance of higher-order information, and the similarity between the embeddings of the main view and the collaborative view, thereby generating more accurate node representations. The formula is as follows:

$$h_i^l = h_i^{G,l} + \beta_i h_i^{\hat{G},l} \tag{16}$$

$$\beta_i = \frac{\lambda}{l + sim\left(h_i^{G,l}, h_i^{\hat{G},l}\right) \cdot d_i} \tag{17}$$

$$d_i = \frac{\log\left(|N_G(i)|\right)}{\frac{1}{|V|}\sum_{v \in V} \log\left(|N_G(v)|\right)} \tag{18}$$

Here, β_i represents the adaptive weight for the collaborative view, $sim(\cdot,\cdot)$ denotes the cosine similarity, d_i is the normalized user activity degree, and λ is a hyperparameter. This design offers several advantages. First, deeper GNN architectures may introduce adverse impacts due to the influence of higher-order neighbor information. Thus, the weight decreases with the layer depth, mitigating its disruptive influence. Second, the first-order neighbor information for highly active nodes is typically sufficient, warranting a reduction in the impact of higher-order neighbors. Furthermore, when the similarity between the embeddings of the main view and the collaborative view is high, the adaptive weights help reduce the influence of redundant information, consequently improving recommendation precision and user-specific relevance.

After completing the propagation through L layers, the model combines the fused embeddings from each layer (h_u^l and h_i^l, where $l \in [0, L]$ denotes the layer number) with the initial embeddings h_u^0 and h_i^0 using mean pooling, thereby generating the final user and item representations. The specific formula is as follows:

$$U_u = \frac{1}{L+1}\sum_{l=0}^{L} h_u^l \tag{19}$$

$$V_i = \frac{1}{L+1}\sum_{l=0}^{L} h_i^l \tag{20}$$

By performing mean pooling across the embeddings from all layers, the model is able to integrate information from different levels, thereby enhancing the robustness of the final representations. The generated user representation U_u and item representation V_i are then used for subsequent recommendation task predictions.

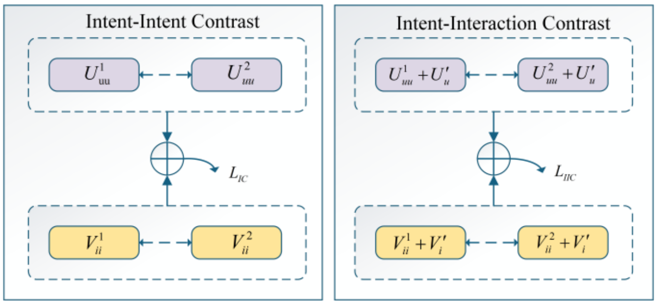

Fig. 2. Dual Contrastive Learning.

D. *Top-N Candidate Recommendation*

In response to the shortcomings of current graph contrastive learning approaches in handling multi-view data, the Dual Contrastive Learning (DCL) framework is proposed, as illustrated in Fig. 2. This framework aligns the diversified semantics of users and items along meta-paths, while dynamically integrating the user-item views through an adaptive fusion strategy to capture global consistency. Specifically, DCL consists of two components: intent-intent contrast and intent-interaction contrast. These components are designed to align semantic relationships across heterogeneous views and to capture the correlation between intent and actual interactions, respectively.

1) Intent-Intent Contrast

The intent-intent contrast mechanism is designed to match user and item representations across different semantic views. Let the user embeddings generated from two different meta-paths be U_{uu}^1 and U_{uu}^2, with each user node represented by u' and u'', respectively. To enhance the mutual information between the two views, the framework utilizes the InfoNCE loss. The intent-intent contrast loss (IC) for the user is defined as follows:

$$L_{IC}^U = \sum_{i \in B} -\log \frac{\exp\left(sim\left(u_i', u_i''\right)/\tau\right)}{\sum_{j \in B}\exp\left(sim\left(u_i', u''\right)/\tau\right)} \tag{21}$$

Here, B represents the sampled training batch, $sim(\cdot,\cdot)$ indicates the cosine similarity measure, and τ denotes the temperature parameter. The contrastive loss L_{IC}^I for the items is computed in a similar manner. Through this contrastive learning approach, the IC captures the similarity and intents alignment of users and items across different meta-paths.

2) Intent-Interaction Contrast

The intent-interaction contrast module integrates embeddings derived from both the primary user-item graph and heterogeneous perspectives to capture the interaction intentions between users and items. For example, for user embeddings, the view fusion is represented by $U_u + U_{uu}^1$ and $U_u + U_{uu}^2$, with each user node denoted as z' and z'', respectively. The intent-interaction contrast loss (IIC) is defined as follows:

$$L_{IIC}^{U} = \sum_{i \in B} -\log \frac{\exp\left(sim\left(z_i', z_i''\right)/\tau\right)}{\sum_{j \in B} \exp\left(sim\left(z_i', z_j''\right)/\tau\right)} \quad (22)$$

Here, τ denotes the temperature parameter. The contrastive loss L_{IIC}^{I} for items is calculated using an identical approach. This module further aligns the semantic relationships between users and items in terms of both intents and actual interactions.

3) Recommendation Prediction

In order to improve the effectiveness of self-supervised recommendation, the model adopts a multi-task learning paradigm to formulate the overall optimization objective. First, the loss functions of the model's primary modules are integrated, including the Information Bottleneck-based Dual-Masked Information Bottleneck Encoder (DMIB) and the Dual Contrastive Learning (DCL) module. DMIB utilizes the information bottleneck principle to denoise the autoencoder, with the loss function defined as follows:

$$L_{IB} = L_{uib} + L_{iib} \quad (23)$$

Since the user and item views are symmetric, there is no need to assign separate weights for each. The DCL module follows a hierarchical structure, integrating the intent-intent contrast loss and intent-interaction contrast loss. The loss function is defined as follows:

$$L_{DCL} = \alpha_{IC} \cdot \left(L_{IC}^{U} + L_{IC}^{I}\right) + \alpha_{IIC} \cdot \left(L_{IIC}^{U} + L_{IIC}^{I}\right) \quad (24)$$

Here, α_{IC} and α_{IIC} represent the weights for intent-intent contrast and intent-interaction contrast, respectively. Furthermore, to improve the quality of recommendations, the model incorporates the Bayesian Personalized Ranking (BPR) loss derived from the recommendation objective, defined as follows:

$$L_{BPR} = -\frac{1}{|B|} \sum_{(i,j,k) \in B} \log \sigma\left(d_i^{\mathrm{T}} d_j - d_i^{\mathrm{T}} d_k\right) \quad (25)$$

Here, B represents the training data, which includes the positive sample i and negative sample j for user u, along with the embedding $d \in \left\{U_u + U_{uu}^1 + U_{uu}^2\right\}$.

The comprehensive optimization objective function is presented below:

$$L = L_{BPR} + \beta_1 \cdot L_{IB} + L_{DCL} + \beta_2 \cdot \|\Psi\|_2^2 \quad (26)$$

Here, β_1 and β_2 represent the weights for the information bottleneck loss and the regularization term, respectively, while $\|\Psi\|_2^2$ denotes the L_2-regularization of the trainable model parameters. Since the weights for the DCL are already determined by α_{IC} and α_{IIC}, there is no need to redundantly set weights in the overall loss function. The combined optimization goal not only boosts recommendation effectiveness but also strengthens the model's robustness and its capacity to generalize across heterogeneous information.

TABLE II
STATISTICS OF THE EXPERIMENTAL DATASETS

Datasets	Amazon	Yelp	Douban-book
User	6170	16239	13024
Item	2753	14284	22347
Interaction	195791	198397	792062
Meta-paths	UIU, UIBIU, IBI, ICI	UU, UBU, BCiB, BCaB	UU, UGU, BAB, BYB

V. EXPERIMENTAL EVALUATION AND ANALYSIS

This section offers a comprehensive overview of the datasets utilized in the experiments, the evaluation criteria applied, and the baseline models chosen. Then, it presents a detailed analysis of the performance of AFVCL based on the evaluation metric data obtained from the experiments.

A. Datasets and Evaluation Metrics

The evaluation of AFVCL and baseline methods was conducted on three publicly available datasets: Amazon, Yelp, and Douban-book. The statistical details of these datasets are summarized in Table II.

- Amazon: This dataset comprises product review information collected from the Amazon platform, aimed at studying the associations between users and items. It constructs complex user-item relationships through user-item interactions (UI), user-user similarity (UIU), user-item interactions combined with item similarity (UIBIU), item-item relationships (IBI), and item-category associations (ICI).
- Yelp: This dataset contains user rating records for businesses collected from the Yelp platform. Rated businesses are treated as interacted items, while unrated ones are considered non-interacted items. It models user-user relationships (UU), user-business interactions (UBU), business similarity (BCiB), and business-category relationships (BCaB) to analyze user preferences for various types of businesses.
- Douban-book: This dataset derives from user rating data on the Douban Book platform, this dataset captures user-book interactions. It explores user-user relationships (UU), user-book category interactions (UGU), book similarity (BAB), and book-category relationships (BYB) to reveal user preferences for books across different categories.

Recall@N and Normalized Discounted Cumulative Gain (NDCG@N) with N={5,10,20} are used in this study as evaluation metrics to assess recommendation effectiveness. Recall@N evaluates the coverage of user interactions by the recommended list, indicating the proportion of actual interacted items successfully predicted. NDCG@N assesses the ranking quality of the recommendations, reflecting whether highly relevant items are positioned at the top of the recommended list.

B. Baselines

To evaluate AFVCL, ten baseline methods derived from various research approaches were selected for comparison:

- SimGCL[22]: Introduces random noise directly into feature representations, simplifying augmentation strategies in graph contrastive learning and reducing reliance on complex data transformations.
- SGL[21]: Constructs augmented views for contrastive learning by generating subgraphs through random walks and randomly removing edges and nodes.
- LightGCN[24]: Proposes a simplified graph convolutional network structure by omitting embedding weights and nonlinear activation, retaining only neighbor aggregation to improve efficiency.
- DisenHAN[25]: Leverages a disentangled learning approach within a heterogeneous graph attention network to model different types of node relationships, capturing latent semantic information of users and items.
- DCCF[26]: Embeds user and item features via local and global convolutional channels and employs multi-scale modeling to extract high-order interaction relationships.
- BIGCF[27]: Enhances recommendation performance by introducing dual graph neural networks within a collaborative filtering framework, modeling both user-item and user-user graphs.
- HAN[23]: Proposes a heterogeneous graph attention network that models node relationships through meta-paths, effectively capturing semantic features in heterogeneous information.
- HeCo[17]: Employs contrastive learning to enhance meta-path representations in heterogeneous graphs, optimizing the representation of associations between different types of nodes.
- SMIN[28]: Utilizes multi-interest modeling to disentangle users' diverse preferences, improving the precision and robustness of recommender systems.
- IHGCL[29]: Integrates multi-path and multi-view augmentation mechanisms into a heterogeneous graph-based contrastive learning framework, significantly enhancing recommendation accuracy and ranking quality.

C. Parameter Setting

For a fair comparison, the parameters of all baseline methods were configured to their optimal values as reported in their respective original papers. Meanwhile, consistent settings were maintained by fixing the batch size at 4096 and the number of training epochs at 100. For the proposed AFVCL framework, the temperature hypernatremia τ was set to 0.2. AFVCL was implemented using PyTorch, with the Adam optimizer employed at a learning rate of 0.001.

D. Analysis of Experimental Results

The results of the experiments, as shown in Table III, allow us to draw the following conclusions:

- The results demonstrate that contrastive learning-based models significantly outperform traditional recommendation models. Moreover, models incorporating multi-view fusion and augmentation mechanisms (e.g., SGL, SimGCL) further enhance recommendation performance. This underscores the critical role of contrastive learning and view augmentation in improving model effectiveness.
- The AFVCL model consistently outperforms all baseline methods, demonstrating significant gains across every evaluation metric. This indicates that AFVCL effectively integrates user and item information from diverse meta-path views through multi-view contrastive learning, thereby capturing user interests and item characteristics more comprehensively to enhance recommendation

TABLE III
PERFORMANCE COMPARISON OF AFVCL WITH BASELINES ON THREE DATASETS

Datasets	Metrics	SimGCL	SGL	LightGCN	DisenHAN	DCCF	BIGCF	HAN	HeCo	SMIN	IHGCL	AFVCL
Amazon	R@5	0.0742	0.0704	0.0653	0.0608	0.0718	0.0743	0.0546	0.0618	0.0640	0.0762	**0.0857**
	N@5	0.1011	0.0948	0.0875	0.0821	0.0990	0.1014	0.0748	0.0840	0.0873	0.1040	**0.1148**
	R@10	0.1197	0.1084	0.1028	0.0958	0.1147	0.1193	0.0885	0.0995	0.1031	0.1230	**0.1296**
	N@10	0.1138	0.1041	0.0976	0.0905	0.1127	0.0139	0.0832	0.0934	0.0969	0.1161	**0.1245**
	R@20	0.1751	0.1646	0.1592	0.1508	0.1758	0.0340	0.1385	0.1519	0.1569	0.1805	**0.1893**
	N@20	0.1306	0.1215	0.1151	0.1098	0.1307	0.0188	0.0988	0.1097	0.1135	0.1346	**0.1426**
Yelp	R@5	0.0398	0.0371	0.0350	0.0317	0.0379	0.0405	0.0264	0.0318	0.0336	0.0425	**0.0476**
	N@5	0.0441	0.0414	0.0406	0.0349	0.0427	0.0446	0.0316	0.0352	0.0352	0.0473	**0.0521**
	R@10	0.0661	0.0622	0.0584	0.0519	0.0632	0.0658	0.0423	0.0523	0.0540	0.0693	**0.0724**
	N@10	0.0525	0.0509	0.0471	0.0409	0.0504	0.0519	0.0357	0.0412	0.0409	0.0546	**0.0582**
	R@20	0.1003	0.0961	0.0883	0.0820	0.0973	0.1009	0.0734	0.0823	0.0854	0.1044	**0.1098**
	N@20	0.0616	0.0604	0.0555	0.0500	0.0605	0.0620	0.0440	0.0497	0.0500	0.0649	**0.0686**
Douban-book	R@5	0.0752	0.0728	0.0626	0.0567	0.0725	0.0742	0.0480	0.0564	0.0597	0.0800	**0.0899**
	N@5	0.1560	0.1547	0.1353	0.1274	0.1541	0.1551	0.0984	0.1284	0.1330	0.1616	**0.1779**
	R@10	0.1147	0.1136	0.0968	0.0909	0.1137	0.1149	0.0801	0.0905	0.0934	0.1190	**0.1248**
	N@10	0.1519	0.1507	0.1312	0.1245	0.1506	0.1535	0.0975	0.1248	0.1298	0.1556	**0.1663**
	R@20	0.1701	0.1665	0.1471	0.1384	0.1665	0.1679	0.1255	0.1387	0.1433	0.1726	**0.1812**
	N@20	0.1579	0.1551	0.1370	0.1311	0.1556	0.1591	0.1076	0.1299	0.1339	0.1611	**0.1704**

quality. Additionally, the incorporation of an adaptive fusion strategy enables dynamic adjustment of contributions from different views, mitigating noise interference introduced by single-view augmentation. By optimizing representation consistency across views via contrastive learning, AFVCL further strengthens its recommendation performance.

E. Variant Analysis

To validate the effectiveness of the data augmentation and adaptive fusion strategies in enhancing the model performance of AFVCL, three variants—AFVCL-a, AFVCL-n, and AFVCL-d—were created by removing specific modules. Specifically, AFVCL-a is the variant where the adaptive fusion module is removed. AFVCL-n excludes the data augmentation module and relies solely on the original features for embedding learning. AFVCL-d removes the data augmentation module and replaces the adaptive fusion module with an attention mechanism.

TABLE IV
PERFORMANCE COMPARISON OF AFVCL WITH OTHER ABLATION METHODS

Datasets	Metrics	AFVCL-a	AFVCL-n	AFVCL-d	AFVCL
Amazon	R@10	0.1259	0.1283	0.1274	**0.1296**
	N@10	0.1220	0.1241	0.1220	**0.1245**
	R@20	0.1857	0.1876	0.1863	**0.1893**
	N@20	0.1404	0.1417	0.1402	**0.1426**
Yelp	R@10	0.0694	0.0704	0.0705	**0.0724**
	N@10	0.0557	0.0568	0.0572	**0.0582**
	R@20	0.1083	0.1096	0.1082	**0.1098**
	N@20	0.0660	0.0677	0.0676	**0.0686**
Douban-book	R@10	0.1204	0.1223	0.1214	**0.1248**
	N@10	0.1615	0.1640	0.1632	**0.1663**
	R@20	0.1782	0.1805	0.1784	**0.1812**
	N@20	0.1669	0.1698	0.1678	**0.1704**

The three variants were evaluated on the Amazon and Yelp datasets, with the results summarized in Table IV. The following conclusions can be drawn:

- AFVCL-a: After removing the adaptive fusion module, the model's performance significantly deteriorates, especially on datasets where higher-order user-user and item-item connections play a key role. This suggests that the adaptive fusion strategy effectively integrates multi-view information and dynamically adjusts the weights of different views, thus improving the precision of the embedding representation and boosting the overall recommendation effectiveness.
- AFVCL-n: After removing the data augmentation module, the model shows varying degrees of performance degradation on both datasets, with a more pronounced decline in sparse data scenarios. This validates that the data augmentation module, through node and edge dropping strategies, effectively increases the diversity of the data and prevents the model from overly relying on the original structural information, significantly improving the model's robustness and generalization ability.
- AFVCL-d: Replacing the adaptive fusion module with an attention mechanism results in a decrease in model performance, indicating that attention mechanisms face convergence instability issues when fusing multi-view information, leading to the introduction of redundant information. In contrast, the adaptive fusion strategy dynamically adjusts weights based on node activity and embedding similarity without requiring parameter learning, resulting in more stable and efficient performance.

F. Parameter Analysis

This section investigates how crucial parameters influence the performance of AFVCL, based on experimental results from the Amazon and Yelp datasets. The analysis includes the model's behavior under different parameter settings, such as the control parameter λ for adaptive fusion weights, the number of message propagation layers and the temperature parameter τ. By comparing these parameters, their influence on recommendation performance is assessed, further validating the model's stability and robustness.

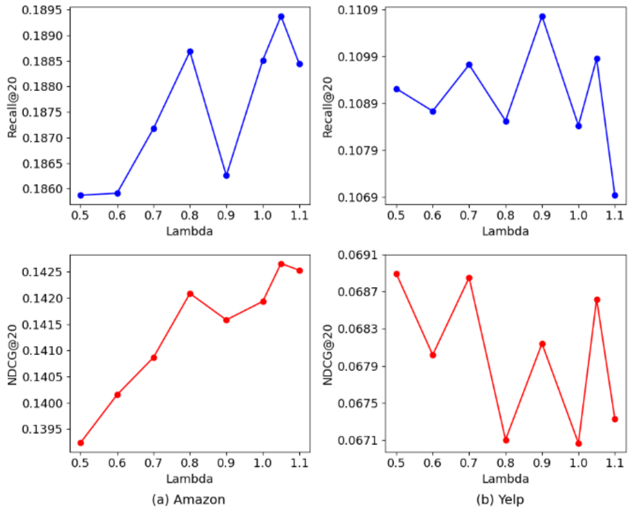

Fig. 3. Performance Comparison of different λ

1) *Impact of Parameter λ*

The parameter λ determines the weight of higher-order information in embedding fusion, which has a significant effect on model performance. The results, shown in Fig 3, indicate that varying λ within different ranges produces noticeable changes in performance. When λ is set too low, the model fails to sufficiently leverage higher-order information, resulting in suboptimal performance. Conversely, excessively large λ values assign too much weight to higher-order information, potentially introducing redundancy or even noise, thereby diminishing the model's ability to learn primary features. A balanced λ value effectively reconciles low-order and high-order information, maximizing the complementary role of higher-order features in embedding representation and significantly improving recommendation performance. The results highlight the critical role of higher-order information in enhancing model performance.

2) *Impact of Propagation Layers*

To evaluate how varying the depth of the graph neural network (GNN) affects model effectiveness, we tested configurations with 1 to 4 propagation layers. The results, presented in Fig 4, show that as the number of GCN layers increases, the recommendation performance first improves and then declines. When the number of layers is set to 3, both

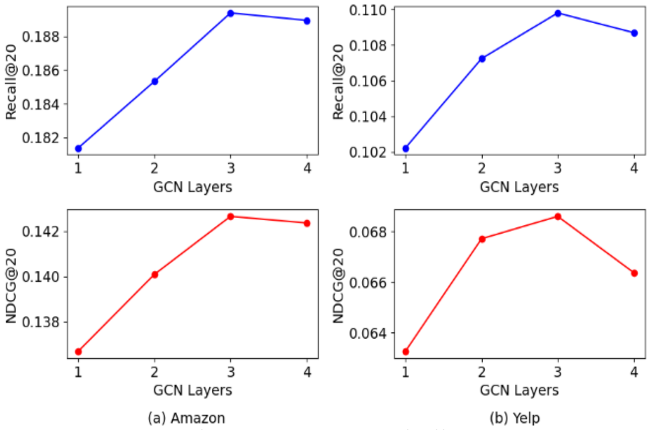

Fig. 4. Performance Comparison of different Layers.

Recall and NDCG achieve their highest values, indicating that the model effectively aggregates information from neighboring nodes and captures richer semantic features at this depth. Nonetheless, a further increase to 4 layers results in a slight performance drop, likely due to over-smoothing or noise accumulation in deeper representations. This decline can be attributed to over-smoothing of the embeddings caused by excessive layers, which reduces the differentiation among nodes and impairs the model's ability to represent user and item characteristics effectively.

3) Impact of Temperature Parameter τ

The temperature parameter τ serves as a crucial hyperparameter in contrastive learning, significantly influencing the model's ability to distinguish between positive and negative samples. To explore its impact, we conducted experiments with τ values ranging from 0.05 to 0.5. The corresponding results are illustrated in Fig. 5. The findings reveal that model performance fluctuates with changes in τ, with significant improvements in recommendation effectiveness only achieved when τ is set within an appropriate range. When τ is too small or too large, the efficacy of contrastive learning diminishes, undermining its ability to optimize model performance. Therefore, selecting a suitable value for τ is essential for enhancing the effectiveness of contrastive learning.

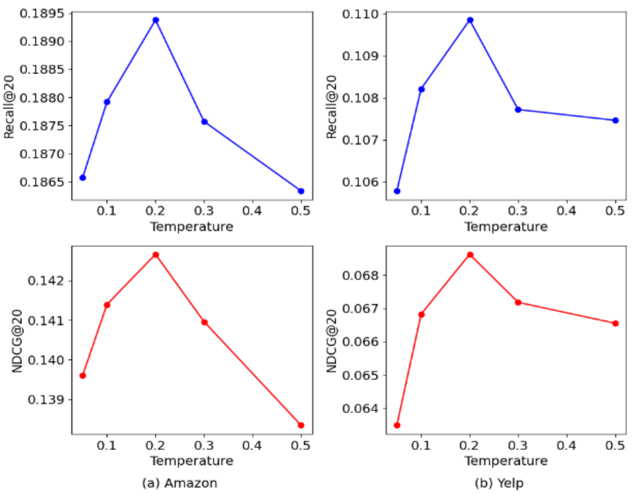

Fig. 5. Performance Comparison of different τ.

VI. Conclusion

We propose a recommendation framework, AFVCL, which integrates adaptive fusion, multi-view contrastive learning, and data augmentation to address the challenges of heterogeneous information modeling and data sparsity in recommender systems. By introducing an adaptive fusion strategy between the primary user-item view and the collaborative neighbor view, our model dynamically adjusts the weights of information from different views, effectively capturing high-order relationships. Meanwhile, we construct heterogeneous subgraphs via meta-paths to facilitate comprehensive modeling of user preferences and item attributes, enhancing the semantic representation of embeddings. Additionally, our data augmentation module employs node and edge dropout to generate diverse features within the primary user-item view, significantly improving the model's robustness and generalization ability. We also introduce a dual contrastive learning scheme to align semantic-level and interaction-level representations across views, further improving representation consistency. The meta-paths efficiently capture complex semantic relationships, while the synergy of adaptive fusion and data augmentation modules further boosts the model's performance. Our experimental results demonstrate that AFVCL consistently outperforms existing contrastive learning-based recommendation methods across multiple public datasets, validating its effectiveness.

Our future work will focus on further exploration of adaptive fusion strategies to better leverage multi-view information for improving recommendation performance. Another promising direction is to optimize data augmentation techniques and investigate how different types of feature perturbations can enhance the model's generalization ability.

References

[1] J. Wu, J. Chen, J. Wu, et al., "BSL: Understanding and improving softmax loss for recommendation," in Proc. 2024 IEEE 40th Int. Conf. Data Eng. (ICDE), 2024, pp. 816–830.
[2] S. Khusro, Z. Ali, and I. Ullah, "Recommender systems: Issues, challenges, and research opportunities," in Information Science and Applications (ICISA) 2016. Singapore: Springer, 2016, pp. 1179–1189.
[3] J. B. Schafer, J. A. Konstan, and J. Riedl, "E-commerce recommendation applications," Data Mining and Knowledge Discovery, vol. 5, pp. 115–153, 2001.
[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol. 42, no. 8, pp. 30–37, 2009.
[5] X. He, L. Liao, H. Zhang, et al., "Neural collaborative filtering," in Proc. 26th Int. Conf. World Wide Web, 2017, pp. 173–182.
[6] X. Wang, X. He, and T. S. Chua, "Learning and reasoning on graph for recommendation," in Proc. 13th Int. Conf. Web Search Data Min., 2020, pp. 890–893.
[7] C. Gao, X. Wang, X. He, et al., "Graph neural networks for recommender system," in Proc. 15th ACM Int. Conf. Web Search Data Min., 2022, pp. 1623–1625.
[8] J. Yu, H. Yin, J. Li, *et al.*, "Self-supervised multi-channel hypergraph convolutional network for social recommendation," in *Proc. Web Conf. 2021*. New York: ACM, 2021, pp. 413–424.
[9] Y. Chen, Z. Liu, J. Li, *et al.*, "Intent contrastive learning for sequential recommendation," in *Proc. ACM Web Conf. 2022*. New York: ACM, 2022, pp. 2172–2182.
[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint*, arXiv:1609.02907, 2016.
[11] S. Zhang, H. Tong, J. Xu, *et al.*, "Graph convolutional networks: A comprehensive review," *Comput. Social Netw.*, vol. 6, no. 1, pp. 1–23, 2019.
[12] M. Schlichtkrull, T. N. Kipf, P. Bloem, et al., "Modeling relational data with graph convolutional networks," in Proc. ESWC 2018, Springer, 2018, pp. 593–607.
[13] W. Fan, Y. Ma, Q. Li, et al., "Graph neural networks for social recommendation," in Proc. The World Wide Web Conf., 2019, pp. 417–426.

[14] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in Advances in Neural Information Processing Systems, vol. 30, 2017.

[15] L. Wang, W. Zhou, L. Liu, et al., "Deep adaptive collaborative graph neural network for social recommendation," Expert Syst. Appl., vol. 229, p. 120410, 2023.

[16] Y. Wang, W. Sun, J. Jin, et al., "MVGCN: Multi-view graph convolutional neural network for surface defect identification using three-dimensional point cloud," J. Manuf. Sci. Eng., vol. 145, no. 3, p. 031004, 2023.

[17] Y. Zhang, L. Sang, and Y. Zhang, "Exploring the individuality and collectivity of intents behind interactions for graph collaborative filtering," in Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2024, pp. 1253–1262.

[18] Y. Wei, X. Wang, Q. Li, et al., "Contrastive learning for cold-start recommendation," in Proc. 29th ACM Int. Conf. Multimedia, 2021, pp. 5382–5390.

[19] T. Chen, S. Kornblith, M. Norouzi, et al., "A simple framework for contrastive learning of visual representations," in Proc. Int. Conf. Machine Learning (ICML), PMLR, 2020, pp. 1597–1607.

[20] J. Yu, H. Yin, X. a, et al., "Self-supervised learning for recommender systems: A survey," IEEE Trans. Knowl. Data Eng., vol. 36, no. 1, pp. 335–355, 2023.

[21] J. Wu, X. Wang, F. Feng, et al., "Self-supervised graph learning for recommendation," in Proc. 44th Int. ACM SIGIR Conf. Research and Development in Information Retrieval, ACM, 2021, pp. 726–735.

[22] Y. Yang, C. Huang, L. Xia, et al., "Knowledge graph contrastive learning for recommendation," in Proc. 45th Int. ACM SIGIR Conf. Research and Development in Information Retrieval, ACM, 2022.

[23] X. Wang, N. Liu, H. Han, and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning," in Proc. ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining (KDD), 2021, pp. 1726–1736.

[24] X. He, K. Deng, X. Wang, et al., "LightGCN: simplifying and powering graph convolution network for recommendation," in Proc. ACM, 2020, doi: 10.1145/3397271.3401063.

[25] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang, and M. Zhang, "DisenHan: disentangled heterogeneous graph attention network for recommendation," in Proc. 29th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2020, pp. 1605–1614.

[26] X. Ren, L. Xia, J. Zhao, D. Yin, and C. Huang, "Disentangled contrastive collaborative filtering," in Proc. 46th Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 2023, pp. 1137–1146.

[27] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in Proc. 28th Int. World Wide Web Conf. (WWW), 2019, pp. 2022–2032.

[28] X. Long, C. Huang, Y. Xu, H. Xu, P. Dai, L. Xia, and L. Bo, "Social recommendation with self-supervised metagraph InfoMax network," in Proc. 30th ACM Int. Conf. Inf. Knowl. Manage. (CIKM), 2021, pp. 1160–1169.

[29] L. Sang, Y. Wang, Y. Zhang, et al., "Intent-guided heterogeneous graph contrastive learning for recommendation," arXiv:2407.17234, 2024.