HEAL: Hybrid Enhancement with LLM-based Agents for Text-attributed Hypergraph Self-supervised Representation Learning

Ruochang Li[♡], **Xiao Luo**[♠ †], **Zhiping Xiao**[◇ †], **Wei Ju**[♣], **Ming Zhang**[♣ †]
[♡] Henan University [♠] University of Wisconsin–Madison
[◇] University of Washington [♣] State Key Laboratory for Multimedia Information Processing, School of Computer Science, PKU-Anker LLM Lab, Peking University
ruochangli@gmail.com, xiao.luo@wisc.edu, patxiao@uw.edu, {juwei,mzhang_cs}@pku.edu.cn

Abstract

This paper studies the problem of text-attributed hypergraph self-supervised representation learning, which aims to generate discriminative representations of hypergraphs without any annotations for downstream tasks. However, real-world hypergraphs could contain incomplete signals, which could deteriorate the representation learning procedure, especially under label scarcity. Towards this end, we introduce a new perspective that leverages large language models to enhance hypergraph self-supervised learning and propose a novel data-centric approach named Hybrid Hypergraph Enhancement with LLM-based Agents (HEAL). The core of our HEAL is to generate informative nodes and hyperedges through multi-round interaction with LLM-based agents. In particular, we first retrieve similar samples for each node to facilitate the node expansion agent for different views. To generate challenging samples, we measure the gradients for each augmented view and select the most informative one using an evaluation agent. From the structural view, we adopt a topology refinement agent to incorporate new hyperedges for the recovery of missing structural signals. The enhanced hypergraphs would be incorporated into a self-supervised learning framework for discriminative representations. Extensive experiments on several datasets validate the effectiveness of our HEAL in comparison with extensive baselines.

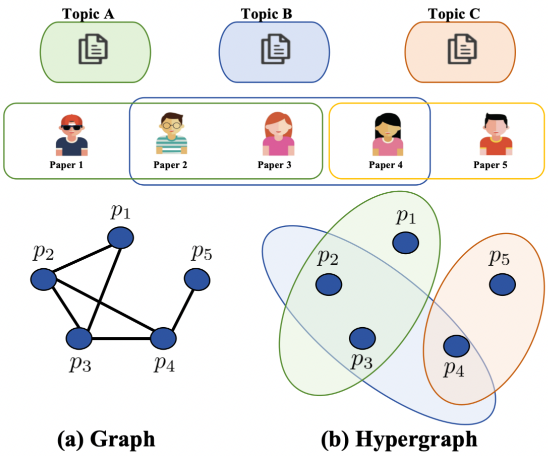


Figure 1: An illustration of modeling higher-order relationships among papers based on topic via text-attributed graph and hypergraph.

1 Introduction

In recent years, Large Language Models (LLMs) have raised a surge of interest in natural language processing (NLP) (Qin et al., 2024; Tahir et al., 2024; Vatsal and Dubey, 2024), and have demonstrated remarkable capabilities in multiple tasks. Due to extensive pre-training, LLMs could serve as a powerful tool to capture comprehensive linguistic patterns and real-world knowledge. Subseqently, LLMs have promoted the development of several domains, such as text generation, information extraction, and semantic understanding, through few-shot and zero-shot learning. In the relation mining field, these advantages enable LLMs to excel at automatically identifying and extracting implicit relationships between entities from unstructured texts (Xu et al., 2024; Schilling-Wilhelmi et al., 2024). Unlike traditional rule-based and machine learning approaches, LLMs have the capability of comprehensively combining in-context information with real-world knowledge to infer latent relationships, significantly enhancing accuracy and robustness. Consequently, researchers widely employ LLMs for relation mining, facilitating downstream tasks such as recommender systems and financial analysis.

Hypergraphs, as an effective data structure, are ubiquitous across various domains, including social network analysis, recommender systems, and bioinformatics (Bretto, 2013). Unlike traditional

[†] Corresponding authors.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 6815–6829
November 4-9, 2025

graphs, hypergraphs are characterized by representing complex higher-order dependencies and asymmetric relationships among nodes. With the prevalence of graph neural networks (GNNs) on graph data processing, hypergraph neural networks (HyperGNNs) (Feng et al., 2019) have emerged as an essential focus for hypergraph representation learning. HyperGNNs could effectively learn node representations through propagating information in the topological structure. Especially, in the self-supervised learning field, hypergraph contrastive learning (HyperGCL) (Wei et al., 2022) has been proven to be a powerful framework via constructing contrastive views to maximize the similarity of homogeneous nodes and minimize the distance between heterogeneous nodes' representations.

Despite the remarkable achievements of existing hypergraph self-supervised representation learning, they exhibit notable limitations when employed to process hypergraphs with textual description, often referred to as Text-Attributed Hypergraphs (TAHGs) (Bazaga et al., 2024; Nakajima and Uno, 2024), as depicted in Figure 1. **Firstly, it matters to balance the relationship between textual information and hypergraph topological structure.** Hypergraphs are often applied to model complicated high-order relationships, whereas textual data typically exhibits sequential or hierarchical structures. Due to different data structures, simply utilizing textual information to derive hyperedges is prone to destroying the original hypergraph structure, resulting in a counterproductive situation. **Secondly, augmenting nodes' text properly constitutes another crucial barrier.** Although LLMs excel at text generation, they are prone to hallucination (Huang et al., 2025), generating plausible yet nonfactual content. These shortcomings would directly impact the quality of constructed hyperedges. Typically, when it comes to excessively large hypergraphs, text processing for each node would incur overwhelming time consumption and computational costs. **Thirdly, how to construct optimal hyperedges remains a difficult challenge.** Given the particularly severe hyperedge sparsity problem, recent works have started to investigate effective frameworks to tackle it, such as k-HyperEdge (Li et al., 2024) and CASH (Ko et al., 2025). However, all of these approaches, without exception, focus solely on initial embeddings to derive better hyperedges. Although they may reasonably perform well, their effectiveness is inherently deteriorated by the failure to harness available textual resources.

To address these challenges, in this paper, we propose HEAL, a novel LLM-based hybrid hypergraph enhancement framework for self-unsupervised representation learning on hypergraphs. Unlike existing approaches, such as KGC with LLMs that complete predefined pairwise relations, our HEAL framework discovers higher-order structures in open-world settings. It also goes beyond local textual enhancement in LLM-GNNs by reconstructing global graph topology in a semantic-aware manner, enabling both fine-grained node understanding and holistic structural modeling. To address the first challenge, we provide LLMs with connectivity-based signals of the target node to improve hyperedge prediction. For large hypergraphs, we adopt a two-stage sampling strategy: nodes are first grouped by label, then high-degree nodes are prioritized. For the second challenge, we use KNN to retrieve similar texts for the central node, enriching its textual input with semantically aligned yet diverse augmentations. To select the most informative one, we apply a GCN with contrastive loss to guide the LLM via gradient-based evaluation. To tackle the last challenge, we design a topology refinement agent that recovers hyperedges with minimal structural noise. Instead of modeling full hyperedge sets, the LLM predicts pairwise connectivity between the current node and selected candidates.

Overall, our main contributions are as follows:

❶ We introduce HEAL, a novel hybrid enhancement framework for text-attributed hypergraph, leveraging LLMs to advance self-supervised representation learning, which bridges two important areas in NLP and hypergraph learning.

❷ We propose an effective two-stage approach to sample nodes from an excessively large hypergraph to perturb hyperedges at minimal cost.

❸ We present a pioneering approach to recover potential hyperedges from optimizing pairwise edges by jointly investigating textual and structural spaces.

❹ We carry out extensive experiments, showing that HEAL outperforms all baselines, including ED-HNN, HyperGT, and HyperGCL, with up to 4.87% improvement on ED-HNN.

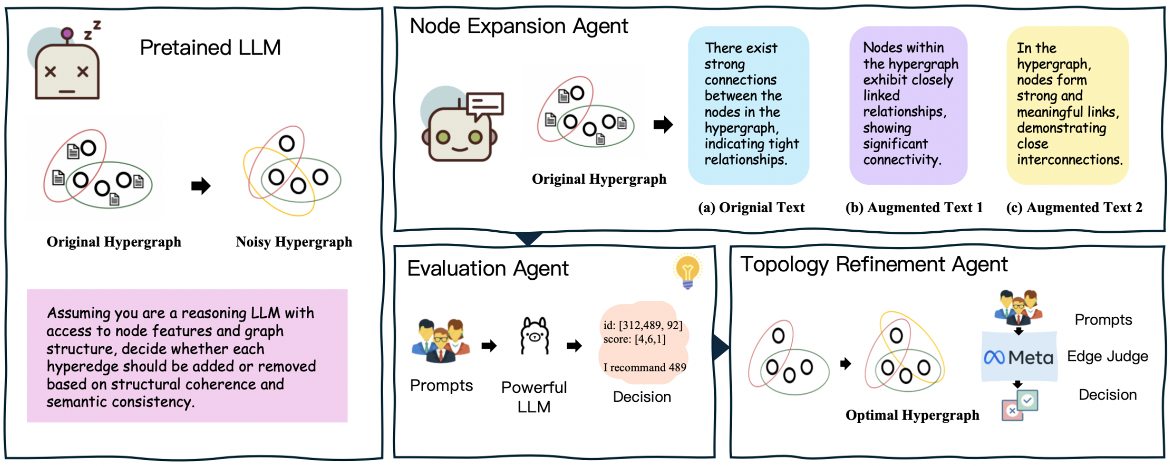


Figure 2: **High-level pipeline of our proposed HEAL.**This pipeline comprises two major processes: text augmentation and edge augmentation. Initially, we employ a BERT-based LLM as a text encoder to provide informative node texts for augmentation preprocessing. **(1) Text Augmentation:** For more precise text augmentation with LLM, we utilize k-nearest neighbors (KNN) to retrieve n similar texts as supplementary knowledge to alleviate the hallucination issue. For each node, to evaluate n augmented texts generated by LLM, we employ a graph convolutional network (GCN) to generate n+1 corresponding gradient messages(including one original text and m augmented ones). **(2) Edge Augmentation:** In order to refine hyperedges simply yet effectively, we leverage LLM to judge the connectivity of pairwise edges first, then reconstruct an optimized hypergraph.

2 Related Work

2.1 Large Language Models for Relation Mining

Recent efforts in refining graph structure have focused on relation mining (Fu et al., 2025), followed by specific downstream tasks such as node classification and link prediction. For instance, ConvE (Dettmers et al., 2018), KBGAN (Cai and Wang, 2018), and ConvTransE (Shang et al., 2018) utilize the initial embeddings to improve the performance of recovering potential relations. Nevertheless, they did not investigate the very original data; instead, they proposed novel frameworks to derive more information from existing embeddings. Some efforts, such as Graphedit (Guo et al., 2024), GraphGPT (Tang et al., 2024), and LLMRec (Wei et al., 2024), have shown that Large Language Model could alleviate these issues by effectively aligning textual and structural signals. Inspired by these findings, we first propose to leverage the potential relation mining capability of LLM to enhance the refinement of hypergraph structure.

2.2 Hypergraph Self-supervised Representation Learning

In contrast with unsupervised learning and semi-supervised learning, self-supervised learning is capable of effectively learn high-quality representations by automatically generating supervisory signals from the data itself. With the success of self-supervised representation learning on graphs, such as GraphCL (You et al., 2021) and GraphMAE (Hou et al., 2022), recent attention has shifted to apply these ideas on hypergraphs to solve more complicated real-world problems. For better capturing the higher-order information, ED-HNN (Wang et al., 2022) combines star expansions of hypergraphs with standard message passing neural networks. HyperGCL (Wei et al., 2022) proposes a variational hypergraph auto-encoder architecture with adversarial attacks for robustness and fairness. However, these methods struggle to uncover latent interactions among nodes from existing hyperedges, which poses inherent limitations of the quality of hypergraph topology. On the contrary, our proposed method tackle these issues at minimal cost yet effectively.

3 The Proposed HEAL

3.1 Framework Overview

The HEAL works, as illustrated in Figure 2, through multi-round interaction with LLM-based agents to generate informative nodes and the topological structure of a hypergraph to enhance self-supervised representation learning. In order to reduce the time complexity of processing excessively large hypergraphs, we only investigate the sampled nodes, which are determined by their classes and degrees. Given the hallucination, consistency, and diversity issues of LLM text augmentation, we first use k-nearest neighbors (KNN) to retrieve the top-n most similar texts for each original text. These texts would assist LLM in diversifying the original text while maintaining core meanings and distinguishing features. Due to the open-source nature and reliability of llama2 (Touvron et al., 2023a), we adopt llama2-7b as the backbone of our proposed framework to ensure confidence in its judgment. After these preparations, the llama serves to augment the original text by leveraging these similar texts, generating n augmented versions. Then, we employ a graph contrastive learning model on both the original text and the augmented text n, resulting in corresponding gradients n + 1. Following this, Llama selects the optimal augmented text based on both the textual and gradient information and determines whether pairwise edges should be added or removed. Finally, each node and its neighbors are used to form new hyperedges, thereby facilitating improved hypergraph contrastive learning.

3.2 Retrieval-augmented Node Expansion Agent for Multiple Views

A text-attributed hypergraph is formally denoted as $H = (V, E, T)$. In particular, V represents a non-empty set of nodes. $T \in \mathbb{T}$ represents text attribute of nodes, with t_i corresponding to the text attribute of v_i. E represents a set of hyperedges, which stand for higher-order relations among $v_i \in V$. As discussed above, previous researches focus solely on how to uncover latent relations among nodes through existing embeddings or higher-order interactions. The major issue is neglecting rich text attributes of nodes. For the purpose of harnessing textual resources and attaining optimal text augmentation, we propose an LLM-based agent, namely the Retrieval-augmented Node Expansion Agent, for multiple views of the original text of each vertex.

The first barrier is how to map texts into a d-dimensional embedding space via $f : \mathbb{T} \rightarrow \mathbb{R}^D$ effectively and discriminatively. Motivated by BERT (Devlin et al., 2019), we select different BERT-based LLMs, such as SciBERT (Beltagy et al., 2019) and BioBERT (Lee et al., 2019), which have been fine-tuned on specific domains. Next, in order to ensure consistency and diversity of multiple augmented texts, we leverage the core idea of retriever-augmented generation (RAG), which sophisticatedly overcomes the shortcomings of relying solely on pre-trained knowledge in our work (Gao et al., 2024). We retrieve the top n similar texts of nodes (except the processing node itself) as input for the LLM-based node expansion agent.

$$LLM_{\text{input}} = prompt(T, T_{\text{similar}}). \tag{1}$$

Subsequently, we would derive n different augmented texts for each vertex.

$$T_{\text{augment}} = LLM(prompt(T, T_{\text{similar}})). \tag{2}$$

3.3 Gradient-based Evaluation Agent for Informative Views

Up to this point, we have already discussed how to augment multiple textual views with an LLM-based node expansion agent. Now we will investigate how to select the best one for each vertex to facilitate hyperedge recovery. Specifically, we deploy graph contrastive learning (GCL) to generate gradient signals (Feng et al., 2022). Each original text and corresponding augmented texts would first be constructed to a small graph $G_i = (V_i, E_i, T_i)$ with i denoting the i-th node in hypergraph, $v_{ij} \in V_i$ denoting the j-th node in G_i, E_i denoting edges and $t_{ij} \in T_i$ denoting the text attribute of v_{ij}. The edges between these entities would be decided based on the cosine similarity of their embeddings.

Here, we aim to form positive and negative sample pairs between the original view and each augmented view, with classes determined by adaptive hyperparameters α and β. α represents the 90th percentile of the text similarity, serving as a threshold for sample pairs classification. β represents the value of the margin defining the minimum distance between negative sample pairs. For graph contrastive learning of G_i, negative contrastive loss is formulated as follows:

$$L_{\text{neg}} = \sum_{j=1}^{N_{\text{neg}}} \left(\max\left(0, \beta - \text{distance}(t_j, t_m)\right)\right)^2. \tag{3}$$

Given the limited input context window of LLM, we extract four representative statistical metrics, including mean, variance, max, and min of gradients. Then we concatenate each augmented text and associated gradients to facilitate the evaluation process for the most challenging and informative one.

Dataset	Nodes	Hyperedges	Features	Classes	h_e	Avg. Length
Cora	2,708	1,579	1,433	7	3.03	955.54
PubMed	19,717	7,963	500	3	4.35	1,649.25
Cora-CA	2,708	1,072	1,433	7	4.28	955.54
Zoo	101	43	16	7	39.93	103.54

Table 1: Dataset statistics of text-attributed hypergraphs (TAHGs). h_e denotes the average degree of hyperedges, and *Avg. Length* represents the average text length of all nodes.

3.4 Topology Refinement Agent for Hyperedge Recovery

In this work, our goal is to uncover latent hyperedges at minimal cost yet effectively. Assuming there are n nodes in a hypergraph, the number of possible combinations could be $2^n - 1$ (excluding the empty set). Due to the time consumption and complexity of iteratively validating each combination, we ingeniously accomplish this objective by refining pairwise edges and utilizing retrieval methods, formally introduced as a topology refinement agent.

For the purpose of refining pairwise edges through denoising noisy edges while identifying latent dependencies, we first explore the most cosine-equivalent non-neighbor nodes as candidates for edge adding and the most distinct neighbor nodes as candidates for edge deleting (Fang et al., 2024). So far, we have shifted the goal of recovering hyperedges to refining pairwise edges. Subsequently, we limit the refining process between each node and selected candidates. On the basis of both candidates and optimal augmented text attributes, we leverage the reasoning capability of LLM to judge the possibility of edge perturbation. Once refining the original topological structure of pairwise edges, we reconstruct hyperedges by gathering all nodes and connected neighbors. Typically, for those large-scale hypergraphs, the attention is restricted to the identified nodes discussed in the framework overview section. In succession, new hyperedges would be modified by means of comparing the degree with the average hyperedge degree in the original hyperedges. The reason is that hyperedges with excessive connectivity often contain information overlap and data noise, resulting in anomalous outcomes. Ultimately, adhering to the principle of preserving the original topological architecture of the hypergraph, we integrate the newly constructed edges with the existing ones.

3.5 Summarization

Considering the currently widespread limitation of data sparsity in hypergraph topological structure, we innovatively present a pioneering methodology to enhance hypergraph representation learning by combining hybrid LLM-based agents. To further elaborate, the methodology conducted by three distinct agents consists of two augmentation steps for text and structure, respectively. The former procedure utilizes a retrieval-augmented node expansion agent to yield diverse feature-level views and a gradient-based evaluation agent to opt for the most suitable one. The latter process incorporates the outcomes from prior work to achieve our aim of optimizing hyperedges with the assistance of a topology refinement agent. The advanced hypergraph would be used to elevate the performance of hypergraph representation learning.

4 Experiments

4.1 Settings

Datasets. We evaluate our proposed HEAL framework on four publicly available benchmark TAHGs datasets, including Cora, Pubmed, Cora-CA, and Zoo, with data statistics depicted in Table 1. Cora and Pubmed are cocitation networks adapted from HyperGCN (Yadati et al., 2019). The coauthorship networks Cora-CA are downloaded from HyperGCN (Yadati et al., 2019). Additionally, we also validate the effectiveness of our method on the Zoo dataset from UCI Categorical Machine Learning Repository (Kelly et al.). When it comes to the principle of data splitting, we consciously decide to follow the recommendation from previous works (Wei et al., 2022). Our efforts focus on how to leverage textual messages to advance the performance of hypergraph representation learning. Ultimately, we report the mean and standard deviation of the test accuracy averaged over multiple runs.

Baselines. We compare our methodology with HyperGCL, the state-of-the-art hypergraph representation learning framework in the self-supervised

Category	Method	Cora		PubMed		Cora-CA		Zoo	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1
Baselines	HCCF	68.89±1.80	68.81±1.74	84.56±0.34	84.42±0.39	73.22±1.65	72.89±1.74	61.05±14.54	60.86±12.96
	DHCN	68.24±1.12	68.19±1.12	84.38±0.38	84.37±0.35	72.74±1.53	72.53±1.25	57.35±18.32	57.38±14.54
	ED-HNN	80.61±1.14	80.50±1.09	89.13±0.48	89.12±0.48	83.63±1.01	83.42±0.93	88.46±4.80	85.17±6.73
	HyperGT	75.05±0.88	74.93±0.84	85.00±0.46	84.98±0.46	78.36±1.35	78.44±1.44	93.46±6.80	93.19±9.51
	HyperGCL	73.12±1.48	73.03±1.57	85.72±0.38	85.68±0.36	76.21±1.26	76.16±1.49	66.89±12.44	66.78±13.37
HEAL	HCCF	69.74±1.75	69.70±1.63	85.70±0.35	85.59±0.38	74.49±1.45	73.76±1.58	63.21±14.77	63.14±13.84
	DHCN	69.03±1.32	68.99±1.32	85.56±0.35	85.40±0.35	73.90±1.53	73.61±1.52	58.86±11.83	58.76±12.71
	ED-HNN	80.86±1.46	80.78±1.57	89.81±0.52	89.79±0.51	85.17±1.61	85.10±1.64	90.38±5.51	90.04±5.88
	HyperGT	76.62±1.27	76.41±1.32	86.95±0.39	86.95±0.38	77.58±0.78	77.36±1.02	93.46±6.29	92.61±8.38
	HyperGCL	74.32±1.46	74.15±1.04	86.99±0.38	86.82±0.41	77.60±1.21	77.49±1.34	69.30±11.88	68.92±14.48

Table 2: Performance comparison (in %) across four benchmark graph classification datasets, with standard deviations calculated over ten runs.

domain. To further test the strong generalizability of enhanced datasets, we also measure their impact on ED-HNN, HyperGT, and the other two baselines. (1) Hypergraph Contrastive Learning (HyperGCL) (Wei et al., 2022) focuses on effectively constructing contrastive views for hypergraphs through fabricated and generative augmentations. The fabricated method works by perturbing edges in an equivalent bipartite graph converted from hypergraphs. The generative technique involves a variational hypergraph auto-encoder architecture to avoid capturing redundant information during representation learning. (2) Equivariant Diffusion Hypergraph Neural Network (ED-HNN) is a hypergraph neural network architecture designed to efficiently model complex higher-order relations via equivariant hypergraph diffusion operators(Wang et al., 2022). It leverages star expansion and standard message passing to improve scalability and expressiveness. (3) HyperGT ingeniously devises a novel approach, namely HyperGraph Transformer, to integrate positional encodings from the incidence matrix and structural regularization to enhance representation learning(Liu et al., 2024). (4) Hypergraph contrastive collaborative filtering (HCCF) (Xia et al., 2022) represents a new self-supervised recommendation framework leveraging self-supervised learning to reinforce the representation quality of recommender systems through jointly capturing local and global collaborative correlations with cross-view contrastive learning supported by a hypergraph. (5) DHCN ingeniously devises a novel approach, namely Dual Channel Hypergraph Convolutional Networks (Xia et al., 2021), to comprehensively learn hypergraph topological structure from both the original hypergraph and the matching line graph.

Implementation Details. We conduct our main experiment on 2 NVIDIA GeForce RTX 4090 GPUs, adopting vllm to employ llama2-7b. To fairly evaluate the performance of new hypergraphs on baseline models, we adhere to the experiment environment settings of their regulations, reporting the mean evaluation results over 10 runs. Additionally, we retrieve and augment 2 texts for PubMed, given the excessive node numbers, while 3 for the other datasets. Ultimately, for the purpose of maintaining the original topological structure of hypergraphs, we prioritize uncovered hypergraphs of high-degree nodes to advance the original data.

Evaluation Metrics. Node classification, often adopted as a downstream task to evaluate the quality of graph self-supervised contrastive learning (Liu et al., 2022). Given this fact, we measure the effectiveness of our HEAL on the basis of the mean and standard deviation accuracy over multi-round epochs. Typically, the accuracy refers to the proportion of truly predicted node labels in the test set, and the F1-score refers to the harmonic mean of precision and recall, which is a balanced evaluation metric, especially under class imbalance.

4.2 Performance Comparison

In this section, we focus on analyzing the effect of our approach on previous representation learning models. As is illustrated in Table **??**, the outstanding results of ample experiments further demonstrate the superior performance of our framework. Consequently, we could draw three conclusions as follows:

- Hypergraph recovery presents potentially essential importance in the self-supervised domain. As evidenced in Table **??**, we could apparently acknowledge that by properly implementing un-

	Correlations			Results			
	a	b	c	Cora	PubMed	Cora-CA	Zoo
M_1		√	√	$73.67 \pm 1.53_{\downarrow 0.65}$	$86.52 \pm 0.35_{\downarrow 0.47}$	$76.76 \pm 1.64_{\downarrow 0.84}$	$64.13 \pm 12.29_{\downarrow 5.17}$
M_2	√		√	$73.87 \pm 1.40_{\downarrow 0.45}$	$86.23 \pm 0.29_{\downarrow 0.76}$	$76.90 \pm 2.10_{\downarrow 0.7}$	$67.16 \pm 12.65_{\downarrow 2.14}$
M_3	√	√		$72.24 \pm 1.48_{\downarrow 2.08}$	$85.79 \pm 0.37_{\downarrow 1.2}$	$75.81 \pm 1.56_{\downarrow 1.79}$	$64.25 \pm 11.83_{\downarrow 5.05}$
M_4	√	√	√	$\mathbf{74.32 \pm 1.46}$	$\mathbf{86.99 \pm 0.38}$	$\mathbf{77.60 \pm 1.21}$	$\mathbf{69.30 \pm 11.88}$

Table 3: Ablation analysis on HyperGCL of three main components of HEAL: (a) Retrieval-augmented Node Expansion Agent, (b) Gradient-based Evaluation Agent, and (c) Topology Refinement Agent.

covered hyperedges could notably advance the learning process of higher-order relationships, with up to 2.41% improvement for HyperGCL, 4.87% for ED-HNN, and 1.97% for HyperGT. These findings demonstrate the contribution of hypergraph recovery for better embedding formats in downstream tasks.

- Gradient-based information could assist LLM in recognizing the most challenging and informative augmented text for better hypergraph contrastive learning. We could infer that the gradient derived from the corresponding agent could lead to a more comprehensive understanding of different augmented texts for LLM. It can be attributed to a novel analytical perspective from the dynamic moving behavior of each augmented text for improved embedding, instead of being limited to LLM only.

4.3 Ablation Study

As illustrated in Table 3, we measure the contribution of each agent in our proposed HEAL on the most representative self-supervised model, HyperGCL, to explain the necessity of these techniques. Typically, the replacements for each module are as follows. ***Firstly***, for the purpose of assessing a retrieval-augmented node expansion agent, we re-conduct our hypergraph uncovering process directly through the original text attributes. As shown in the results, the accuracy declines by 0.47% and 5.17%, respectively. We could notice that the impact is more noteworthy on the zoo dataset. The underlying cause of this phenomenon is that the original node description about the zoo is in a one-hot manner. To utilize the feature attributes, we design diverse descriptions for each feature, then generate the original text descriptions. Therefore, if we don't employ LLM to augment the text information based on the specific characteristics of each animal, the discrimination would be weaker. ***Secondly***, when it comes to a gradient-based evaluation agent, we replace the procedure of identifying the optimal text with random selection. Instead, we

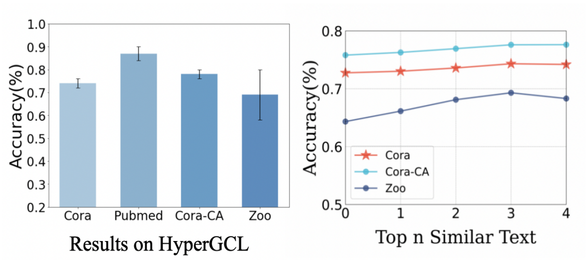


Figure 3: The right-hand figure is the bar chart of results on HyperGCL. The left-hand figure refers to the sensitivity analysis of our proposed HEAL.

measure the effectiveness of augmented text by providing LLM with the textual information solely. As shown in Table 3, the accuracy drops to 86.23% from 86.99% for PubMed. Meanwhile, there is also a similar shrinkage by 2.14% when applied to the Zoo dataset. Based on this evidence, the significance of the valuation agent is validated statistically. ***Thirdly***, as for the topology refinement agent, we could observe that both Pubmed and Zoo achieve their lowest point, with 1.20% decline for Pubmed and 5.05% decline for Zoo. Intuitively, we could come to the conclusion that the refinement agent is the most indispensable component. In particular, we develop the ablation study without the hyperedge verification process, which has already been discussed in the methodology part. This further validates the fact that excessively high-degree nodes and augmented data might instead lead to detrimental effects.

4.4 Sensitivity Analysis

In this section, we assess the impact of the hyperparameter n on the performance of HEAL to optimize the model's configuration and improve its interpretability. In particular, n refers to the number of supplemented materials in the text augmentation stage, illustrated in Figure **??**. As shown in Figure 3, the accuracy of Cora and Cora-CA is more stable compared to Zoo in response to the change n. We could infer that the reason lies in the original text attributes for nodes, which means nodes in Zoo

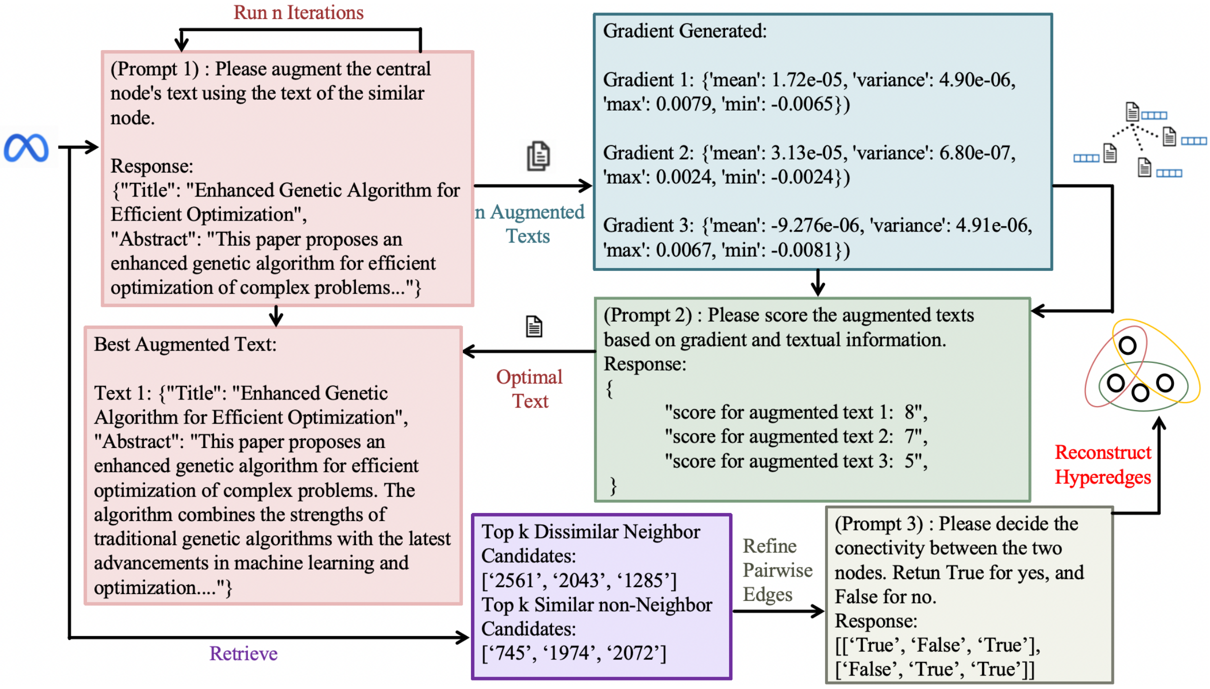


Figure 4: Case study for understanding the function and effectiveness of our proposed HEAL. Typically, gradient i belongs to the i-th augmented text. The gradient for the original text is discarded in this illustration.

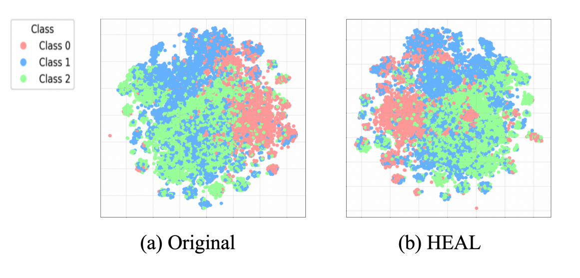


Figure 5: Impact of our proposed HEAL: the analysis of outcomes with T-SNE visualization on the PubMed dataset based on HyperGCL.

datasets could benefit more from supplementation from LLM enhancement. Typically, on Cora and Zoo, the performance achieves the peaks at n = 3, with 74.32% and 69.30%, respectively. While there is still an increase from n = 3 to n = 4 for Cora-CA, the fluctuation is minimal.

4.5 Case Study

Explainability. We meticulously conduct a case study, illustrated in Figure 4, to assist in a deeper understanding of our work on the proposed HEAL. As a prerequisite for the text augmentation stage, we first derive top n = 3 similar texts as input for prompt 1, which would run n iterative interactions with the llama to generate n versions of different texts. Then we utilize GCL to generate corresponding gradients for these n + 1 texts, resulting in n + 1 gradients. Subsequently, we employ a llama to score these texts from gradient and textual viewpoints. In the edge augmentation stage, we initially retrieve top k = 3 neighbor candidates for edge addition and edge deletion, respectively. Ultimately, Llama would judge the connectivity for the improved topological structure of the hypergraph.

t-SNE Visualization. Moreover, we conduct t-SNE visualization in Figure 5. From the results, LLM could boost the text encoder process through a proper prompt manner. As illustrated in Figure 5, we visualize the enhanced text embedding with t-SNE. It clearly shows that the superior discriminative and aggregative characteristics of the right picture, which is processed carefully by leveraging the text generation capability of LLM. While the picture on the left-hand side is shadowly encoded in one-hot format.

5 Conclusion

In this work, we propose a novel framework, HEAL, a hybrid LLM-based agents to boost hypergraph representation learning in a self-supervised domain. By leveraging the text generation capability and pre-trained knowledge of LLM, we ingeniously refine the topological structure of hypergraphs at minimal cost. In particular, we introduce three specific agents for textual-level augmentation and structural-level refinement, namely the retrieval-augmented node expansion agent, the gradient-based evaluation agent, and the topology refinement agent. Our evaluation demonstrates that the refined hypergraph structure could advance hypergraph representation learning effectively. These findings indicate the potential of LLM-based agents to tackle severe data sparsity issue in hypergraphs.

6 Limitations

Although our work proposes a novel framework to tackle the hypergraph refinement problem, it still reveals several limitations. Firstly, our hybrid agents rely on LLM to conduct enhancement, which means the outcomes would be restricted by the quality of LLM. Moreover, as LLM is treated as a black box, the explainability is weaker. Finally, our work utilizes only llama2-7b due to the computational constraints. In the future, we aim to explore the performance of various large language models to seek the upper bound of our method.

7 Acknowledgment

Ming Zhang is supported by grants from the National Key Research and Development Program of China with Grant No. 2023YFC3341203 and the National Natural Science Foundation of China (NSFC Grant Number 62276002). The authors are grateful to the anonymous reviewers for their criticism- cally, reading this article and for giving important suggestions to improve this article.

References

Adrián Bazaga, Pietro Liò, and Gos Micklem. 2024. Hyperbert: Mixing hypergraph-aware layers with language models for node classification on text-attributed hypergraphs. *Preprint*, arXiv:2402.07309.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. *Preprint*, arXiv:1903.10676.

Alain Bretto. 2013. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 1.

Liwei Cai and William Yang Wang. 2018. Kbgan: Adversarial learning for knowledge graph embeddings. *Preprint*, arXiv:1711.04071.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Fang Zeng, Wei Liu, et al. 2025. Auggpt: Leveraging chatgpt for text data augmentation. *IEEE Transactions on Big Data*.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. *Preprint*, arXiv:1707.01476.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. 2024. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 747–758.

Shengyu Feng, Baoyu Jing, Yada Zhu, and Hanghang Tong. 2022. Adversarial graph contrastive learning with information regularization. In *Proceedings of the ACM web conference 2022*, pages 1362–1371.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. *Preprint*, arXiv:1809.09401.

Yiwei Fu, Yuxing Zhang, Chunchun Chen, JianwenMa JianwenMa, Quan Yuan, Rong-Cheng Tu, Xinli Huang, Wei Ye, Xiao Luo, and Minghua Deng. 2025. Mark: Multi-agent collaboration with ranking guidance for text-attributed graph clustering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6057–6072.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

Zirui Guo, Lianghao Xia, Yanhua Yu, Yuling Wang, Zixuan Yang, Wei Wei, Liang Pang, Tat-Seng Chua, and Chao Huang. 2024. Graphedit: Large language models for graph structure learning. *Preprint*, arXiv:2402.15183.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. 2022. Graphmae: Self-supervised masked graph autoencoders. *Preprint*, arXiv:2205.10803.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository.

Yunyong Ko, Hanghang Tong, and Sang-Wook Kim. 2025. Enhancing hyperedge prediction with context-aware self-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, page 1–13.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Feijiang Li, Jieting Wang, Liuya zhang, Yuhua Qian, Shuai jin, Tao Yan, and Liang Du. 2024. k-hyperedge medoids for clustering ensemble. *Preprint*, arXiv:2412.08289.

Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and Philip Yu. 2022. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, page 1–1.

Zexi Liu, Bohan Tang, Ziyuan Ye, Xiaowen Dong, Siheng Chen, and Yanfeng Wang. 2024. Hypergraph transformer for semi-supervised classification. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7515–7519. IEEE.

Kazuki Nakajima and Takeaki Uno. 2024. Inferring community structure in attributed hypergraphs using stochastic block models. *Preprint*, arXiv:2401.00688.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S. Yu. 2024. Large language models meet nlp: A survey. *Preprint*, arXiv:2405.12819.

Mara Schilling-Wilhelmi, Martiño Ríos-García, Sherjeel Shabih, María Victoria Gil, Santiago Miret, Christoph T. Koch, José A. Márquez, and Kevin Maik Jablonka. 2024. From text to insight: Large language models for materials science data extraction. *Preprint*, arXiv:2407.16867.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2018. End-to-end structure-aware convolutional networks for knowledge base completion. *Preprint*, arXiv:1811.04441.

Munief Hassan Tahir, Sana Shams, Layba Fiaz, Farah Adeeba, and Sarmad Hussain. 2024. Benchmarking the performance of pre-trained llms across urdu nlp tasks. *Preprint*, arXiv:2405.15453.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. Graphgpt: Graph instruction tuning for large language models. *Preprint*, arXiv:2310.13023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023a. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Shubham Vatsal and Harsh Dubey. 2024. A survey of prompt engineering methods in large language models for different nlp tasks. *Preprint*, arXiv:2407.12994.

Peihao Wang, Shenghao Yang, Yunyu Liu, Zhangyang Wang, and Pan Li. 2022. Equivariant hypergraph diffusion neural operators. *arXiv preprint arXiv:2207.06680*.

Tianxin Wei, Yuning You, Tianlong Chen, Yang Shen, Jingrui He, and Zhangyang Wang. 2022. Augmentations in hypergraph contrastive learning: Fabricated and generative. *Preprint*, arXiv:2210.03801.

Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Llmrec: Large language models with graph augmentation for recommendation. *Preprint*, arXiv:2311.00423.

Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 70–79. ACM.

Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4503–4511.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Preprint*, arXiv:2312.17617.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. 2019. Hypergcn: A new method of training graph convolutional networks on hypergraphs. *Preprint*, arXiv:1809.02589.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2021. Graph contrastive learning with augmentations. *Preprint*, arXiv:2010.13902.

A Time Complexity

In this part, we provide the time complexity analysis of our proposed HEAL framework. The overall computational cost consists of two main phases:

1. Data Preprocessing: We conduct KNN to compute the similarity among M nodes. The time complexity of this step is:

$$T_{\mathrm{KNN}} = O(M^2) \tag{4}$$

since pairwise similarity computations are required between nodes.

2. LLM Inference: Let N be the total number of queries sent to the language model. Assuming a batch size of B, we perform $\lceil \frac{N}{B} \rceil$ batches of inference. For each batch, the language model processes approximately L_{batch} tokens, with an inference complexity of $O(L_{\text{batch}}^d)$ per batch, where d is an architecture-dependent exponent. Thus, the total time complexity is:

$$T_{\mathrm{LLM}} = \left\lceil \frac{N}{B} \right\rceil \cdot O(L_{\text{batch}}^d) \tag{5}$$

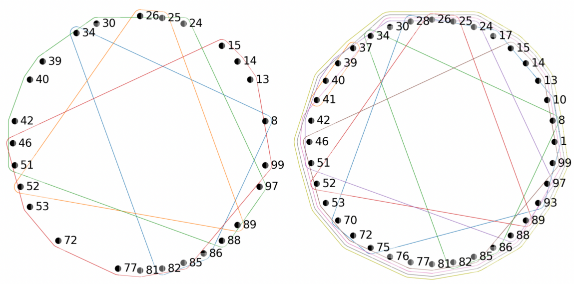
where $B = 20$, $N \in O(10^3)$, $L_{\text{batch}} \in O(10^2)$, and $d = 2$, with all notations summarized as: B—batch parallel calls, N—total queries, L_{batch}—tokens per batch, d—LLM inference complexity exponent (Touvron et al., 2023b; Vaswani et al., 2017).

B Analysis of LLMs

The effectiveness of our HEAL is inherently dependent on the quality of the underlying LLMs. For example, the diversity and faithfulness of the augmented texts used to refine hypergraph structure are directly tied to the generative capacity of the LLM. Similarly, the quality of embeddings used for KNN-based neighbor retrieval is also model-sensitive. Recent studies, such as AugGPT(Dai et al., 2025) have shown that stronger LLMs such as GPT-4 and Claude consistently produce more coherent, semantically rich outputs across a variety of generation tasks, suggesting that our framework may yield even greater improvements when used with such models.

C Hypergraph Structure Comparision

D Prompt Templates



(a) Original Hyperedges (b) Enhanced Hyperedges

Figure 6: Visualization of the original and enhanced hypergraph structure (only demonstrating the most significant hyperedges for simplicity).

Prompts for Retrieval-augmented Node Expansion Agent:

System: You are a knowledgeable assistant specializing in hypergraph contrastive learning and text augmentation. Your task is to enhance the central node's description by utilizing information from a neighbor node while maintaining the central node's key distinguishing features and original meaning. Ensure that the augmented content enriches the central node without changing its core essence, and supports finding potential neighboring nodes or removing noise edges for hypergraph learning tasks.

User: Please revise the central node's text using the text of a neighbor node.

Ensure that the revised central node's content enhances the representation of the central node without losing its key distinguishing features or original meaning. The goal is to expand and enrich the content of the central node by incorporating relevant insights from the neighbor node while maintaining the original intent and core message of the central node. The augmented content should help in: Identifying potential neighboring nodes and removing noise edges by ensuring the central node's representation is rich and accurate.

Central Node text: {central_node_abstract}

Neighbor Node text: {similar_node_abstract}

Return the answer in the following JSON format:

```
{{
   "Title": "Replace with the revised title of the central node",
   "Abstract": "Replace with the revised abstract of the central node"
}}
"""
```

Figure 7: Prompt for Retrieval-augmented Node Expansion Agent

Prompts for Gradient-based Evaluation Agent:

System: You are an knowledgeable assistant specializing in using text and corresponding gradient to evaluate the effectiveness of augmented texts in the context of hypergraph contrastive learning.

User: Here is the original text and two augmented texts with their corresponding gradient information, including mean, variance, min, and max.

Please score each of these two augmented texts on a scale from 0 to 10 based on their **effectiveness for hypergraph contrastive learning in order to improve the accuracy for node classification. Use both the text and the gradient information to determine the scores.

A higher mean and lower variance indicate that the gradient is stronger and more stable, meaning this gradient is better"

Each augmented texts has just one score."

Please just score the two augmented texts; the original text is for comparison only.

Return the answer in the following JSON format with:

```
{{
    "score for augmented text 1:  <score1>",
    "score for augmented text 2:  <score2>"
}}
"""
```

Figure 8: Prompt for Gradient-based Evaluation Agent

Add Edge:

System: "You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe."

User: "Here are the attributes of the two nodes:"
"Central Node {i}: {node_text}"
"Nonneighbor Node {candidate}: {candidate_text}"
"Your task is to decide whether there should be an edge between these two nodes for better hypergraph contrastive learning."
"Consider the following when making your decision:"
"1. The textual similarity between the central node and the neighbor node."
"2. Whether connecting these nodes would help improve the learning task, such as enhancing the contrastive representation."
"Please return 'True' if an edge should be added (i.e., they are similar and should be connected), or 'False' if no edge should be added."
"Only respond with 'True' or 'False'. Do not include any other explanation or text. For example: True"

Figure 9: Prompt for Topology Refinement Agent

Delete Edge:

System: "You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being safe."

User: "Here are the attributes of the two nodes:"
"Central Node {i}: {node_text}"
"Neighbor Node {candidate}: {candidate_text}"
"Your task is to decide whether there should be an edge between these two nodes for better hypergraph contrastive learning."
"Consider the following when making your decision:"
"1. The textual similarity between the central node and the neighbor node."
"2. Whether connecting these nodes would help improve the learning task, such as enhancing the contrastive representation."
"Please return 'True' if an edge should be removed (i.e., they are dissimilar and shouldn't be connected), or 'False'if no edge should be removed."
"Only respond with 'True' or 'False'. Do not include any other explanation or text. For example: True"

Figure 10: Prompt for Topology Refinement Agent