ADHERE: ADAPTIVE DENOISING FOR ROBUST HYPERGRAPH CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

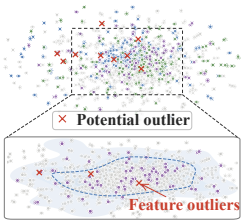
Hypergraph clustering aims to discover cluster structures by jointly exploiting node features and high-order relationships encoded by hyperedges. However, real-world hypergraphs inevitably contain feature and structure noise, which can propagate throughout the clustering process by degrading input reliability, distorting learned representations, and introducing unstable optimization signals. To address these issues, we propose an adaptive denoising-aware joint optimization framework for hypergraph clustering that tackles noise from input, representation, and optimization perspectives. Specifically, we design a differentiable denoising mechanism based on the reparameterization trick to adaptively select reliable node features and hyperedge incidences, thus generating two complementary denoised views. Then we perform cross-view contrastive learning in a high-dimensional space to learn consistent representations, while introducing pseudo-label-based self-supervised learning in a low-dimensional label space to explicitly reinforce cluster-level semantic structures. Furthermore, we adopt a parameter-pruning gradient update strategy to suppress unstable optimization signals during training. Extensive experiments on multiple benchmark datasets demonstrate the effectiveness and robustness of the proposed framework across several clustering metrics.

1 INTRODUCTION

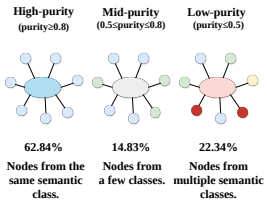
Hypergraphs are important graph structures in which hyperedges can connect multiple nodes simultaneously (Bretto, 2013; Ju et al., 2024b; Yi et al., 2025). This property enables hypergraphs to effectively capture high-order relationships among data objects, making them more suitable for modeling complex non-independent and identically distributed data in real-world scenarios, and thus they are widely used in social networks, citation networks, recommender systems, and biological information networks (Yang et al., 2020; Song et al., 2024; Guo et al., 2024; Tao et al., 2023). In recent years, node-level hypergraph clustering gains increasing attention, which aims to partition nodes in a hypergraph into several semantically consistent and structurally compact clusters without relying on manual labels. The key challenge lies in jointly exploiting node feature and structure information encoded by hyperedges to learn node representations with strong cluster discriminability.

Existing hypergraph clustering methods can be broadly divided into traditional methods and deep learning-based methods. Traditional methods usually include spectral partitioning-based methods (Zhou et al., 2006), matrix factorization-based methods (Zeng et al., 2014), and probabilistic model-based methods (Chodrow et al., 2021). These methods have solid theoretical foundations, but their expressive ability is limited when dealing with complex feature information and nonlinear high-order relationships. With the development of graph neural networks (GNN) (Ju et al., 2024a), deep hypergraph clustering methods attract increasing attention. Some methods employ hypergraph neural networks to pass messages between nodes and hyperedges, thereby capturing high-order structural dependencies (Song et al., 2024). Other methods introduce contrastive learning mechanisms, where different hypergraph views are constructed and the consistency of the same node across different views is maximized to improve the robustness and discriminability of node representations (Ni et al., 2026). In addition, some studies attempt to enhance the clustering-friendliness of hypergraph representations through structure alignment, hyperedge-level supervision, or cluster assignment constraints (Wang et al., 2025; Gu et al., 2026; Ni et al., 2026). Although these methods improve hypergraph clustering performance to some extent, an important challenge remains insufficiently addressed in face of complex real-world data.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Feature noise



(b) Structure noise

Figure 1: Motivation of ADHERE. (a) The t-SNE visualization highlights potential feature outliers that deviate from their class distributions in the original feature space but close to them after dimensionality reduction. (b) Hyperedges are grouped by label purity, defined as the proportion of nodes belonging to the dominant class within each hyperedge. High, mid, and low-purity hyperedges account for 62.84%, 14.83% and 22.34%.

Through experiments, we observe that real-world hypergraphs commonly contain both feature and structure noise. As shown in Figure 1, the raw feature space contains abnormal nodes that deviate from the main class distributions, suggesting noisy or clustering-irrelevant attributes, while a non-negligible proportion of hyperedges connect nodes from different semantic classes, suggesting potential noisy structures. Such noise can be further propagated through high-order message passing and affect the reliability of hypergraph clustering. However, designing a robust hypergraph clustering framework that can effectively handle such noise remains challenging due to several critical obstacles. **1) The lack of an adaptive denoising mechanism.** Noise often degrades reliability of input, i.e., the original hypergraph view, and existing hypergraph clustering methods often construct augmented views through random feature masking (Lee & Shin, 2023) or random hyperedge perturbation (Wei et al., 2022). These predefined perturbations treat different features and hyperedge incidences indiscriminately, making it difficult to distinguish informative content from unreliable information and thus fail to perform adaptive denoising. **2) The lack of cluster-level semantic guidance.** Noise from unreliable inputs can propagate into learned representations and distort the underlying cluster structure. However, existing methods mainly emphasize representation consistency (Wang et al., 2025), with limited cluster-level semantic guidance to mitigate such distortion. **3) The lack of a strategy for suppressing unstable optimization signals.** Noise can introduce unstable optimization signals during training, which may propagate through parameter updates and undermine the robustness of learned representations (Han et al., 2025). However, few existing methods explicitly explore effective strategies to suppress such unreliable signals during optimization.

To address the aforementioned challenges, we propose an **A**daptive **D**enoising-aware **HypER**graph clust**E**ring joint optimization framework (ADHERE). Unlike augmentation strategies that rely on random perturbations, our ADHERE starts from the perspective of reliable view generation. Specifically, we first design a differentiable hypergraph denoising module based on the reparameterization trick, which adaptively selects node features and structural information to generate two semantically reliable and complementary denoised views. The two denoised views are fed into a shared hypergraph encoder to obtain the corresponding node representations. Then we perform node-level contrastive learning in a high-dimensional representation space, enabling the model to learn consistent representations between views. Meanwhile, we construct a low-dimensional label space and perform multi-view clustering-guided self-supervised learning by using the pseudo-labels generated from clustering to learn cluster-discriminative representations. In addition, since unstable optimization signals can arise in the gradients during backpropagation, we introduce a parameter-pruning gradient update strategy that selectively updates important parameters, thereby reducing noisy fluctuations and improving optimization stability. Extensive experiments on multiple benchmark datasets demonstrate the effectiveness and robustness of our proposed ADHERE across several clustering metrics.

2 The Proposed Method

2.1 Notations and Problem Definition

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X, H)$ denote a hypergraph with node set $\mathcal{V} = \{v_i\}_{i=1}^n$ and hyperedge set $\mathcal{E} = \{e_j\}_{j=1}^m$. The node features are $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n\times d}$, where $x_i \in \mathbb{R}^d$, and the incidence matrix $H \in \{0,1\}^{n\times m}$ encodes the structure, with $H_{ij} = 1$ if $v_i \in e_j$. The node and hyperedge degree matrices are $D_v \in \mathbb{R}^{n\times n}$ and $D_e \in \mathbb{R}^{m\times m}$, where $D_v(v_i) = \sum_{j=1}^m H_{ij}$ and $D_e(e_j) = \sum_{i=1}^n H_{ij}$. Hypergraph clustering learns node representations $Z = f_\theta(\mathcal{G})$ without supervision and partitions $\mathcal{V}$ into K clusters $\mathcal{C} = \{C_k\}_{k=1}^K$ satisfying $C_i \cap C_j = \emptyset$ for $i \neq j$ and $\bigcup_{k=1}^K C_k = \mathcal{V}$.

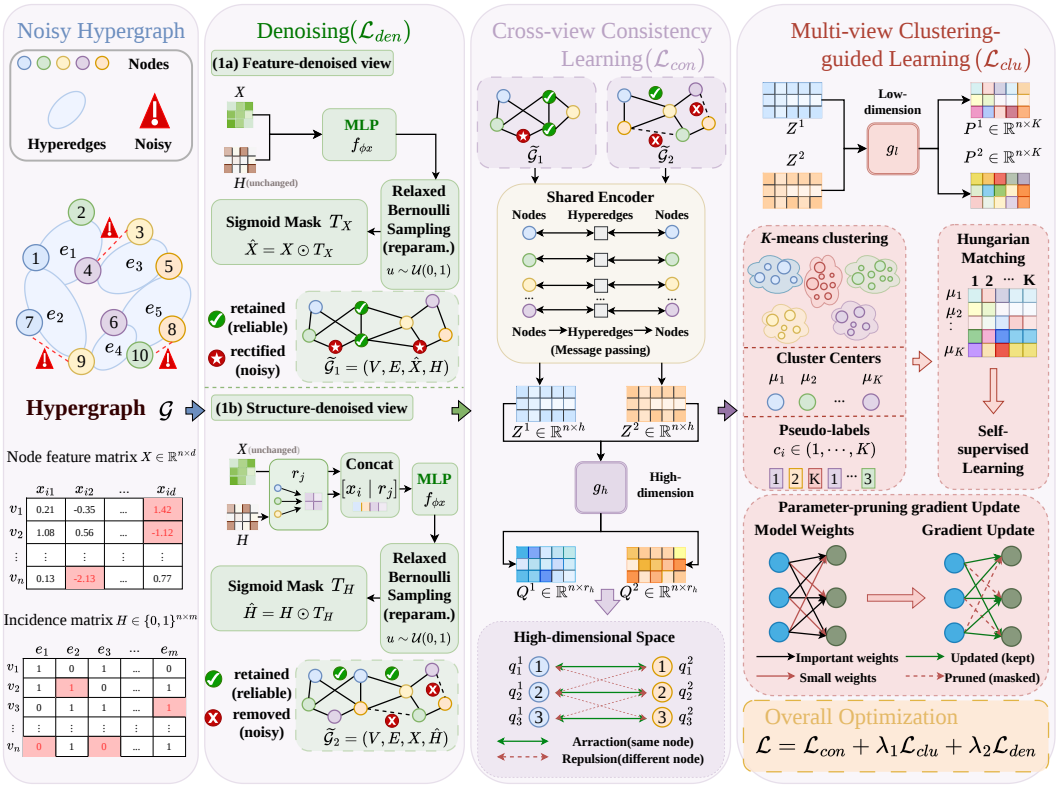


Figure 2: An overview of our proposed hypergraph clustering framework ADHERE, which consists of four components: (1) Adaptive denoising mechanism. (2) Cross-view consistency learning. (3) Multi-view clustering-guided learning. (4) Parameter-pruning gradient update.

2.2 Adaptive Denoising Mechanism

As discussed in Section 1, noises in features and structures are inevitable. Therefore, we design an adaptive denoising mechanism through a learnable stochastic masking method. Specifically, we first generate learnable parameters α for nodes or edges:

$$\alpha_{ij}^{X} = f_{\phi_X}(X_{ij}), \quad r_j = \frac{1}{D_e(e_j)} \sum_{i=1}^{n} H_{ij} x_i, \quad \alpha_{ij}^{H} = f_{\phi_H}([x_i \mid r_j]). \tag{1}$$

Then we model the mask t as a Bernoulli variable and use the reparameterization trick to generate a continuous relaxation of the mask:

$$\pi_{ij} = \sigma(\alpha_{ij}), t_{ij} \sim \text{Bernoulli}(\pi_{ij}) \rightarrow \hat{t}_{ij} = \sigma\left(\frac{\alpha_{ij} + \epsilon_{ij}}{\tau_d}\right), \tag{2}$$

where ϵ_{ij} is a random variable derived from a uniform distribution and τ_d is the temperature. Through this way, we approximate the discrete sampling process as a differentiable continuous function composed of the learnable parameter α_{ij} and the random variable ϵ_{ij}, thus enabling end-to-end optimization. Then we produce the denoised matrices and construct two complementary denoised views to obtain more reliable hypergraph information:

$$\hat{X} = X \odot T_X, \quad \widetilde{\mathcal{G}}^1 = (\mathcal{V}, \mathcal{E}, \hat{X}, H), \quad \hat{H} = H \odot T_H, \quad \widetilde{\mathcal{G}}^2 = (\mathcal{V}, \mathcal{E}, X, \hat{H}), \tag{3}$$

where $\odot$ denotes the Hadamard product and $T_X = [\hat{t}_{ij}^X]^{n \times d}, T_H = [\hat{t}_{ij}^H]^{n \times m}$. To avoid the trivial solution of preserving all structural and feature information, we introduce a denoising regularization term to encourage sparsity in the learned mask matrices, which drives the model to retain only compact and reliable information during adaptive denoising:

$$\mathcal{L}_{den} = \frac{1}{nd} \sum_{i=1}^{n} \sum_{j=1}^{d} \hat{t}_{ij}^X + \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} \hat{t}_{ij}^H. \tag{4}$$

2.3 CROSS-VIEW CONSISTENCY LEARNING

In order to align the consistency of the same node across the two views while distinguishing different nodes, we perform node-level contrastive learning in the representation space. First we employ HGNN (Feng et al., 2019b) as the base encoder to learn node representations of the two denoised views. For the s-th denoised view $\widetilde{\mathcal{G}}^s, s \in \{1, 2\}$, the normalized propagation matrix used in hypergraph convolution is defined as:

$$\Theta^s = (D_v^s)^{-\frac{1}{2}} H^s W_e (D_e^s)^{-1} (H^s)^\top (D_v^s)^{-\frac{1}{2}}. \tag{5}$$

Based on this matrix, we perform hypergraph convolution at the l-th layer:

$$Z_l^s = \rho\left(\Theta^s Z_{l-1}^s W_l\right), \quad l \in \{1, 2, \ldots, L\}, \tag{6}$$

where W_l is the learnable parameter of the l-th layer, and $\rho(\cdot)$ is a nonlinear activation function. Let $Z_0^s = X^s$, after stacking L propagation layers, we can get the final node representations:

$$Z^1 = f_\theta(\hat{X}, H), \quad Z^2 = f_\theta(X, \hat{H}). \tag{7}$$

Instead of directly applying the contrastive objective in the original hypergraph encoding space, we map the representations of the two views into a high-dimensional projection space and perform cross-view contrastive learning for the same node:

$$Q^s = g_h(Z^s) \in \mathbb{R}^{n \times r_h}, \quad \mathcal{L}_{con} = \frac{1}{n} \sum_{i=1}^{n} -\log \frac{\exp\left(\mathrm{sim}(q_i^1, q_i^2)/\tau_h\right)}{\sum_{j=1}^{n} \exp\left(\mathrm{sim}(q_i^1, q_j^2)/\tau_h\right)}, \tag{8}$$

where $g_h(\cdot)$ is the high-dimensional projection head, $q_i^s \in \mathbb{R}^{r_h}$ denotes the representation of node v_i in high-dimensional space, r_h denotes the dimension of the projection space, τ_h is the temperature coefficient for contrastive learning, and $\mathrm{sim}(\cdot)$ denotes cosine similarity.

2.4 MULTI-VIEW CLUSTERING-GUIDED LEARNING

Cross-view consistency learning mainly captures instance-level discrimination. However, clustering tasks emphasize cluster-level semantic structures, which are more difficult to preserve in the presence of noise, so we construct a label space and use clustering-derived pseudo-labels as self-supervision to jointly optimize representation learning and cluster assignment. Specifically, we first project original representations into the low-dimensional label space and generate the temporary pseudo-label l_i^s:

$$P^s = \mathrm{softmax}(g_l(Z^s)) \in \mathbb{R}^{n \times K}, \quad l_i^s = \arg\max_k p_{ik}^s, \tag{9}$$

where $g_l(\cdot)$ denotes the projection head and p_i^s is the representation of node v_i in the label space. We apply k-means clustering in the original representations to obtain K cluster centers $\{\mu_1^s, \mu_2^s, \ldots, \mu_K^s\}$ and get the clustering pseudo-label $c_i^s = \arg\min_k \|z_i^s - \mu_k^s\|_2^2$ of node v_i based on its relation to these centers. Then we employ the Hungarian algorithm (Kuhn, 1955) to align the clustering pseudo-labels c_i^s with the pseudo-labels in the low-dimensional space l_i^s:

$$\Pi^s = \arg\max_{\Pi} \sum_{a=1}^{K} \sum_{b=1}^{K} \Pi_{ab} A_{ab}^s, \quad s.t. \sum_{a=1}^{K} \Pi_{ab} = 1, \quad \sum_{b=1}^{K} \Pi_{ab} = 1, \quad \Pi_{ab} \in \{0, 1\}, \tag{10}$$

where $A_{ab}^s = \sum_{i=1}^{n} \mathbf{1}(l_i^s = a)\mathbf{1}(c_i^s = b)$ denotes the cost matrix. With the corrected clustering pseudo-labels $\hat{y}_i^s$ for the two views, we then perform intra-view clustering-guided learning, encouraging the label space distribution of each view to approach its corresponding clustering result, thus obtain a consistency between the label space and the clustering structure:

$$\hat{y}_i^s = a, \quad \text{if } \Pi_{a,c_i^s}^s = 1, \quad \mathcal{L}_{intra} = -\frac{1}{2n} \sum_{s=1}^{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{y}_{ik}^s \log p_{ik}^s. \tag{11}$$

Furthermore, since these two views are derived from the same original nodes, their clustering semantics should remain consistent. To enhance clustering consistency across multiple views, we introduce cross-view contrastive learning in the low-dimensional label space:

$$\mathcal{L}_{cross} = \frac{1}{n} \sum_{i=1}^{n} -\log \frac{\exp\left(\mathrm{sim}(p_i^1, p_i^2)/\tau_h\right)}{\sum_{j=1}^{n} \exp\left(\mathrm{sim}(p_i^1, p_j^2)/\tau_h\right)}. \tag{12}$$

Finally, the multi-view clustering-guided learning loss in the low-dimensional space is defined as:

$$\mathcal{L}_{clu} = \mathcal{L}_{intra} + \mathcal{L}_{cross}. \tag{13}$$

2.5 Parameter-pruning Gradient Update

Optimizing the above losses can improve representation robustness; however, as discussed in Section 1, noise may still induce unstable gradient signals and compromise optimization stability. To mitigate this, we adopt a magnitude-aware parameter-pruning gradient update strategy, which selectively updates parameters with large magnitudes while masking the gradients of less important ones, thereby suppressing unstable optimization signals and improving training robustness. Specifically, let all learnable parameters of the model at training step t be $\Omega^{(t)} = \{\omega_1^{(t)}, \omega_2^{(t)}, \ldots, \omega_O^{(t)}\}$. For a parameter $\omega_o^{(t)}$, we measure its importance by the magnitude of its absolute value, i.e., $\Gamma(\omega_o^{(t)}) = |\omega_o^{(t)}|$. To dynamically distinguish important parameters from less important ones, we set a threshold based on the importance of all parameters. Let $\psi(\beta)$ denote the β-th percentile of all $\Gamma(\omega_o^{(t)})$. The set of important parameters is defined as:

$$\Omega_c^{(t)} = \{\omega_o^{(t)} \mid \Gamma(\omega_o^{(t)}) \geq \psi(\beta)\}, \tag{14}$$

where β controls the proportion of parameters regarded as important. Meanwhile, we define $\Omega_u^{(t)} = \Omega^{(t)} \setminus \Omega_c^{(t)}$ as the set of less important parameters. For the optimization objective $\mathcal{L}$, the standard gradient of $\omega_o^{(t)}$ can be written as $g_o^{(t)} = \nabla_{\omega_o^{(t)}} \mathcal{L}$. Based on the parameter importance defined above, we construct a parameter-level mask and prune the gradient to get the final parameter update rule:

$$M_{\Omega,o}^{(t)} = \mathbf{1}\left(|\omega_o^{(t)}| \geq \psi(\beta)\right), \quad \widetilde{g}_o^{(t)} = g_o^{(t)} \odot M_{\Omega,o}^{(t)}, \quad \omega_o^{(t+1)} = \omega_o^{(t)} - \eta \widetilde{g}_o^{(t)}, \tag{15}$$

where $\mathbf{1}(\cdot)$ is the indicator function and η is the learning rate. Unlike permanent pruning, our strategy retains all parameters but selectively updates them, this less aggressive strategy reduces the noisy gradients without sacrificing model capacity.

2.6 Overall Optimization

Based on the above modules, we formulate the overall joint optimization objective as:

$$\mathcal{L} = \mathcal{L}_{con} + \lambda_1 \mathcal{L}_{clu} + \lambda_2 \mathcal{L}_{den}, \tag{16}$$

where λ_1, λ_2 are parameters to balance losses, and the overall framework is illustrated in Figure 2. During optimization, we update the model parameters using the parameter-pruning gradient update strategy introduced in Section 2.5. We next provide a theoretical analysis of the adaptive denoising mechanism, with the proofs provided in the Appendix A.

3 Theoretical Analysis

3.1 Definitions

Definition 1 (Clean and denoised hypergraphs)**.** Let $\mathcal{G}^\star = (\mathcal{V}, \mathcal{E}, X^\star, H^\star)$ denote an underlying clean hypergraph, where $X^\star \in \mathbb{R}^{n\times d}$ and $H^\star \in \mathbb{R}^{n\times m}$ are feature and incidence matrices without noise. For $s \in \{1, 2\}$, let $\widetilde{\mathcal{G}}^s = (\mathcal{V}, \mathcal{E}, X^s, H^s)$ denote a denoised view. Following the hypergraph propagation rule in Section 2.3, the clean and denoised representations at the l-th layer are defined as:

$$Z_l^\star = \rho(\Theta^\star Z_{l-1}^\star W_l), \quad Z_l^s = \rho(\Theta^s Z_{l-1}^s W_l), \quad l = \{1, 2, \ldots, L\},$$

where $Z_0^\star = X^\star$, $Z_0^s = X^s$, W_l is the trainable weight matrix of the l-th layer.

3.2 Theorems

Theorem 1 (Topology-aware stability of denoised hypergraph propagation)**.** *Assume that $\rho(\cdot)$ is L_ρ-Lipschitz continuous, the clean and denoised propagation operators are spectrally bounded by κ, the layer-wise weight matrices are spectrally bounded by ν_l, and the clean hidden representations are bounded by R_l. Define $\delta_X^s = \|X^s - X^\star\|_F$ and $\delta_\Theta^s = \|\Theta^s - \Theta^\star\|_2$. Let $Z^\star \in \mathbb{R}^{n\times q}$ be the clean representation and $Z^s \in \mathbb{R}^{n\times q}$ be the denoised representation of view s produced by the hypergraph encoder. Then the representation perturbation at the L-th layer satisfies:*

$$\|Z_L^s - Z_L^\star\|_F \leq A_L \delta_X^s + B_L \delta_\Theta^s,$$

where $A_L = \prod_{t=1}^{L} a_t, B_L = \sum_{l=1}^{L} b_l \prod_{t=l+1}^{L} a_t$ with $a_l = L_\rho \kappa \nu_l$ and $b_l = L_\rho \nu_l R_{l-1}$.

Remark 1. Theorem 1 shows that the final hypergraph representation error is controlled by two terms: the feature perturbation δ_X^s and the topology-induced propagation perturbation δ_Θ^s. The feature-denoised view mainly reduces δ_X^1 and the structure-denoised view mainly reduces δ_Θ^2. Therefore, the two denoised views are complementary in reducing the upper bound of the representation error.

Theorem 2 (Clustering stability of denoised hypergraph embeddings)**.** *Assume that $Z^\star$ is K-clusterable with clean labels $c_i^\star$, clean centers $\mu_1^\star, \ldots, \mu_K^\star$, minimum center separation $\Delta > 0$, and clean within-cluster radius r. Let $\hat{M}^s$ be the center-assignment matrix obtained by a $(1+\xi)$-approximate K-means solution on Z^s, and define $\epsilon_Z^s = \|Z^s - Z^\star\|_F$. Then the set of potentially misclustered nodes $\mathcal{B}^s$ is bounded by:*

$$|\mathcal{B}^s| \leq \frac{4(1+\sqrt{1+\xi})^2}{\Delta^2}(\epsilon_Z^s + \sqrt{n}r)^2,$$

where $\mathcal{B}^s$ denotes the set of nodes whose assigned center is at least $\Delta/2$ away from its clean center. If additionally $|\mathcal{B}^s| < n_{\min}$, with $n_{\min} = \min_k |\{i : c_i^\star = k\}|$, then there exists a permutation of the K-means cluster indices such that the clustering error satisfies:

$$\mathrm{Err}(Z^s) \leq \frac{4(1+\sqrt{1+\xi})^2}{n\Delta^2}(\epsilon_Z^s + \sqrt{n}r)^2.$$

Combining this result with Theorem 1 gives

$$\mathrm{Err}(Z^s) \leq \frac{4(1+\sqrt{1+\xi})^2}{n\Delta^2}\left(A_L\delta_X^s + B_L\delta_\Theta^s + \sqrt{n}r\right)^2.$$

Remark 2. Theorem 2 shows that if the clean hypergraph representation has a separable cluster structure, then the clustering error of the denoised representation is controlled by the representation perturbation ϵ_Z^s. Together with Theorem 1, this implies that reducing the feature perturbation δ_X^s and the topology propagation perturbation δ_Θ^s can reduce the upper bound of the final clustering error.

4 Experiments

4.1 Datasets, baselines, and evaluation metrics

For datasets, we evaluate our proposed ADHERE on eight widely used public hypergraph datasets, including Cora-C, Citeseer, Pubmed (Sen et al., 2008), Cora-A, DBLP (Rossi & Ahmed, 2015), 20NewsW100, Mushroom (Asuncion et al., 2007; Zhou et al., 2006), and NTU2012 (Feng et al., 2019b). For baselines, we compare ADHERE with several representative graph and hypergraph representation learning or clustering methods: Node2vec (Grover & Leskovec, 2016), DGI (Veličković et al., 2018), RAGC (Zhao et al., 2025), Hyper2vec (Huang et al., 2019), TriCL (Lee & Shin, 2023), SE-HSSL (Li et al., 2024), HCN (Wang et al., 2025), and CAHC (Ni et al., 2026). For metrics, we evaluate clustering performance using four widely adopted metrics: clustering accuracy(ACC) (Xie et al., 2016), F1-score(F1) (Larsen & Aone, 1999), Normalized Mutual Information(NMI) (Strehl & Ghosh, 2002), and Adjusted Rand Index(ARI) (Hubert & Arabie, 1985). More details about the datasets, baselines, metrics, implementations and experiments are provided in the Appendix E.

4.2 Overall Performance Comparison

We compare our ADHERE with representative baselines on clustering tasks shown in Table 1. Overall, our ADHERE achieves the best or highly competitive results on most clustering tasks, showing stable effectiveness across different data scales and label structures. Compared with graph-based methods, our ADHERE generally performs better because it directly models high-order node-hyperedge-node relationships instead of converting the hypergraph into pairwise edges. Compared with hypergraph-based baselines, the advantage of our ADHERE mainly comes from adaptive denoising and clustering-guided representation learning. Existing contrastive methods often construct views through random perturbations, which may discard useful information or keep noisy associations. In contrast, our ADHERE adaptively denoises both node features and hypergraph structures, and further uses pseudo-label supervised learning in the low-dimensional label space to make the learned representations more suitable for clustering. Meanwhile, some baselines still obtain slightly better results on a few specific metrics, indicating that clustering performance is still affected by dataset-specific properties. Nevertheless, the overall results verify the robustness of our ADHERE and the effectiveness of preserving high-order structures while reducing feature and structural noise.

Table 1: Clustering performance on eight benchmark datasets. We report the results of four metrics with standard deviations (%), while **bold** the best and underline the second best.

Dataset	Metric	Graph Methods			Hypergraph Methods					
		Node2vec	DGI	RAGC	Hyper2vec	TriCL	SE-HSSL	HCN	CAHC	ADHERE(ours)
Cora-C	ACC	47.4±5.4	68.4±3.4	68.6±1.6	59.7±5.1	70.0±1.1	72.0±3.5	63.65±2.21	71.4±1.7	**75.17±0.25**
	F1	44.7±7.1	62.0±3.0	64.8±1.6	58.5±0.4	65.1±1.8	67.0±4.3	57.42±3.01	69.1±2.8	**74.03±0.98**
	NMI	32.5±3.9	54.0±1.0	50.9±1.7	40.6±1.1	55.2±0.8	**56.5±2.4**	43.11±2.84	56.4±0.9	55.51±1.18
	ARI	20.2±4.2	47.1±2.0	43.5±2.2	33.6±0.9	49.2±0.2	49.3±3.5	36.96±2.27	50.8±1.6	**52.35±2.23**
Citeseer	ACC	36.2±1.6	63.8±4.3	69.4±1.4	42.7±1.3	67.6±1.5	68.2±1.7	63.17±2.28	69.8±0.8	**69.97±0.76**
	F1	29.2±3.1	55.7±3.4	61.9±1.9	41.1±1.2	62.1±2.0	62.5±1.5	57.35±1.14	**65.1±0.6**	65.03±1.88
	NMI	20.0±1.5	40.0±2.1	43.5±0.6	21.0±1.0	44.1±0.4	43.8±2.1	35.89±3.24	**45.9±1.2**	44.99±1.14
	ARI	5.7±0.6	38.6±2.8	44.3±1.7	16.9±1.1	45.1±0.3	43.3±2.0	37.00±1.16	46.2±1.2	**46.26±1.8**
Pubmed	ACC	53.8±0.5	63.9±8.0	67.0±2.5	69.4±5.4	62.4±0.6	65.8±2.3	69.87±2.4	69.4±0.2	**73.84±1.91**
	F1	54.6±0.5	61.8±10.6	65.4±2.2	67.7±5.0	61.3±0.6	65.1±2.4	69.61±2.27	69.0±0.2	**72.87±1.86**
	NMI	21.6±0.4	27.0±6.4	27.1±2.4	29.6±3.3	29.7±0.6	31.2±1.1	32.16±1.98	34.4±0.4	**37.23±2.11**
	ARI	12.1±0.5	24.9±8.0	26.6±3.1	33.5±6.1	24.7±0.8	27.5±2.2	30.85±2.26	32.2±0.3	**39.22±3.37**
Cora-A	ACC	34.0±2.0	60.9±5.0	59.7±1.1	22.7±0.2	70.2±1.5	61.0±3.9	62.62±3.46	**72.6±0.4**	71.26±1.67
	F1	24.4±2.2	53.5±6.4	54.9±1.8	17.8±0.1	69.1±1.2	52.3±2.5	57.02±2.03	**70.6±0.4**	68.28±1.59
	NMI	12.5±1.2	44.1±1.6	42.2±0.3	3.0±0.5	50.2±1.2	47.6±1.9	46.84±2.31	**53.1±0.7**	48.95±1.63
	ARI	3.1±1.2	37.0±3.6	34.5±0.9	0.6±0.2	45.8±2.5	38.8±3.6	35.78±4.56	**50.1±0.7**	46.37±2.52
DBLP	ACC	39.91±3.3	OOM	OOM	OOM	65.9±4.1	OOT	OOM	72.3±8.0	**79.16±1.94**
	F1	38.05±3.8				63.7±5.3			70.4±8.6	**78.11±1.77**
	NMI	24.20±1.9				62.4±3.7			**63.9±2.8**	61.66±1.71
	ARI	9.63±1.3				50.4±5.2			58.2±5.9	**59.63±3.36**
20NewsW100	ACC	44.6±1.2	OOM	OOM	OOM	61.5±3.7	70.2±0.7	55.05±1.26	64.6±3.1	**74.78±0.95**
	F1	41.8±0.4				54.9±2.7	65.0±1.0	48.75±1.05	63.2±2.8	**72.92±1.61**
	NMI	11.1±0.7				36.0±3.8	39.1±1.4	29.88±2.25	39.5±1.1	**42.40±2.13**
	ARI	12.1±1.0				29.4±9.4	40.4±0.2	26.80±1.62	35.6±2.5	**47.70±1.58**
Mushroom	ACC	69.9±6.5	70.7±2.5	87.5±0.8	OOM	51.3±1.4	78.3±6.7	88.01±1.03	80.5±2.4	**88.55±1.05**
	F1	69.3±6.8	67.8±3.1	87.4±0.8		38.8±2.4	77.4±7.6	87.86±1.24	80.5±2.5	**88.53±1.06**
	NMI	15.0±6.8	20.9±4.8	46.6±2.8		3.4±0.6	28.7±10.5	**51.38±0.97**	30.3±4.2	50.52±2.88
	ARI	17.6±8.2	17.3±3.8	56.3±2.5		0.5±1.0	34.2±14.4	54.43±2.35	37.5±5.9	**59.49±3.26**
NTU2012	ACC	56.7±2.2	65.0±2.1	66.1±0.3	58.8±0.6	70.5±0.5	70.5±0.6	70.48±0.78	70.8±1.1	**71.44±0.66**
	F1	51.4±1.7	59.8±2.0	57.4±1.1	54.0±0.6	63.8±0.8	64.0±0.7	**64.13±0.52**	63.7±1.1	63.94±1.5
	NMI	77.3±0.4	80.0±0.6	79.9±0.6	78.1±0.1	83.1±0.2	83.2±0.3	82.22±1.03	**83.8±0.2**	83.4±0.33
	ARI	50.1±2.5	59.4±3.2	63.1±1.2	51.8±0.6	66.8±0.5	66.9±1.5	64.84±1.37	67.0±2.0	**67.91±1.45**

Table 2: Ablation study in terms of ACC across multiple datasets (%).

Ablation Variant	Cora-C	Citeseer	Pubmed	Cora-A	DBLP	20NewsW100	Mushroom	NTU2012
w/o $\mathcal{L}_{den}$	64.78	56.65	69.40	67.46	76.50	69.99	86.20	69.98
w/o $\widetilde{\mathcal{G}}^1$	71.69	58.29	72.16	65.49	78.33	70.81	84.88	68.73
w/o $\widetilde{\mathcal{G}}^2$	68.06	59.80	72.55	64.07	77.42	70.36	86.04	69.28
w/o $\mathcal{L}_{con}$	59.83	56.99	67.81	60.97	68.21	69.63	85.78	67.94
w/o $\mathcal{L}_{clu}$	74.26	65.91	71.53	69.80	74.40	73.16	84.50	69.18
ADHERE	**75.17**	**69.97**	**73.84**	**71.26**	**79.16**	**74.78**	**88.55**	**71.44**

4.3 Ablation Study

To assess the contribution of each component in ADHERE, we conduct an ablation study. As shown in Table 2, the full model achieves the highest ACC on almost all datasets, demonstrating the complementary effects of its components. Removing the denoising module (w/o $\mathcal{L}_{den}$) generally degrades performance, confirming the importance of suppressing feature and structural noise. Removing either $\widetilde{\mathcal{G}}^1$ or $\widetilde{\mathcal{G}}^2$ also reduces performance, indicating that the two views provide complementary information with dataset-dependent importance. Moreover, removing $\mathcal{L}_{con}$ weakens cross-view alignment, while removing $\mathcal{L}_{clu}$ eliminates direct clustering guidance. Overall, the ablation study demonstrates that adaptive denoising, contrastive learning, and clustering supervision work synergistically to enhance the robustness and discriminability of learned representations.

4.4 Effect of Denoised View Construction

To evaluate the effect of view construction, we compare three strategies: Denoised, using adaptive denoising; Original, using the raw input view; and Random, using random feature and structure

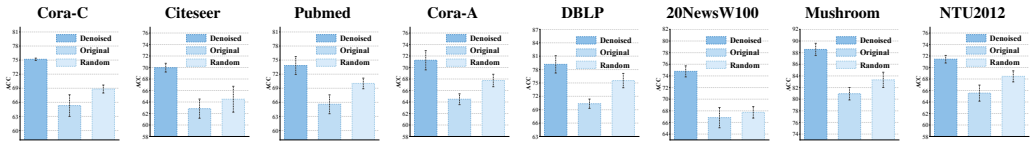


Figure 3: Comparison of three view construction strategies on eight datasets about ACC.

perturbations. As shown in Figure 3, Denoised consistently outperforms the other two strategies across eight datasets and multiple metrics. Compared with Original, it reduces the influence of redundant features and unreliable hyperedge incidences; compared with Random, it better preserves information that are relevant to clustering task. These results demonstrate the effectiveness of constructing high quality denoised views for subsequent contrastive and clustering-guided learning.

4.5 Analysis on Denoising Masks

To further evaluate adaptive denoising, we track the feature and structure masking ratios together with clustering accuracy. As shown in Figure 4, the left Y-axis shows the masking rates, the right Y-axis shows accuracy, and the X-axis denotes training epochs. On Cora-C, the structure masking ratio gradually increases and stabilizes at a relatively high level, while the feature masking ratio slightly decreases, suggesting that the model preserves more reliable high-order structural information while filtering redundant or noisy attributes. On Citeseer, both feature and structure masking ratios continuously decrease during training, indicating stronger suppression of unreliable information from both views. Meanwhile, the clustering accuracy on both datasets generally increases and gradually stabilizes despite several fluctuations during training. These results show that the learned masks dynamically adjust according to dataset characteristics and progressively retain more clustering-relevant information, further demonstrating the effectiveness of the adaptive denoising mechanism.

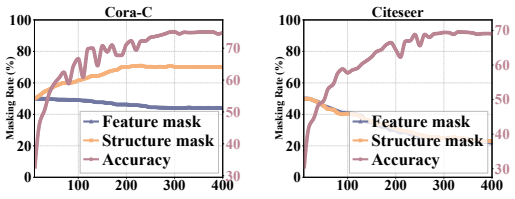


Figure 4: Illustration of the dynamic changes of the denoising masks during training.

4.6 Hyper-parameter Analysis

We analyze sensitivity to the pruning ratio β, which controls the proportion of masked parameter gradients. As shown in Figure 5, our proposed ADHERE remains stable across different ratios, with both datasets achieving the best or near-best performance at $\beta = 0.5$. Smaller ratios insufficiently suppress noisy gradients, whereas larger ratios prevent too many parameters from being updated and limit model adaptability. Therefore, $\beta = 0.5$ provides a favorable balance between training stability and model capacity and is used as the default setting.

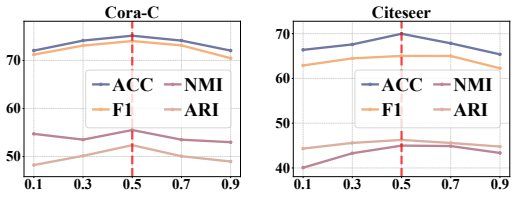


Figure 5: Impact of different values of β on the model. The best performance is highlighted by the vertical red dotted line.

4.7 Visualization of Node Embedding

To intuitively evaluate representation quality, we visualize the original features and learned embeddings using t-SNE. As shown in Figure 6, the original features exhibit severe class mixing on several datasets, indicating that the raw feature space is not naturally suitable for clustering. CAHC forms more compact local structures, but noticeable overlaps remain. In contrast, our proposed ADHERE produces clearer cluster structures and better class separability across most datasets, with more compact intra-class distributions and more distinguishable inter-class boundaries. These visualization results are consistent with the quantitative comparisons and further show that our ADHERE learns more robust and clustering-friendly node representations

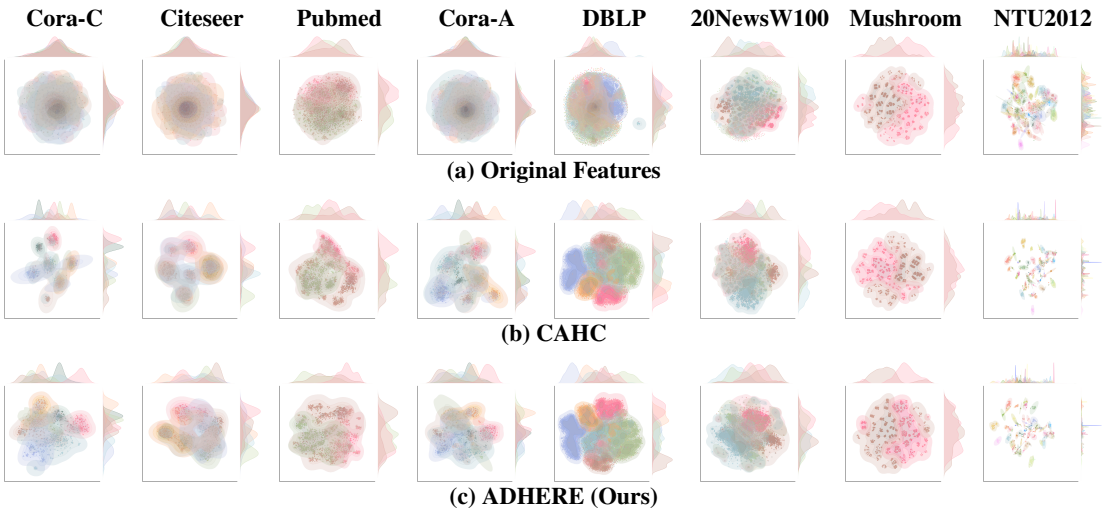


Figure 6: The result of t-SNE visualization of the original features and embeddings learned by CAHC and our ADHERE on eight datasets, each color in each subfigure represents a specific category.

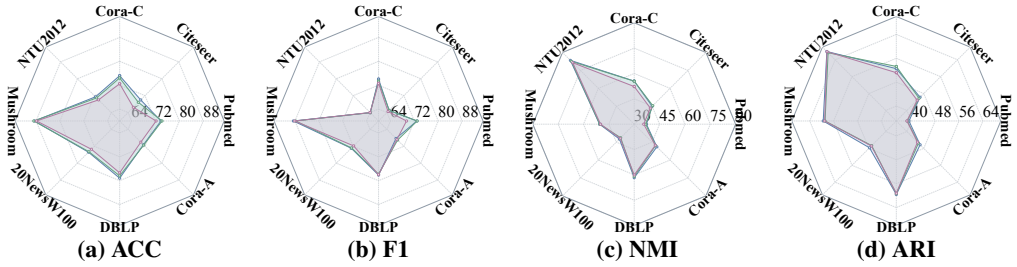


Figure 7: Backbone generalization comparison with different hypergraph encoders. Each radar chart reports one clustering metric across eight datasets, and the colored regions correspond to using HGNN, HNHN, and HyperGCN as the backbone encoder in ADHERE.

4.8 Analysis of Backbone Generalization

To explore the backbone generalization of ADHERE, we replace the HGNN encoder with HNHN Dong et al. (2020) and HyperGCN Yadati et al. (2019) while keeping the remaining framework unchanged. As shown in Figure 7, all three backbones achieve competitive results across eight datasets, indicating that our method is not tied to a specific encoder. Among them, HGNN usually gives the most stable results, especially on DBLP and NTU2012, suggesting that incidence-based hypergraph convolution is well matched with the denoised views generated by ADHERE. HNHN also performs well on several cases, such as Pubmed, Cora-A, and Mushroom, while HyperGCN remains competitive and achieves the best F1 on NTU2012. These differences are expected because the three encoders model high-order relations in different ways. Overall, the results verify that ADHERE can generalize to different backbones while preserving its main performance advantages.

5 Conclusion

In this paper, we propose an adaptive denoising-aware framework for robust hypergraph clustering under feature and structure noise. We construct denoised feature and structure views, align them through contrastive learning, and leverage pseudo-label self-supervised learning to learn clustering-oriented representations, while using gradient pruning strategy to improve optimization stability. In this way, the framework jointly enhances input reliability, representation discriminability, and training robustness. Theoretical analysis and extensive experiments on eight datasets further demonstrate its effectiveness, robustness, and generality. For the future work, we will explore finer-grained noise modeling and extend the framework to dynamic, heterogeneous, and large-scale hypergraphs.

AI USE STATEMENT

In this work, we used generative AI tools solely for language polishing of the manuscript and for no other purpose. All AI-polished content was reviewed and revised by the authors, and we take responsibility for the final content of this work.

REFERENCES

Arthur Asuncion, David Newman, et al. Uci machine learning repository, 2007.

Alain Bretto. Hypergraph theory. *An introduction. Mathematical Engineering. Cham: Springer*, 1: 209–216, 2013.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *International Conference on Learning Representations*, 2022.

Philip S Chodrow, Nate Veldt, and Austin R Benson. Generative hypergraph clustering: From blockmodels to modularity. *Science Advances*, 7(28):eabh1303, 2021.

Yihe Dong, Will Sawin, and Yoshua Bengio. Hnhn: Hypergraph networks with hyperedge neurons. *arXiv preprint arXiv:2006.12278*, 2020.

Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. Graph adversarial training: Dynamically regularizing based on graph structure. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2493–2504, 2019a.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3558–3565, 2019b.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. *Advances in neural information processing systems*, 32, 2019.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

Zhibin Gu et al. Hypergraph-enhanced contrastive learning for multi-view clustering with hyper-laplacian regularization. *Advances in Neural Information Processing Systems*, 38:54600–54620, 2026.

Liangmin Guo, Shiming Zhou, Haiyue Tang, Xiaoyao Zheng, and Yonglong Luo. Multi-behavior hypergraph contrastive learning for session-based recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 37(3):1325–1338, 2024.

Andi Han, Wei Huang, Zhanpeng Zhou, Gang Niu, Wuyang Chen, Junchi Yan, Akiko Takeda, and Taiji Suzuki. On the role of label noise in the feature learning process. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21733–21773. PMLR, 2025.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Youpeng Hu, Xunkai Li, Yujie Wang, Yixuan Wu, Yining Zhao, Chenggang Yan, Jian Yin, and Yue Gao. Adaptive hypergraph auto-encoder for relational data clustering. *IEEE Transactions on Knowledge and Data Engineering*, 35(3):2231–2242, 2021.

Jie Huang, Chuan Chen, Fanghua Ye, Jiajing Wu, Zibin Zheng, and Guohui Ling. Hyper2vec: Biased random walk for hyper-network embedding. In *International Conference on Database Systems for Advanced Applications*, pp. 273–277. Springer, 2019.

Jing Huang and Jie Yang. Unignn: a unified framework for graph and hypergraph neural networks. *arXiv preprint arXiv:2105.00956*, 2021.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. Dynamic hypergraph neural networks. In *Ijcai*, pp. 2635–2641, 2019.

Yangqin Jiang, Chao Huang, and Lianghao Huang. Adaptive graph contrastive learning for recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4252–4261, 2023.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 66–74, 2020.

Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, et al. A comprehensive survey on deep graph representation learning. *Neural Networks*, 173:106207, 2024a.

Wei Ju, Zhengyang Mao, Siyu Yi, Yifang Qin, Yiyang Gu, Zhiping Xiao, Yifan Wang, Xiao Luo, and Ming Zhang. Hypergraph-enhanced dual semi-supervised graph classification. *arXiv preprint arXiv:2405.04773*, 2024b.

Wei Ju, Siyu Yi, Kangjie Zheng, Yifan Wang, Ziyue Qiao, Li Shen, Yongdao Zhou, Xiaochun Cao, and Jiancheng Lv. Compactness and consistency: A conjoint framework for deep graph clustering. In *The Fourteenth International Conference on Learning Representations*, 2026.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 16–22, 1999.

Dongjin Lee and Kijung Shin. I'm me, we're us, and i'm us: Tri-directional contrastive learning on hypergraphs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 8456–8464, 2023.

Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. In *International Conference on Machine Learning*, pp. 2965–2974. PMLR, 2018.

Fan Li, Xiaoyang Wang, Dawei Cheng, Wenjie Zhang, Ying Zhang, and Xuemin Lin. Hypergraph self-supervised learning with sampling-efficient signals. *arXiv preprint arXiv:2404.11825*, 2024.

Meng Liu, Nate Veldt, Haoyu Song, Pan Li, and David F Gleich. Strongly local hypergraph diffusions for clustering and semi-supervised learning. In *Proceedings of the Web Conference 2021*, pp. 2092–2103, 2021.

Li Ni, Shuaikang Zeng, Lin Mu, and Longlong Lin. From representation to clusters: A contrastive learning approach for attributed hypergraph clustering. In *Proceedings of the ACM Web Conference 2026*, pp. 1080–1091, 2026.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Yuhan Quan, Jingtao Ding, Chen Gao, Lingling Yi, Depeng Jin, and Yong Li. Robust preference-guided denoising for graph based social recommendation. In *Proceedings of the ACM web conference 2023*, pp. 1097–1108, 2023.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

Ryan Rossi and Nesreen Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Shota Saito, Danilo Mandic, and Hideyuki Suzuki. Hypergraph p-laplacian: A differential geometry view. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Yumeng Song, Yu Gu, Tianyi Li, Jianzhong Qi, Zhenghao Liu, Christian S Jensen, and Ge Yu. Chgnn: A semi-supervised contrastive hypergraph learning network. *IEEE Transactions on Knowledge and Data Engineering*, 36(9):4515–4530, 2024.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec):583–617, 2002.

Wen Tao, Yuansheng Liu, Xuan Lin, Bosheng Song, and Xiangxiang Zeng. Prediction of multi-relational drug–gene interaction via dynamic hypergraph contrastive learning. *Briefings in Bioinformatics*, 24(6):bbad371, 2023.

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Lu Wang, Wenchao Yu, Wei Wang, Wei Cheng, Wei Zhang, Hongyuan Zha, Xiaofeng He, and Haifeng Chen. Learning robust representations with graph denoising policy network. In *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1378–1383. IEEE, 2019.

Qianqian Wang, Bowen Zhao, Zhengming Ding, Xiangdong Zhang, and Quanxue Gao. A simple yet effective hypergraph clustering network. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 6352–6360, 2025.

Tianxin Wei, Yuning You, Tianlong Chen, Yang Shen, Jingrui He, and Zhangyang Wang. Augmentations in hypergraph contrastive learning: Fabricated and generative. *Advances in neural information processing systems*, 35:1909–1922, 2022.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487. PMLR, 2016.

Zhengrui Xu, Guan'an Wang, Xiaowen Huang, and Jitao Sang. Denoiserep: Denoising model for representation learning. *Advances in Neural Information Processing Systems*, 37:40032–40056, 2024.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. Hypergcn: A new method for training graph convolutional networks on hypergraphs. *Advances in neural information processing systems*, 32, 2019.

Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudré-Mauroux. Lbsn2vec++: Heterogeneous hypergraph embedding for location-based social networks. *IEEE Transactions on Knowledge and Data Engineering*, 34(4):1843–1855, 2020.

Siyu Yi, Zhengyang Mao, Yifan Wang, Yiyang Gu, Zhiping Xiao, Chong Chen, Xian-Sheng Hua, Ming Zhang, and Wei Ju. Hypergraph consistency learning with relational distillation. *IEEE Transactions on Multimedia*, 2025.

Kun Zeng, Jun Yu, Cuihua Li, Jane You, and Taisong Jin. Image clustering by hyper-graph regularized non-negative matrix factorization. *Neurocomputing*, 138:209–217, 2014.

Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33:9263–9275, 2020.

Tianxiang Zhao, Youqing Wang, Jinlu Wang, Jiapu Wang, Mingliang Cui, Junbin Gao, and Jipeng Guo. Hybrid-collaborative augmentation and contrastive sample adaptive-differential awareness for robust attributed graph clustering. *arXiv preprint arXiv:2510.02731*, 2025.

Hao Zhong, Yubo Zhang, Chenggang Yan, Zuxing Xuan, Ting Yu, Ji Zhang, Shihui Ying, and Yue Gao. Penalized flow hypergraph local clustering. *IEEE Transactions on Knowledge and Data Engineering*, 36(5):2110–2125, 2023.

Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19, 2006.

Dingyuan Zhu, Ziwei Zhang, Peng Cui, and Wenwu Zhu. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1399–1407, 2019.

Appendix

A Proof of Theorem

A.1 Proof of Theorem 1

Proof. The assumptions in Theorem 1 imply that the activation function satisfies:

$$\|\rho(P_1) - \rho(P_2)\|_F \le L_\rho \|P_1 - P_2\|_F,$$

the clean and denoised propagation operators satisfy:

$$\|\Theta^s\|_2 \le \kappa, \qquad \|\Theta^\star\|_2 \le \kappa,$$

and the layer-wise weights and clean hidden representations satisfy:

$$\|W_l\|_2 \le \nu_l, \qquad \|Z_l^\star\|_F \le R_l,$$

which are easy to satisfy in practice. Let $e_l = \|Z_l^s - Z_l^\star\|_F$. For the l-th layer, we have:

$$e_l = \left\|\rho(\Theta^s Z_{l-1}^s W_l) - \rho(\Theta^\star Z_{l-1}^\star W_l)\right\|_F.$$

By the L_ρ-Lipschitz continuity of $\rho(\cdot)$, we have:

$$e_l \le L_\rho \left\|\Theta^s Z_{l-1}^s W_l - \Theta^\star Z_{l-1}^\star W_l\right\|_F.$$

Adding and subtracting $\Theta^s Z_{l-1}^\star W_l$ gives:

$$\begin{aligned}&\Theta^s Z_{l-1}^s W_l - \Theta^\star Z_{l-1}^\star W_l\\ =&\Theta^s (Z_{l-1}^s - Z_{l-1}^\star) W_l + (\Theta^s - \Theta^\star) Z_{l-1}^\star W_l.\end{aligned}$$

Therefore, we have:

$$\begin{aligned}e_l \le\ & L_\rho \left\|\Theta^s (Z_{l-1}^s - Z_{l-1}^\star) W_l\right\|_F\\ &+ L_\rho \left\|(\Theta^s - \Theta^\star) Z_{l-1}^\star W_l\right\|_F.\end{aligned}$$

Using the submultiplicativity of matrix norms, we have:

$$\left\|\Theta^s (Z_{l-1}^s - Z_{l-1}^\star) W_l\right\|_F \le \|\Theta^s\|_2 \|Z_{l-1}^s - Z_{l-1}^\star\|_F \|W_l\|_2,$$

and:

$$\left\|(\Theta^s - \Theta^\star) Z_{l-1}^\star W_l\right\|_F \le \|\Theta^s - \Theta^\star\|_2 \|Z_{l-1}^\star\|_F \|W_l\|_2.$$

By the boundedness assumptions, we obtain:

$$e_l \le L_\rho \kappa \nu_l e_{l-1} + L_\rho \nu_l R_{l-1} \delta_\Theta^s.$$

That is,

$$e_l \le a_l e_{l-1} + b_l \delta_\Theta^s,$$

where $a_l = L_\rho \kappa \nu_l$ and $b_l = L_\rho \nu_l R_{l-1}$. Since $e_0 = \|X^s - X^\star\|_F = \delta_X^s$, recursively expanding the above inequality yields:

$$e_L \le \left(\prod_{t=1}^{L} a_t\right) \delta_X^s + \left(\sum_{l=1}^{L} b_l \prod_{t=l+1}^{L} a_t\right) \delta_\Theta^s.$$

This completes the proof. □

A.2 Proof of Theorem 2

Proof. Let the clean cluster labels be $c_i^\star \in \{1, \dots, K\}$ and the clean cluster centers be $\mu_1^\star, \dots, \mu_K^\star \in \mathbb{R}^q$. The ideal center matrix $M^\star \in \mathbb{R}^{n \times q}$ is defined by $M_i^\star = \mu_{c_i^\star}^\star$. The minimum center separation and the clean within-cluster residual are:

$$\Delta = \min_{a \ne b} \|\mu_a^\star - \mu_b^\star\|_2 > 0,$$

and:

$$E^\star = Z^\star - M^\star, \qquad \|E^\star\|_F \le \sqrt{n}r.$$

Since $\hat{M}^s$ is obtained by a $(1+\xi)$-approximate K-means solution on Z^s, it satisfies:

$$\|Z^s - \hat{M}^s\|_F^2 \le (1+\xi) \min_{M \in \mathcal{M}_K} \|Z^s - M\|_F^2,$$

where $\mathcal{M}_K$ denotes the set of matrices with at most K distinct rows. Since $M^\star \in \mathcal{M}_K$, it is a feasible center-assignment matrix for the K-means objective. Hence,

$$\|Z^s - \hat{M}^s\|_F^2 \le (1+\xi)\|Z^s - M^\star\|_F^2.$$

Taking square roots gives:

$$\|Z^s - \hat{M}^s\|_F \le \sqrt{1+\xi}\|Z^s - M^\star\|_F.$$

By the triangle inequality, we have:

$$\begin{aligned}\|\hat{M}^s - M^\star\|_F &\le \|\hat{M}^s - Z^s\|_F + \|Z^s - M^\star\|_F \\ &\le (1+\sqrt{1+\xi})\|Z^s - M^\star\|_F.\end{aligned}$$

Furthermore, we have:

$$\begin{aligned}\|Z^s - M^\star\|_F &\le \|Z^s - Z^\star\|_F + \|Z^\star - M^\star\|_F \\ &\le \epsilon_Z^s + \sqrt{n}r.\end{aligned}$$

Therefore,

$$\|\hat{M}^s - M^\star\|_F \le (1+\sqrt{1+\xi})(\epsilon_Z^s + \sqrt{n}r).$$

Now we define the bad node set as

$$\mathcal{B}^s = \left\{ i : \|\hat{M}_i^s - M_i^\star\|_2 \ge \frac{\Delta}{2} \right\},$$

where $\hat{M}_i^s$ and $M_i^\star$ denote the i-th rows of $\hat{M}^s$ and $M^\star$, respectively. For every $i \in \mathcal{B}^s$, we have:

$$\|\hat{M}_i^s - M_i^\star\|_2^2 \ge \frac{\Delta^2}{4}.$$

Hence, we have:

$$|\mathcal{B}^s|\frac{\Delta^2}{4} \le \sum_{i \in \mathcal{B}^s} \|\hat{M}_i^s - M_i^\star\|_2^2 \le \|\hat{M}^s - M^\star\|_F^2.$$

Using the upper bound on $\|\hat{M}^s - M^\star\|_F$, we obtain:

$$|\mathcal{B}^s| \le \frac{4(1+\sqrt{1+\xi})^2}{\Delta^2}(\epsilon_Z^s + \sqrt{n}r)^2.$$

It remains to connect the bad-set bound with clustering error. For any node $i \notin \mathcal{B}^s$, we have:

$$\|\hat{M}_i^s - \mu_{c_i^\star}^\star\|_2 < \frac{\Delta}{2}.$$

Since different clean centers are separated by at least Δ, the same estimated cluster center cannot be within distance $\Delta/2$ of two different clean centers. Thus, each good estimated cluster has a unique corresponding clean cluster. If $|\mathcal{B}^s| < n_{\min}$, where $n_{\min} = \min_k |\{i : c_i^\star = k\}|$, then every clean cluster contains at least one good node, which ensures a one-to-one correspondence between the estimated clusters and the clean clusters. After applying the corresponding permutation of cluster indices, all good nodes are correctly assigned, and only nodes in $\mathcal{B}^s$ can be misclustered. Therefore, we have:

$$\mathrm{Err}(Z^s) \le \frac{|\mathcal{B}^s|}{n} \le \frac{4(1+\sqrt{1+\xi})^2}{n\Delta^2}(\epsilon_Z^s + \sqrt{n}r)^2.$$

Finally, by Theorem 1, we have:

$$\epsilon_Z^s = \|Z^s - Z^\star\|_F \le A_L\delta_X^s + B_L\delta_\Theta^s.$$

Substituting this into the clustering error bound gives:

$$\mathrm{Err}(Z^s) \le \frac{4(1+\sqrt{1+\xi})^2}{n\Delta^2}\left(A_L\delta_X^s + B_L\delta_\Theta^s + \sqrt{n}r\right)^2.$$

This completes the proof. □

B Time Complexity Analysis

Let $S = \|H\|_0$ denote the number of observed hyperedge incidences, h the encoder dimension, and $p = |\Omega|$ the number of learnable parameters. Generating the masks costs $\mathcal{O}(nd + Sd)$. The L-layer HGNN encoder requires $\mathcal{O}(L(Sh + nh^2))$, while the high- and low-dimensional projections cost $\mathcal{O}(nh(r_h + K))$. The cross-view contrastive objectives require $\mathcal{O}(n^2(r_h + K))$, and the clustering-guided module costs $\mathcal{O}(nKh)$. Computing the pruning threshold and masking gradients further requires $\mathcal{O}(p \log p)$. Therefore, the overall time complexity can be approximated as $\mathcal{O}(n^2)$. And the implementation of our ADHERE are in: https://anonymous.4open.science/r/ADHERE-F581/

C Optimization Algorithm

The complete training procedure of ADHERE is summarized in Algorithm 1.

Algorithm 1 The overall training process of ADHERE

Input: Hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, X, H)$; cluster number K; maximum epochs T; hyperparameters $\tau_d, \tau_h, \lambda_1, \lambda_2, \beta$, and η
Output: Optimized parameters Ω and view representations Z^1, Z^2

Initialize Ω and $t \leftarrow 0$
while not converged and $t < T$ **do**
Adaptive denoising and dual-view encoding
  Learn T_X, T_H via stochastic reparameterization and construct $\widetilde{\mathcal{G}}^1, \widetilde{\mathcal{G}}^2$ ▷ Eqs. (1)–(3)
  Compute $\mathcal{L}_{den}$ and encode $Z^1 = f_\theta(\hat{X}, H)$, $Z^2 = f_\theta(X, \hat{H})$ ▷ Eqs. (4)–(7)
Cross-view contrastive learning
  Project Z^s to Q^s and compute $\mathcal{L}_{con}$ across the two views ▷ Eq. (8)
Multi-view clustering-guided learning
  for $s \in \{1, 2\}$ **do**
    Obtain P^s and aligned K-means pseudo-labels $\hat{y}^s$ ▷ Eqs. (9)–(10)
  end for
  Compute $\mathcal{L}_{clu} = \mathcal{L}_{intra} + \mathcal{L}_{cross}$ ▷ Eqs. (11)–(13)
  $\mathcal{L} \leftarrow \mathcal{L}_{con} + \lambda_1 \mathcal{L}_{clu} + \lambda_2 \mathcal{L}_{den}$ ▷ Eq. (16)
Magnitude-aware gradient update
  $g_o^{(t)} \leftarrow \nabla_{\omega_o^{(t)}} \mathcal{L}$, $M_{\Omega,o}^{(t)} \leftarrow \mathbf{1}(|\omega_o^{(t)}| \geq \psi(\beta))$
  $\omega_o^{(t+1)} \leftarrow \omega_o^{(t)} - \eta\, g_o^{(t)} \odot M_{\Omega,o}^{(t)}, \forall \omega_o^{(t)} \in \Omega^{(t)}$ ▷ Eqs. (15)
  $t \leftarrow t + 1$
end while
return Ω, Z^1, Z^2

D Related Work

D.1 Hypergraph Representation Learning

Hypergraph representation learning aims to learn underlying node embeddings from hypergraph data, preserving both node feature information and the high-order structural information encoded by hyperedges for downstream tasks. Early hypergraph learning methods are mostly based on spectral graph theory or hypergraph Laplacian operators, which transform hypergraph partitioning into spectral embedding or matrix factorization problems (Zhou et al., 2006; Saito et al., 2018). With the development of GNN (Ju et al., 2024a), researchers begin to extend message passing mechanisms to hypergraph structures (Feng et al., 2019b; Jiang et al., 2019; Huang & Yang, 2021). Most early methods are designed for tasks such as semi-supervised node classification (Yadati et al., 2019; Chien et al., 2022) and usually rely on label supervision. However, they do not sufficiently consider how to learn robust representations in unsupervised scenarios. Recently, some studies focus on exploiting hypergraph data in unsupervised settings for clustering tasks (Liu et al., 2021; Wang et al., 2025; Ni et al., 2026; Hu et al., 2021; Zhong et al., 2023). Among them, PFHLC (Zhong et al., 2023) enhances hypergraph local clustering by incorporating random-walk-expanded seeds into penalized flow. Moving beyond flow-based local clustering, HCN (Wang et al., 2025) preserves cluster structures through smoothing preprocessing, self-diagonal consistency, and structure alignment. More

recently, CAHC (Ni et al., 2026) jointly optimizes node embeddings and cluster assignments through node- and hyperedge-level contrastive learning with clustering guidance. Although these methods can achieve promising clustering performance, they largely ignore the noise in hypergraph data. In contrast, in our method we enable the model to adaptively remove noise from both node features and hypergraph structures, thereby improving the robustness of the learned representations.

D.2 Denoising Learning

Denoising learning aims to recover stable and reliable representations from corrupted or incomplete data. One early representative work is the Denoising Autoencoder (Vincent et al., 2008; 2010), which artificially corrupts the input data and trains the model to reconstruct the original input, encouraging the model to learn robust features that are insensitive to local perturbations. This idea is then be widely used in representation learning, image restoration, speech enhancement, and other tasks (Lehtinen et al., 2018; Ho et al., 2020; He et al., 2022; Xu et al., 2024; Jiang et al., 2023). In graph representation learning, the noise problem becomes more complex because noise may arise from both node features and graph structure. To address this, existing studies explore various strategies from different perspectives (Feng et al., 2019a; Zhang & Zitnik, 2020; Gasteiger et al., 2019; Ju et al., 2026; Zhu et al., 2019; Jin et al., 2020; Rong et al., 2019). GDC (Gasteiger et al., 2019) mitigates structural noise by replacing the original adjacency matrix with a sparsified diffusion matrix. In addition, RGCN (Zhu et al., 2019) enhances robustness by modeling node embeddings as Gaussian distributions to mitigate adversarial perturbations. Furthermore, Pro-GNN (Jin et al., 2020) jointly learns a clean graph structure and a robust GNN under sparsity and feature-smoothness constraints. However, most existing denoising methods are designed for standard graphs or recommendation bipartite graphs (Wang et al., 2019; Quan et al., 2023), leaving denoising learning largely unexplored in hypergraph clustering. In this paper we introduce denoising learning into hypergraph clustering and develop an adaptive denoising mechanism to enhance the robustness and performance of the learned representations in clustering tasks.

E Supplementary Material of Experiments

E.1 Datasets and baselines

Details of Datasets. The datasets used in our experiments span co-citation (Cora-C, Citeseer, Pubmed), co-authorship (Cora-A, DBLP), news documents (20NewsW100), mushroom (Mushroom) and visual domains (NTU2012). In Cora-C, Citeseer, and Pubmed, nodes denote papers and hyperedges connect papers co-cited by the same work. In Cora-A and DBLP, hyperedges correspond to authors and connect their publications. For 20NewsW100, nodes are documents and selected words define hyperedges linking documents in which they appear. In Mushroom, hyperedges connect instances sharing categorical attributes, while labels indicate edibility. In NTU2012, nodes represent 3D objects, and hyperedges are constructed from feature-space nearest neighbors. Node features are textual, categorical, or visual descriptors depending on the dataset, and labels denote semantic categories. Since we study unsupervised clustering, all features and hypergraph structures are used for training, whereas ground-truth labels are used only for evaluation.

Details of Baselines. For embedding-based baselines, the learned embeddings are fed into k-means to obtain clustering results. For graph-based baselines, hypergraphs are first converted into graphs via clique expansion. For hypergraph-based baselines, we use the results reported under their available official experimental settings. The details of the baselines are as follows:

Node2vec (Grover & Leskovec, 2016): This is a classical graph representation learning method that generates node sequences through biased random walks and uses the skip-gram model.

DGI (Veličković et al., 2018): It is an unsupervised graph representation learning framework that maximizes mutual information between local node representations and a global graph summary.

RAGC (Zhao et al., 2025): This is a robust attributed graph clustering method that employs hybrid collaborative augmentation and adaptive contrastive sample weighting.

Hyper2vec (Huang et al., 2019): A hypergraph method that extends the random walk paradigm to hypergraphs and combines biased second-order random walks with the skip-gram framework.

TriCL (Lee & Shin, 2023): A representative contrastive hypergraph learning method that captures node, group, and membership-level information through tri-directional contrastive learning.

SE-HSSL (Li et al., 2024): A hypergraph learning method that learns node representations by exploiting hypergraph structural information.

HCN (Wang et al., 2025): It is a hypergraph clustering method that preserves semantic and structural patterns through smoothing preprocessing, self-diagonal consistency, and structure alignment.

CAHC (Ni et al., 2026): An attributed hypergraph clustering method that jointly learns node embeddings and cluster assignments by combining node-level and hyperedge-level contrastive learning with clustering-oriented guidance.

Evaluation Metrics. We use four widely adopted metrics to evaluate clustering performance: Accuracy (ACC), Macro F1-score (F1), Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI). ACC measures the best matching between predicted clusters and ground-truth labels, and F1 evaluates the harmonic mean of precision and recall across classes. NMI measures the mutual dependence between predicted cluster assignments and ground-truth labels, while ARI measures the agreement between two partitions after adjusting for chance. For all of the metrics above, higher values indicate better clustering performance.

E.2 Implementation Details.

We conduct experiments in PyTorch (Paszke et al., 2019) and PyG (Fey & Lenssen, 2019) frameworks, and all models are trained on an NVIDIA RTX 3090 GPU with 24GB memory. Since different hypergraph datasets have different initial feature dimensions, the dimensions of the latent space are not fixed across datasets. However, we use the same settings for the following hyperparameters: the learning rate η is set to 0.0002, the parameter pruning rate β is set to 0.5, the weights for balancing losses are set to $\lambda_1 = \lambda_2 = 0.5$, and the weight decay ratio is set to 10^{-5}. For the final node representations learned by the model, we apply k-means clustering and use the Hungarian algorithm to align the predicted cluster labels with the ground-truth labels, thereby obtaining the four evaluation metrics. For each dataset, we train the model for 400 epochs and report the final results averaged over five runs. For the baselines, we follow the experimental settings reported in their official papers.

E.3 Ablation Study

In order to verify the contribution of each key component in ADHERE, we conduct an ablation study and report the results in Table 1. The complete model achieves the best results on almost all datasets and metrics, indicating that these components are not isolated improvements, but jointly contribute to robust and clustering-friendly hypergraph representations.

Removing the whole denoising module, i.e., w/o $\mathcal{L}_{den}$, generally leads to clear performance degradation, confirming the necessity of suppressing feature and structural noise before representation learning. When the two denoised views are removed separately, i.e., w/o $\widetilde{\mathcal{G}}^1$ and w/o $\widetilde{\mathcal{G}}^2$, the results also decline in most cases. This suggests that feature denoising and structure denoising capture complementary information, while their relative importance may vary across datasets.

The objective-level variants further show the importance of the proposed learning strategy. Removing $\mathcal{L}_{con}$ weakens cross-view semantic alignment and often causes large drops, indicating that contrastive learning is essential for exploiting the complementary denoised views. Removing $\mathcal{L}_{clu}$ also reduces performance, showing that high-dimensional representations learned by contrastive learning still need direct clustering guidance. Overall, the ablation results demonstrate that adaptive denoising, cross-view contrastive learning, and low-dimensional clustering supervision work together to improve the robustness and discriminability of the learned representations.

E.4 Hyper-parameter Analysis

We further analyze the influence of the parameter pruning ratio β on clustering performance, which controls the proportion of parameters to be pruned during gradient updates. We vary β in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and report ACC, F1, NMI, and ARI. The results are shown in Figure 1. Overall, ADHERE remains relatively stable under different values of β, indicating that it is not

Table 1: Ablation study in terms of all metrics across multiple datasets (%).

Ablation Variant	Metric	Cora-C	Citeseer	Pubmed	Cora-A	DBLP	20NewsW100	Mushroom	NTU2012
w/o $\mathcal{L}_{den}$	ACC	64.78	56.65	69.40	67.46	76.50	69.99	86.20	69.98
	F1	64.13	54.23	68.23	66.04	74.29	64.29	86.07	60.98
	NMI	41.88	31.14	29.86	43.07	58.03	36.46	43.35	82.31
	ARI	35.34	29.44	30.41	40.46	53.16	39.40	52.41	67.77
w/o $\widetilde{\mathcal{G}}^1$	ACC	71.69	58.29	72.16	65.49	78.33	70.81	84.88	68.73
	F1	69.25	53.81	71.34	65.32	76.96	67.71	84.77	60.76
	NMI	49.44	29.98	35.35	43.51	60.34	38.05	39.35	81.89
	ARI	47.48	29.20	35.39	36.58	55.17	40.75	48.66	61.86
w/o $\widetilde{\mathcal{G}}^2$	ACC	68.06	59.80	72.55	64.07	77.42	70.36	86.04	69.28
	F1	66.50	56.12	71.58	64.14	75.64	66.14	85.91	60.83
	NMI	48.49	32.89	34.20	42.78	59.28	37.03	42.84	81.99
	ARI	42.91	32.56	36.37	35.32	52.83	40.61	51.95	63.20
w/o $\mathcal{L}_{con}$	ACC	59.83	56.99	67.81	60.97	68.21	69.63	85.78	67.94
	F1	59.13	52.88	67.25	51.87	66.98	63.95	85.63	62.79
	NMI	36.01	29.26	32.12	39.27	45.88	35.84	42.48	82.17
	ARI	29.52	28.44	29.28	38.57	43.90	38.83	51.20	60.39
w/o $\mathcal{L}_{clu}$	ACC	74.26	65.91	71.53	69.80	74.40	73.16	84.50	69.18
	F1	72.62	61.79	70.73	67.91	70.80	68.63	84.50	61.20
	NMI	54.49	39.84	32.66	47.75	52.91	38.80	38.12	82.03
	ARI	52.11	39.63	34.18	44.88	51.92	45.13	47.61	62.62
ADHERE	ACC	**75.17**	**69.97**	**73.84**	**71.26**	**79.16**	**74.78**	**88.55**	**71.44**
	F1	**74.03**	**65.03**	**72.87**	**68.28**	**78.11**	**72.92**	**88.53**	**63.94**
	NMI	**55.51**	**44.99**	**37.23**	**48.95**	**61.66**	**42.40**	**50.52**	**83.40**
	ARI	**52.35**	**46.26**	**39.22**	**46.37**	**59.63**	**47.70**	**59.49**	**67.91**

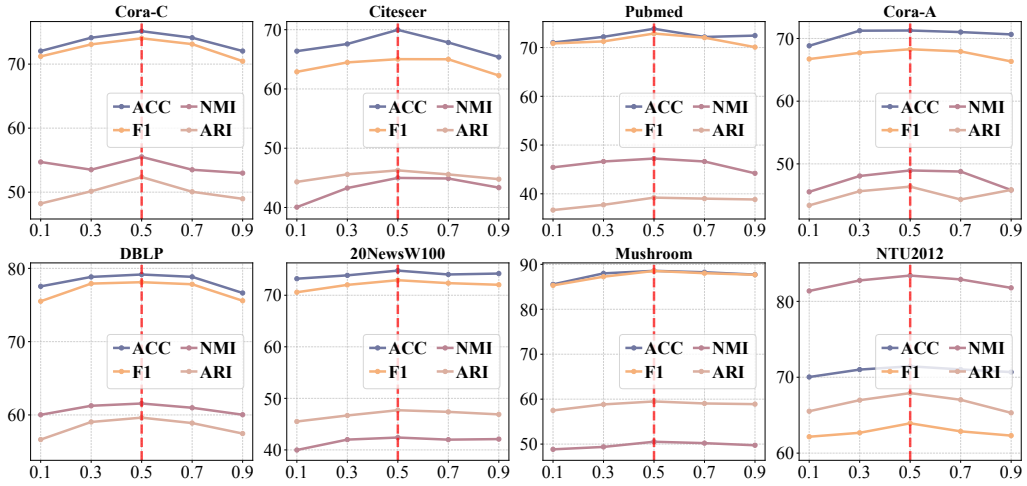


Figure 1: Impact of different values of β on model performance across the benchmark datasets. The best-performing setting is highlighted by the vertical red dotted line.

highly sensitive to the pruning ratio. Most datasets achieve the best or near-best results around $\beta = 0.5$, suggesting that this moderate pruning strength provides a good balance between suppressing noisy gradients and preserving model capacity. When β increases from 0.1 to 0.5, the performance generally improves on most datasets. This is because a small pruning ratio cannot sufficiently filter unstable gradients from less important parameters, allowing noisy updates to affect representation learning. However, when β further increases to 0.7 or 0.9, the performance begins to decline on several datasets. An overly large pruning ratio prevents too many parameters from being updated, which weakens the expressive capacity and adaptability of the model. This is particularly harmful for hypergraph clustering, where the model needs to jointly learn denoised views, contrastive repre-

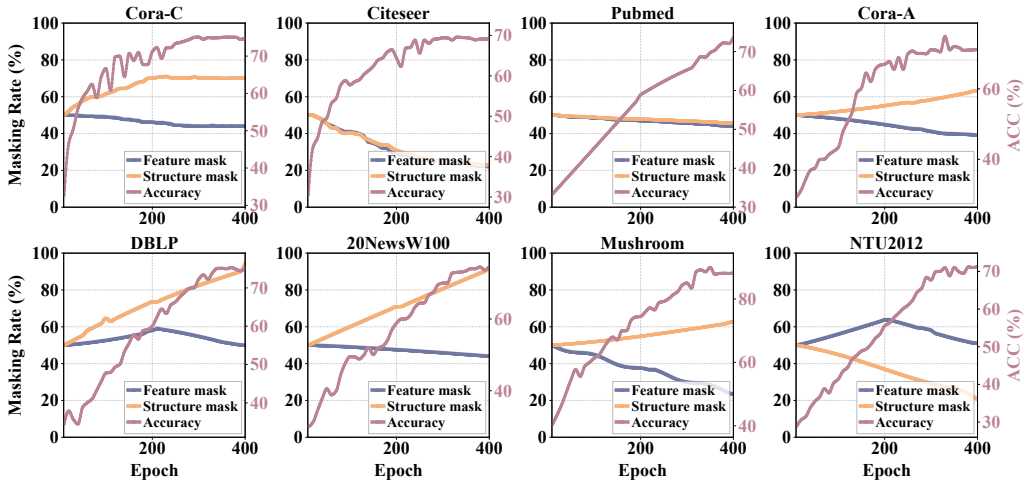


Figure 2: Illustration of the dynamic changes of the denoising module during training. The left Y-axis represents the masking rate of feature mask and structure mask, i.e., the proportions of masked feature information and masked hyperedge incidences; The right Y-axis represents the values of accuracy; The X-axis represents the training epochs.

sentations, and label-space distributions. With a moderate pruning ratio, the model focuses more on updating important parameters, which improves training stability and clustering performance. Across different datasets, $\beta = 0.5$ usually yields the most stable and competitive results. Therefore, we set the default parameter pruning ratio to $\beta = 0.5$ in this paper.

E.5 Analysis on Denoising Module

To verify whether the adaptive denoising module can learn useful feature and structural information during training, we record the retention ratios of the feature and structure masks together with clustering accuracy. As shown in Figure 2, the accuracy on all datasets generally increases and then becomes stable, indicating that the denoising module gradually selects input components that are beneficial to clustering rather than relying on fixed or random masking.

The mask curves further show dataset-dependent denoising behaviors. For structural denoising, the retention ratio increases on datasets such as DBLP and 20NewsW100, suggesting that their hypergraph structures contain useful high-order information and should be largely preserved. In contrast, the structural retention ratio decreases on datasets such as Citeseer, Pubmed, and NTU2012, which implies that suppressing unreliable hyperedge incidences can reduce structural noise propagation. For feature denoising, the retention ratio decreases on most datasets, indicating that redundant or noisy attribute dimensions are progressively filtered out. On DBLP and NTU2012, the feature mask first retains more information and then becomes stable or decreases, suggesting that the model may first build semantic representations and then remove less useful features.

Overall, the improvement in accuracy is usually accompanied by the stabilization of both masks. This training pattern shows that ADHERE can adaptively balance feature preservation and structure preservation according to dataset characteristics, thereby selecting reliable information and improving the discriminability of hypergraph clustering.

E.6 Effect of Denoised View Construction

To evaluate the effect of view construction, we compare three strategies: Denoised, which uses the proposed adaptive denoising mechanism; Original, which directly uses the raw input view without denoising; and Random, which constructs views through random feature and structure perturbations. Figure 3 reports the results on eight datasets under several metrics. Across all datasets and metrics, Denoised consistently outperforms both Original and Random, demonstrating the effectiveness of adaptive denoised view construction. Compared with Original, the denoised views alleviate the

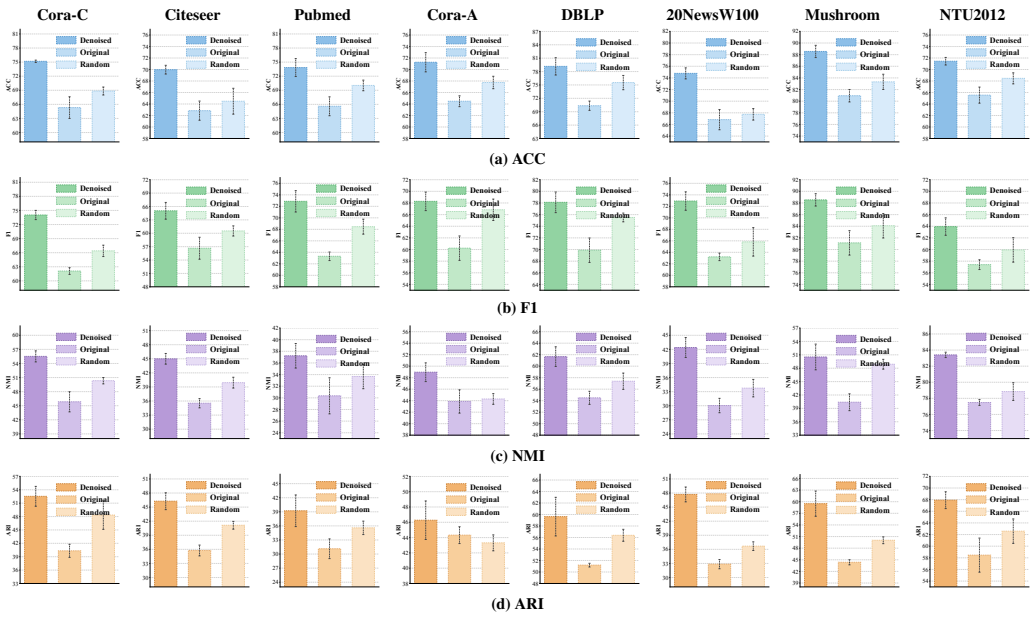


Figure 3: Comparison of three view construction strategies on eight datasets. Each column corresponds to a dataset, and each row reports ACC, F1, NMI, and ARI, respectively. In each subfigure, Denoised denotes the proposed adaptive denoising strategy, while Original and Random denote the original-view and random-augmentation variants, respectively.

influence of redundant feature dimensions and unreliable hyperedge incidences that may otherwise propagate through hypergraph convolution. Compared with Random, adaptive denoising avoids indiscriminately perturbing useful information and better preserves cluster-relevant features and structures. These results confirm that the performance gains stem not only from using multiple views, but also from constructing high-quality denoised views for subsequent learning modules.