CONFLICT-AWARE COMPLEMENTARY BALANCING FOR ATTRIBUTED GRAPH CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Attributed graph clustering (AGC) often combines multiple objectives to learn a shared representation. Although these objectives can provide complementary supervision, their gradients may become incompatible when jointly updating the same encoder, leading to unstable representation evolution and less coherent cluster structure. To address this issue, we propose Conflict-Aware Complementary Balancing (CACB), which explicitly coordinates complementary objectives according to their gradient compatibility during shared-encoder optimization. CACB first introduces objective-specific linear mappings to decouple heterogeneous supervisory signals in dedicated learned spaces, allowing different objectives to preserve their distinct optimization roles. CACB further imposes compatibility constraints on the aggregate encoder gradient. When incompatibility is detected, it formulates gradient correction as a projection that minimizes deviation over an affine update set subject to these constraints, yielding a closed-form update with minimal intervention that optimally repairs first-order incompatibility across objectives. Extensive experiments on six benchmark datasets demonstrate that CACB consistently outperforms competitive attributed graph clustering methods. Code is available at `https://anonymous.4open.science/r/CACB-EC44`.

1 INTRODUCTION

Attributed graph clustering (AGC) aims to learn clustering-oriented node representations without labels by jointly exploiting graph topology and node attributes. Early deep approaches combined structural encoding with representation fusion (Wang et al., 2019; Tu et al., 2021), while structural self-training further refined clustering assignments (Bo et al., 2020). Subsequent methods increasingly adopted scalable self-supervision to exploit structural and attribute information (Devvrit et al., 2022), and recent studies further construct richer clustering-oriented supervision through reliable relational signals (Liu et al., 2024c), topology-aware complementary views (Zhao et al., 2026b), and compactness–consistency learning (Ju et al., 2026). Task-dependent fusion further adapts how different supervisory signals contribute to individual nodes (Zhu et al., 2024), while recent re-evaluation emphasizes standardized comparison across deep AGC methods (Lai et al., 2023). Despite these advances in designing and combining supervisory signals, their optimization interaction on a shared encoder remains comparatively underexplored: objectives that are complementary in function need not induce compatible gradients during joint optimization.

Figure 1 provides an intuitive illustration of the problem. As shown in Fig. 1(a), neighborhood attraction and nonedge separation capture distinct but jointly useful aspects of the local geometry: observed neighbors are encouraged to contract, whereas sampled nonedges discourage excessive similarity. Although both signals are individually meaningful, their interaction through a shared encoder does not guarantee a compatible aggregate gradient. As illustrated in Fig. 1(b), fixed aggregation can drive the representation toward an ambiguous geometry. More generally, different supervisory signals need not induce consistent optimization directions. When their gradients conflict, fixed aggregation may cause useful signals to partially cancel each other, leading to a less coherent representation. Existing AGC studies mainly focus on constructing more informative supervision, while related graph-learning studies examine augmentation–representation interaction (Yang et al., 2024), message–contrastive conflict (Zhang et al., 2026), or temporal coordination of self-supervised objectives (Grover et al., 2026). The compatibility of the complete encoder gradient when multiple

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.

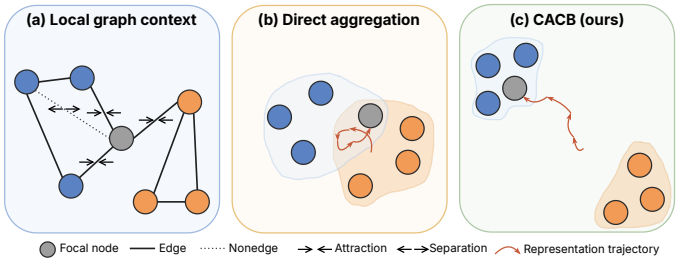


Figure 1: **Motivation of CACB. (a)** Distinct supervisory signals act on the same focal node. **(b)** Direct aggregation may yield an incompatible aggregate gradient. **(c)** CACB preserves compatible gradients and repairs incompatible ones by relational reallocation.

AGC objectives jointly update a shared encoder remains comparatively underexplored, motivating CACB's structured gradient repair.

To address this issue, we propose Conflict-Aware Complementary Balancing (CACB), a complementarity-constrained framework for coordinating shared-encoder optimization in AGC. Objective-specific projections first separate heterogeneous supervisory signals into dedicated learned spaces. CACB then evaluates the natural aggregate under compatibility constraints. Once incompatibility is detected, it performs compatibility-guided relational rebalancing that minimally perturbs the natural objective structure while preserving the base contribution and total relational budget. This one-dimensional formulation yields a closed-form minimum-intervention repair and restores joint first-order compatibility across objectives. This coordinated optimization promotes more coherent representations for attributed graph clustering. Our contributions are as follows:

- We propose CACB, which casts the joint optimization of complementary objectives in AGC as a gradient compatibility problem. This moves AGC supervision beyond loss design toward explicit coordination of objective gradients.
- We introduce a complementarity-constrained gradient repair, alongside lightweight objective-specific projections for separating heterogeneous supervisory signals. The repair minimally corrects incompatible aggregate gradients to restore joint first-order compatibility, with closed-form optimization and quantitative aggregate-descent guarantees.
- Extensive experiments on six benchmark datasets demonstrate the effectiveness of CACB against competitive attributed graph clustering methods.

2 Related Work

Deep Attributed Graph Clustering. Deep AGC methods jointly exploit graph structure and node attributes through attentional encoding (Wang et al., 2019), structural self-training (Bo et al., 2020), and representation fusion (Tu et al., 2021). More recent studies have focused on strengthening clustering supervision through scalable self-supervision (Devvrit et al., 2022), reliable similarities (Liu et al., 2024c), modularity-aware contrast (Liu et al., 2024b), compactness–consistency learning (Ju et al., 2026), and improved positive-sample construction (Shan et al., 2026). Learnable and cluster-preserving augmentations further enrich the views used for representation learning (Yang et al., 2024; Zheng et al., 2024), while homophily-aware affinity (Li et al., 2023), prototype-based false-negative treatment (Ma et al., 2025), and trustworthy neighborhood modeling (Peng et al., 2025) improve relational reliability.

Recent AGC studies increasingly enrich or adapt the supervisory signals used to learn a shared representation. CDRS couples pseudo-classification and clustering supervision through collaborative decision reinforcement (Zhu et al., 2022). DyFSS introduces node-specific fusion across self-supervised tasks (Zhu et al., 2024), while ADMRGC adaptively balances attribute and topological manifold regularization (Li & Chen, 2024). MEFGC combines multiple embeddings through joint self- and mutual supervision (Liu et al., 2024a), and RAGC further incorporates sample-adaptive weighting into hybrid contrastive learning (Zhao et al., 2026a). These efforts expand the forms and combinations of supervision available to AGC. CACB instead focuses on what happens when

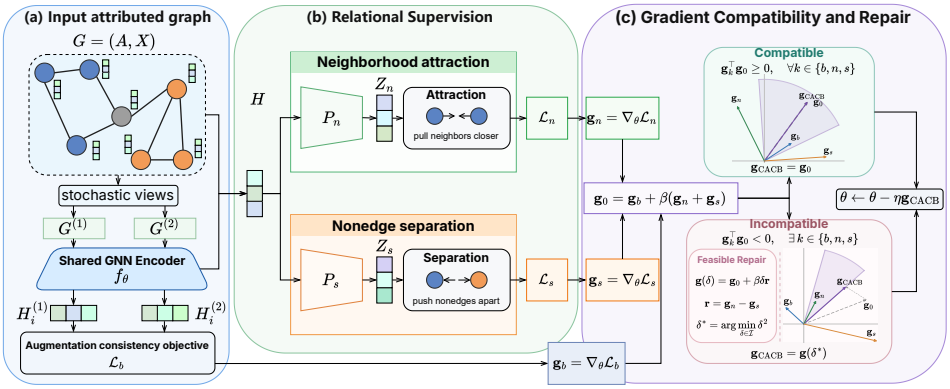


Figure 2: **Overview of CACB. (a)** Augmentation-consistent shared representation. **(b)** Objective-specific projections for neighborhood attraction and nonedge separation. **(c)** Gradient compatibility check and complementary repair.

such objectives jointly update the same encoder, namely whether their aggregate gradient remains compatible with the objectives that generate it.

Multi-Objective Gradient Coordination. Shared-parameter learning with multiple objectives can produce competing optimization directions (Zhang & Yang, 2021). MGDA formulates joint learning as multi-objective optimization and searches for a Pareto-oriented gradient combination (Sener & Koltun, 2018), whereas PCGrad modifies conflicting gradients through pairwise projection (Yu et al., 2020).

Graph-learning studies have also examined optimization interactions among self-supervised signals. G2CL studies conflict between message passing and contrastive learning and introduces gradient-guided coordination (Zhang et al., 2026), while ControlG uses gradient interference to regulate the temporal allocation of multiple self-supervised objectives (Grover et al., 2026). CACB focuses on complementary objectives in AGC and operates on their aggregate shared-encoder gradient, imposing gradient compatibility under the prescribed objective structure.

3 Method

Figure 2 illustrates the overall framework of CACB. It consists of three components: a base self-supervised objective for augmentation-consistent representation learning, objective-specific relational supervision in dedicated projection spaces, and complementarity-constrained gradient repair for coordinating shared-encoder gradients.

3.1 Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ be an undirected attributed graph with nodes $\mathcal{V} = \{v_1, \ldots, v_N\}$, observed edges $\mathcal{E}$, node attributes $\mathbf{X} \in \mathbb{R}^{N \times D}$, and adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$. Given the number of clusters K but without node labels, AGC learns a shared representation $\mathbf{H} = f_\theta(\mathbf{A}, \mathbf{X}) \in \mathbb{R}^{N \times d}$ and partitions the nodes into K clusters:

$$\hat{\mathbf{y}} = \text{Cluster}(\mathbf{H}, K), \qquad \hat{\mathbf{y}} \in \{1, \ldots, K\}^N, \tag{1}$$

where f_θ denotes the graph encoder and Cluster denotes a clustering algorithm such as K-means.

3.2 Base Self-Supervised Objective

We first define a base self-supervised objective that promotes augmentation-consistent representations. Two stochastic graph views are generated through independent feature masking and edge dropping and encoded by the same graph encoder as follows:

$$\mathbf{H}^{(r)} = f_\theta\left(\widetilde{\mathbf{A}}^{(r)}, \widetilde{\mathbf{X}}^{(r)}\right), \qquad r \in \{1, 2\}. \tag{2}$$

We enforce node-wise consistency between the two views by minimizing

$$\mathcal{L}_b = \frac{1}{N}\sum_{i=1}^{N}\left[1 - s\left(\mathbf{h}_i^{(1)}, \mathbf{h}_i^{(2)}\right)\right], \tag{3}$$

where $s(\cdot,\cdot)$ denotes cosine similarity. This base objective anchors the shared representation across stochastic views, while the relational objectives introduced below further shape inter-node structure on the representation $\mathbf{H} = f_\theta(\mathbf{A}, \mathbf{X})$ of the original input graph. Its encoder gradient participates in the gradient compatibility test, while its coefficient remains fixed during repair.

3.3 Objective-Specific Relational Supervision

While $\mathcal{L}_b$ enforces consistency across augmented views, we define two complementary relational objectives on the representation $\mathbf{H} = f_\theta(\mathbf{A}, \mathbf{X})$ of the original input graph. Neighborhood attraction encourages compactness along observed edges, while nonedge separation discourages excessive similarity between sampled nonedge pairs (Yang et al., 2023; Liu et al., 2024c). Since these two relations convey distinct supervisory semantics, we model them in separate learned spaces through objective-specific linear projections:

$$\mathbf{Z}_n = \mathbf{H}\mathbf{P}_n, \qquad \mathbf{Z}_s = \mathbf{H}\mathbf{P}_s, \tag{4}$$

where $\mathbf{P}_n, \mathbf{P}_s \in \mathbb{R}^{d\times d_p}$ are learnable bias-free projection heads for neighborhood attraction and nonedge separation, respectively. Both projections are used only during training, while final clustering is performed on the shared representation $\mathbf{H}$.

The neighborhood attraction objective is defined as follows:

$$\mathcal{L}_n = -\frac{1}{|\mathcal{E}|}\sum_{(i,j)\in\mathcal{E}} s\left(\mathbf{z}_{n,i}, \mathbf{z}_{n,j}\right), \tag{5}$$

where observed edges provide local structural supervision by encouraging neighboring nodes to remain similar in the corresponding relational space.

For nonedge separation, we adopt the following soft-margin objective:

$$\mathcal{L}_s = \frac{1}{|\overline{\mathcal{E}}_q|}\sum_{(i,j)\in\overline{\mathcal{E}}_q} \mathrm{softplus}\left(\frac{s\left(\mathbf{z}_{s,i}, \mathbf{z}_{s,j}\right) - m}{\tau}\right), \tag{6}$$

where $\overline{\mathcal{E}}_q$ contains qN directed nonedge pairs sampled at each step after excluding self-pairs and observed edges, $s(\cdot,\cdot)$ denotes cosine similarity, m is the similarity margin, and $\tau > 0$ controls the smoothness of the separation penalty.

For $k \in \{n, s\}$, the positive semidefinite matrix $\mathbf{M}_k = \mathbf{P}_k\mathbf{P}_k^\top$ characterizes the learned similarity geometry:

$$s\left(\mathbf{h}_i\mathbf{P}_k, \mathbf{h}_j\mathbf{P}_k\right) = \frac{\mathbf{h}_i\mathbf{M}_k\mathbf{h}_j^\top}{\sqrt{\mathbf{h}_i\mathbf{M}_k\mathbf{h}_i^\top}\sqrt{\mathbf{h}_j\mathbf{M}_k\mathbf{h}_j^\top}}. \tag{7}$$

Thus, each projection induces its own normalized bilinear similarity. Appendix A.1 further formalizes the separate parameterization of the two relational geometries. Because both relational objectives still backpropagate through the shared encoder, $\nabla_{\mathbf{H}}\mathcal{L}_k = \nabla_{\mathbf{Z}_k}\mathcal{L}_k\mathbf{P}_k^\top$, separate projection spaces alone do not guarantee compatibility of the resulting aggregate encoder gradient, motivating the repair introduced next.

3.4 Complementarity-Constrained Gradient Repair

Let $\mathcal{K} = \{b, n, s\}$ index the three objectives. At each training step, their encoder gradients and natural aggregate are defined as

$$\mathbf{g}_k = \nabla_\theta\mathcal{L}_k \quad (k \in \mathcal{K}), \qquad \mathbf{g}_0 = \mathbf{g}_b + \beta(\mathbf{g}_n + \mathbf{g}_s), \qquad \beta > 0, \tag{8}$$

where β controls the overall strength of relational supervision. Here $\mathbf{g}_0$ is the natural aggregate encoder gradient of $\mathcal{L}_b + \beta(\mathcal{L}_n + \mathcal{L}_s)$. The base consistency objective $\mathcal{L}_b$ acts as the representation anchor, whereas $\mathcal{L}_n$ and $\mathcal{L}_s$ form the complementary relational pair.

Under the natural aggregate, the encoder step $\theta - \eta \mathbf{g}_0$ induces the first-order change $-\eta \mathbf{g}_k^\top \mathbf{g}_0$ in objective k. We therefore regard $\mathbf{g}_0$ as compatible when $\mathbf{g}_k^\top \mathbf{g}_0 \geq 0$ for all $k \in \mathcal{K}$, such that no objective increases to first order. CACB retains the natural aggregate whenever

$$\mathbf{g}_{\mathrm{CACB}} = \mathbf{g}_0 \quad \text{if} \quad \mathbf{g}_k^\top \mathbf{g}_0 \geq 0, \quad \forall k \in \mathcal{K}. \tag{9}$$

When the natural aggregate is incompatible, CACB keeps the base coefficient fixed and redistributes the relational contribution between neighborhood attraction and nonedge separation:

$$\mathbf{g}(\delta) = \mathbf{g}_b + \beta\big[(1+\delta)\mathbf{g}_n + (1-\delta)\mathbf{g}_s\big] = \mathbf{g}_0 + \beta\delta(\mathbf{g}_n - \mathbf{g}_s), \qquad \delta \in [-1, 1]. \tag{10}$$

Here $\delta = 0$ recovers the natural aggregate, while positive and negative values shift relational emphasis toward neighborhood attraction and nonedge separation, respectively. The base coefficient remains fixed, and $(1+\delta)+(1-\delta) = 2$ preserves the total relational coefficient budget. Thus, repair is restricted to redistributing the complementary relational contribution along the direction $\mathbf{g}_n - \mathbf{g}_s$.

Under this constrained reallocation, CACB seeks the smallest change from the natural aggregate that satisfies all three compatibility constraints:

$$\delta^* = \arg\min_{\delta \in [-1,1]} \delta^2 \quad \text{s.t.} \quad \mathbf{g}_k^\top \mathbf{g}(\delta) \geq 0, \quad \forall k \in \mathcal{K}. \tag{11}$$

The resulting encoder gradient is $\mathbf{g}_{\mathrm{CACB}} = \mathbf{g}(\delta^*)$. Each compatibility constraint can be written as:

$$a_k = \mathbf{g}_k^\top \mathbf{g}_0, \qquad b_k = \beta \mathbf{g}_k^\top (\mathbf{g}_n - \mathbf{g}_s), \qquad \mathbf{g}_k^\top \mathbf{g}(\delta) = a_k + b_k \delta. \tag{12}$$

Thus, $b_k > 0$ imposes a lower bound on δ, whereas $b_k < 0$ imposes an upper bound. When $b_k = 0$, the constraint is either always satisfied if $a_k \geq 0$ or infeasible otherwise. The resulting bounds are:

$$\delta_{\min} = \max\left(\{-1\} \cup \left\{-\frac{a_k}{b_k} : b_k > 0\right\}\right), \qquad \delta_{\max} = \min\left(\{1\} \cup \left\{-\frac{a_k}{b_k} : b_k < 0\right\}\right). \tag{13}$$

If all zero-slope constraints are satisfied and $\delta_{\min} \leq \delta_{\max}$, the closest feasible reallocation has the closed-form solution:

$$\delta^* = \mathrm{clip}(0, \delta_{\min}, \delta_{\max}) = \min\{\delta_{\max}, \max\{\delta_{\min}, 0\}\}. \tag{14}$$

If no exactly compatible reallocation exists, CACB minimizes the worst normalized residual violation. Let $\mathcal{K}_+ = \{k \in \mathcal{K} : \|\mathbf{g}_k\|_2 > 0\}$. The relaxed problem is:

$$\varepsilon^* = \min_{\delta \in [-1,1]} \max_{k \in \mathcal{K}_+} \left[\frac{-a_k - b_k\delta}{\|\mathbf{g}_k\|_2}\right]_+, \qquad [x]_+ = \max(x, 0). \tag{15}$$

The positive-part operator penalizes only violated compatibility constraints, while normalization prevents gradient scale from dominating the relaxation. Among the minimizers, CACB selects the δ^* with the smallest $|\delta|$ and sets $\mathbf{g}_{\mathrm{CACB}} = \mathbf{g}(\delta^*)$. The objective is convex and piecewise linear, and an exact solution can be obtained from a finite set of candidate points, as detailed in Appendix A.3. The resulting ε^* quantifies the residual incompatibility that cannot be removed by the constrained relational reallocation.

For feasible repairs, CACB also guarantees descent on the original aggregate objective.

Theorem 1 (Aggregate Descent under Feasible Repair) *Let $F = \mathcal{L}_b + \beta(\mathcal{L}_n + \mathcal{L}_s)$. For a feasible repair, we have:*

$$\nabla_\theta F^\top \mathbf{g}_{\mathrm{CACB}} \geq \frac{\|\mathbf{g}_{\mathrm{CACB}}\|_2^2}{1 + |\delta^*|} \geq \tfrac{1}{2}\|\mathbf{g}_{\mathrm{CACB}}\|_2^2. \tag{16}$$

If $\nabla_\theta F$ is L-Lipschitz on a convex neighborhood containing θ and $\theta - \eta \mathbf{g}_{\mathrm{CACB}}$, with $L > 0$ and $0 < \eta \leq 1/(2L)$, the following descent bound holds:

$$F(\theta - \eta \mathbf{g}_{\mathrm{CACB}}) \leq F(\theta) - \tfrac{\eta}{4}\|\mathbf{g}_{\mathrm{CACB}}\|_2^2. \tag{17}$$

Proof sketch. Let $w = (1, \beta, \beta)$ and $v = (1, \beta(1+\delta^*), \beta(1-\delta^*))$. For $c = (1 + |\delta^*|)^{-1}$, $w - cv$ is componentwise nonnegative. Since feasible repair ensures $\mathbf{g}_k^\top \mathbf{g}_{\mathrm{CACB}} \geq 0$ for all $k \in \mathcal{K}$, we obtain:

$$\nabla_\theta F^\top \mathbf{g}_{\mathrm{CACB}} = c\|\mathbf{g}_{\mathrm{CACB}}\|_2^2 + \sum_{k \in \mathcal{K}} (w_k - c v_k) \mathbf{g}_k^\top \mathbf{g}_{\mathrm{CACB}} \geq c\|\mathbf{g}_{\mathrm{CACB}}\|_2^2. \tag{18}$$

Because $|\delta^*| \leq 1$, we have $c \geq 1/2$, which yields the second inequality in Eq. 16. Applying the descent lemma under the stated smoothness and step-size conditions gives Eq. 17. □

Thus, feasible repair preserves first-order compatibility with all objectives while guaranteeing descent of the original aggregate objective. Additional analysis is provided in Appendix A.4.

Table 1: Clustering performance (%) on six benchmark datasets. Mean ± standard deviation is reported. Best results are in **bold**, and second-best results are underlined.

Dataset	Metric	DGI ICLR'19	SSGC ICLR'21	S3GC NeurIPS'22	NS4GC TKDE'24	MAGI KDD'24	MS2CAG KDD'25	CoCo ICLR'26	SPGCL ICML'26	CACB Ours
Cora	ACC	72.44±2.48	65.38±1.47	70.94±2.32	74.07±1.30	73.54±0.58	70.29±0.66	74.69±0.65	74.81±0.21	**76.95±0.43**
	NMI	56.37±0.57	51.85±1.05	56.00±1.27	59.15±1.37	58.50±0.41	53.64±0.64	60.38±0.43	59.24±0.04	**61.02±0.48**
	ARI	52.22±2.68	41.25±1.69	49.26±2.97	55.03±2.09	52.04±1.08	45.76±0.98	57.63±1.29	54.83±0.12	**58.51±0.67**
	F1	70.25±2.29	65.16±1.32	69.53±1.76	72.08±1.01	72.34±0.38	69.17±0.65	73.31±0.76	72.98±0.38	**74.83±0.41**
Citeseer	ACC	64.72±0.25	62.13±2.97	53.81±0.72	69.18±0.06	59.29±0.90	61.85±0.33	69.85±1.01	68.33±0.01	**70.94±0.05**
	NMI	40.80±0.07	40.90±0.66	34.31±0.96	44.24±0.10	38.04±1.56	35.87±0.53	43.38±0.95	43.46±0.14	**46.09±0.05**
	ARI	40.49±0.10	34.75±1.97	23.98±0.45	45.36±0.11	33.49±2.62	35.22±0.51	44.20±1.13	43.63±0.10	**47.74±0.08**
	F1	61.39±0.26	58.86±2.66	51.30±0.67	64.64±0.09	55.46±1.45	58.95±0.30	61.62±1.16	64.51±0.05	**65.70±0.08**
Photo	ACC	77.01±0.13	76.34±4.04	74.13±3.62	79.21±0.03	76.83±0.08	79.13±0.05	78.26±0.26	79.09±0.04	**80.09±0.27**
	NMI	67.32±0.08	70.75±1.39	67.15±2.56	72.62±0.79	68.86±0.05	72.49±0.79	70.48±0.54	72.25±0.04	**75.49±0.20**
	ARI	57.60±0.24	59.17±3.81	57.47±3.80	62.64±0.62	58.91±0.13	63.88±0.54	60.75±0.39	61.94±0.06	**66.82±0.28**
	F1	71.58±0.09	70.17±3.78	68.39±4.25	73.73±1.11	70.93±0.09	74.94±0.95	71.82±0.37	73.05±0.07	**77.71±0.32**
Computers	ACC	44.87±1.92	55.13±0.30	59.88±0.12	57.46±0.02	54.02±1.60	49.52±0.07	60.10±0.02	54.97±0.17	**61.61±0.03**
	NMI	46.56±0.30	52.03±0.33	54.75±0.06	56.51±0.02	54.07±0.68	53.31±0.04	53.33±0.01	56.57±0.09	**56.72±0.04**
	ARI	27.05±1.14	38.76±0.18	44.07±0.10	43.98±0.04	37.89±1.82	33.61±0.02	41.13±0.01	36.49±0.13	**45.15±0.05**
	F1	36.46±0.31	44.10±2.28	48.89±0.08	47.82±0.09	43.61±2.20	40.60±0.02	49.14±0.02	49.09±0.10	**50.72±0.04**
CS	ACC	65.91±1.59	71.70±1.03	72.42±0.25	79.73±0.94	75.62±0.54	72.78±0.02	75.71±1.04	73.13±0.21	**84.98±1.09**
	NMI	71.68±0.41	75.74±0.92	71.02±0.06	80.22±0.35	79.21±0.22	75.82±0.01	78.93±0.42	80.52±0.07	**82.54±0.54**
	ARI	58.74±2.47	64.09±0.58	61.63±0.15	75.38±2.70	68.85±0.52	64.27±0.02	68.25±0.61	68.09±0.49	**82.61±1.84**
	F1	59.06±1.38	70.59±2.24	67.13±0.27	74.43±0.38	73.98±0.21	71.60±0.02	72.94±3.84	70.04±0.77	**81.10±2.83**
Physics	ACC	86.06±7.25	65.03±12.17	63.81±0.73	90.19±0.07	62.39±3.09	90.45±0.04	68.94±0.06	88.09±0.09	**93.83±0.01**
	NMI	72.93±2.36	64.49±4.20	59.63±0.10	75.35±0.07	62.26±3.18	72.45±0.03	62.17±0.10	72.47±0.11	**80.53±0.01**
	ARI	77.90±9.69	56.50±13.07	49.91±0.47	85.14±0.07	43.65±6.67	80.89±0.05	53.20±0.05	77.93±0.30	**88.16±0.01**
	F1	81.40±7.41	63.57±10.46	61.94±0.44	85.01±0.10	64.34±4.37	87.60±0.05	65.90±0.13	84.58±0.03	**91.44±0.02**

3.5 Training and Inference

During training, the three objectives are optimized jointly through the shared encoder. At each step, CACB computes their encoder gradients and supplies $\mathbf{g}_{\mathrm{CACB}}$ to update the encoder parameters θ. The projection heads $\mathbf{P}_n$ and $\mathbf{P}_s$ are updated through $\mathcal{L}_n$ and $\mathcal{L}_s$, respectively, and their gradients are not included in the compatibility repair. Thus, CACB coordinates objective interaction only at the shared encoder while retaining objective-specific relational projection spaces. At inference, the original input graph is passed through the shared encoder to obtain $\mathbf{H} = f_\theta(\mathbf{A}, \mathbf{X})$, on which K-means is applied to produce the final clustering assignments.

4 Experiments

4.1 Experimental Setup

We compare against eight representative attributed graph clustering methods: DGI (Veličković et al., 2019), SSGC (Zhu & Koniusz, 2021), S3GC (Devvrit et al., 2022), NS4GC (Liu et al., 2024c), MAGI (Liu et al., 2024b), MS2CAG (Lin et al., 2025), CoCo (Ju et al., 2026), and SPGCL (Shan et al., 2026).

CACB is implemented within the PyAGC benchmark framework (Liu et al., 2026) following its unified preprocessing and evaluation protocol. Each experiment is repeated five times with different random initializations, and the mean results with standard deviations are reported. Ground-truth labels are used only for clustering evaluation. We report clustering accuracy (ACC), normalized mutual information (NMI), adjusted Rand index (ARI), and F1. Detailed implementation settings and hyperparameter configurations are provided in Appendix C.

4.2 Experimental Results

Comparison with state-of-the-art methods. Table 1 reports the clustering results on six benchmarks, from which two observations can be made. First, methods incorporating richer or multiple

Table 2: Ablation study on four representative datasets. Mean ± standard deviation is reported; best results are in **bold**.

	Cora		Photo		CS		Physics	
Variant	**NMI**	**F1**	**NMI**	**F1**	**NMI**	**F1**	**NMI**	**F1**
w/o L_n	55.34±0.03	62.53±0.15	70.00±0.16	76.47±0.14	72.90±1.30	67.90±4.08	68.88±0.03	67.14±0.02
w/o L_s	53.79±0.22	65.09±0.97	59.04±2.34	65.80±0.73	73.50±1.45	64.14±3.24	68.89±0.22	81.37±0.28
w/o Proj.	61.00±0.93	74.57±0.03	74.73±1.23	76.78±2.84	81.10±0.17	74.88±0.18	76.33±0.06	86.62±0.08
w/o CACB	59.73±0.21	73.24±0.11	67.96±0.87	71.72±1.93	80.28±0.73	71.20±4.48	77.47±0.01	79.75±0.00
CACB	**61.02±0.48**	**74.83±0.41**	**75.49±0.20**	**77.71±0.32**	**82.54±0.54**	**81.10±2.83**	**80.53±0.01**	**91.44±0.02**

Table 3: Comparison of optimization strategies on four representative datasets. Mean ± standard deviation is reported; best results are in **bold**.

	Cora		Photo		CS		Physics	
Strategy	**NMI**	**F1**	**NMI**	**F1**	**NMI**	**F1**	**NMI**	**F1**
Fixed	59.73±0.21	73.24±0.11	67.96±0.87	71.72±1.93	80.28±0.73	71.20±4.48	77.47±0.01	79.75±0.00
MGDA	60.35±0.13	73.65±0.07	70.59±0.09	72.79±1.83	80.56±0.43	73.28±3.16	78.53±0.02	84.95±0.03
PCGrad	60.43±0.12	74.37±0.08	72.99±0.01	73.25±0.00	81.37±0.82	76.47±2.97	80.02±0.03	90.76±0.04
CACB	**61.02±0.48**	**74.83±0.41**	**75.49±0.20**	**77.71±0.32**	**82.54±0.54**	**81.10±2.83**	**80.53±0.01**	**91.44±0.02**

supervisory signals generally outperform approaches driven by a single learning objective, indicating the benefit of exploiting complementary structural and semantic information for graph clustering. Second, CACB achieves the best mean performance across all dataset–metric combinations, with particularly clear gains on the co-authorship networks. This suggests that simply introducing complementary supervision is not sufficient; coordinating how these objectives jointly optimize the shared representation can provide further improvements. By extending complementary supervision from objective design to gradient coordination, CACB yields more consistent gains across different graph domains.

Ablation study. We evaluate four variants of CACB to examine its main components. The w/o L_n and w/o L_s variants remove neighborhood attraction and nonedge separation, respectively, to assess whether the two relational signals provide complementary supervision rather than being dominated by either objective. The w/o Proj. variant applies both relational objectives directly to the shared representation, isolating the effect of objective-specific projection spaces. Finally, w/o CACB retains the same objectives and projections but disables compatibility repair, evaluating the contribution of gradient coordination itself.

Table 2 reveals three observations. First, removing either relational objective consistently degrades clustering performance, while their relative importance varies across datasets. This suggests that neighborhood attraction and nonedge separation provide complementary rather than interchangeable supervision. Second, removing the objective-specific projections also leads to consistent deterioration, indicating that separating the two relational signals in learned similarity spaces is beneficial. More importantly, disabling CACB while retaining the same objectives and projections still produces a clear performance drop. This shows that the gains cannot be attributed solely to richer supervision or additional projection heads; explicitly coordinating their interaction at the shared encoder provides an additional benefit. Full results for all datasets and metrics are reported in Table 7.

Comparison of optimization strategies. To evaluate the optimization design, we compare CACB with three alternative strategies. Fixed directly uses the natural aggregate $\mathbf{g}_0$, while MGDA searches for a Pareto-oriented gradient combination (Sener & Koltun, 2018) and PCGrad resolves conflicting gradients through pairwise projection (Yu et al., 2020). In contrast, CACB performs AGC-specific compatibility repair by preserving the consistency anchor and minimally redistributing the attraction–separation pair when incompatibility arises.

Table 3 reveals a consistent performance ordering across all reported settings: Fixed $<$ MGDA $<$ PCGrad $<$ CACB. The improvements of MGDA and PCGrad over Fixed indicate that explicitly handling interactions among objective gradients is beneficial for AGC, while the stronger results of PCGrad further highlight the importance of conflict-aware optimization. CACB consistently advances beyond both generic strategies, suggesting that gradient coordination benefits from respecting

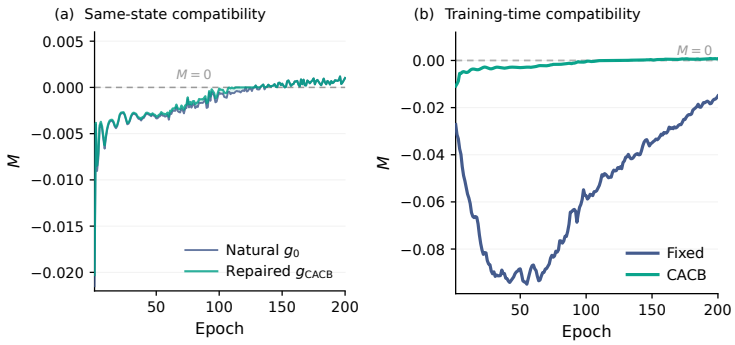


Figure 3: **Gradient compatibility analysis on CS.** (a) Same-state compatibility margins of the natural gradient $\mathbf{g}_0$ and repaired gradient $\mathbf{g}_{\mathrm{CACB}}$. (b) Training-time compatibility trajectories of Fixed and CACB over the first 200 epochs.

the specific structure of complementary AGC objectives. In particular, CACB evaluates the natural aggregate against all objectives and repairs incompatibility through constrained redistribution of the attraction–separation pair, rather than freely recombining or independently projecting objective gradients.

Gradient compatibility analysis. To directly examine how CACB affects gradient compatibility and how this effect evolves during training, we track the compatibility of the natural and repaired gradients. For $\mathcal{K}_+ = \{k : \|\mathbf{g}_k\|_2 > 0\}$ and a nonzero candidate gradient $\mathbf{u}$, we define the compatibility margin as:

$$M(\mathbf{u}) = \min_{k \in \mathcal{K}_+} \frac{\mathbf{g}_k^\top \mathbf{u}}{\|\mathbf{g}_k\|_2 \|\mathbf{u}\|_2}, \qquad M_{\mathrm{before}} = M(\mathbf{g}_0), \quad M_{\mathrm{after}} = M(\mathbf{g}_{\mathrm{CACB}}). \tag{19}$$

A negative margin indicates violation of at least one first-order compatibility constraint. For feasible repair, CACB guarantees $M_{\mathrm{after}} \geq 0$, while the relaxed repair reduces residual incompatibility when exact feasibility is unavailable. Since CACB selects the minimum-deviation feasible reallocation, an active repair typically reaches the boundary $M = 0$.

Figure 3 examines the first 200 training epochs from two complementary perspectives. Figure 3(a) compares $\mathbf{g}_0$ and $\mathbf{g}_{\mathrm{CACB}}$ at the same model state. Early in training, CACB reduces the incompatibility of the natural aggregate through relaxed repair; once exact repair becomes feasible, the repaired margin reaches the boundary $M = 0$, and the two trajectories eventually coincide as the natural aggregate itself becomes compatible. This progression reflects the minimum-intervention design, where repair is applied only to the extent required by compatibility. Figure 3(b) further shows the accumulated effect during training: CACB remains close to the compatibility boundary and subsequently enters the compatible region, whereas Fixed remains incompatible throughout the displayed interval. Together, these results connect the immediate effect of gradient repair with its longer-term influence on the optimization trajectory.

Case study of representation evolution. Figure 4 traces the representation evolution of node 16627 from the robotics class on CS within its local neighborhood. For visualization, Fixed and CACB embeddings across all displayed checkpoints are projected using a single jointly fitted two-component PCA, with the raw-feature starting point aligned to this shared 2D space. In the raw feature space, the target lies in an ambiguous region surrounded by nodes from multiple classes. During training, multiple supervisory signals impose different geometric pressures on the same shared representation. Under Fixed aggregation, conflicts among these signals can produce competing updates that keep the target trapped in an ambiguous local region rather than allowing it to move decisively toward a coherent cluster. In contrast, CACB coordinates their interaction during shared-encoder optimization, gradually moving the target toward a more coherent robotics region while increasing its separation from nearby competing classes. This case illustrates how compatibility-aware coordination can reshape the evolution of a locally ambiguous representation rather than merely affect its final position.

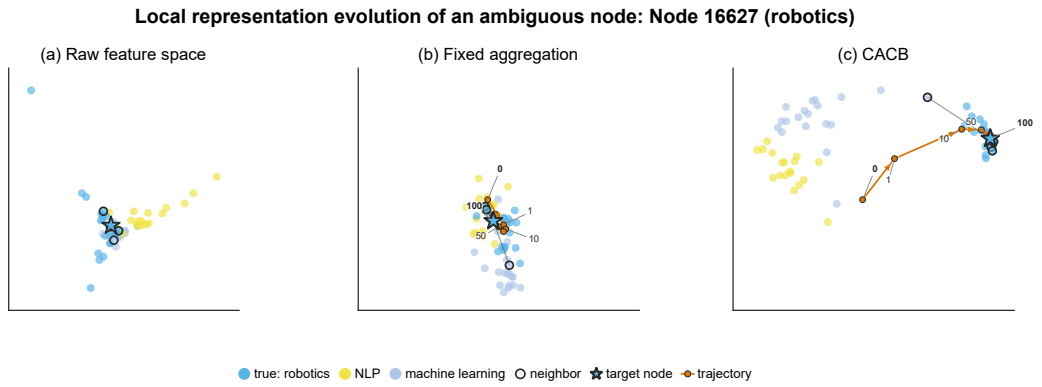


Figure 4: **Local representation evolution of node 16627 on CS. (a)** Raw feature space. **(b)** Fixed aggregation. **(c)** CACB.

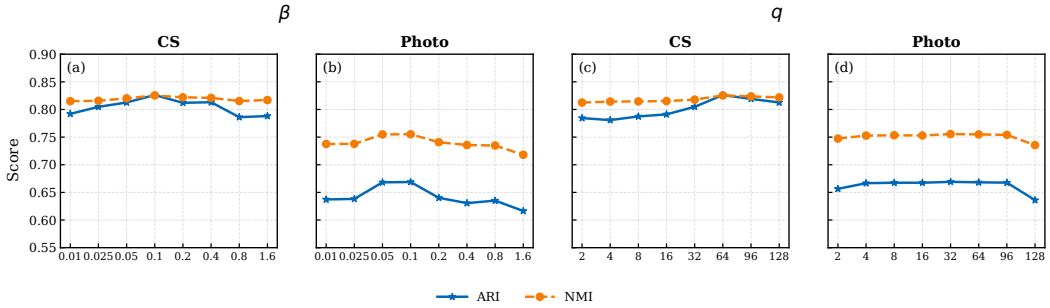


Figure 5: Parameter sensitivity of CACB on CS and Photo. Panels (a)–(b) vary the relational supervision weight β, while Panels (c)–(d) vary the nonedge sampling budget q.

Parameter sensitivity. We investigate the sensitivity of CACB to the relational supervision weight β and the nonedge sampling budget q. Figure 5 reports the results on CS and Photo, while Appendix D.3 provides results on the remaining four datasets. CACB performs well over a moderate range of β, with the strongest results generally around $\beta = 0.05$–0.1. Performance decreases when β becomes overly large, particularly on Photo. The results are relatively stable across a broad range of q, with noticeable degradation mainly at the largest sampling budget on Photo. Overall, CACB shows reasonable robustness to both hyperparameters. We further provide computational efficiency analyses and representation visualizations in Appendices D.1 and D.4, respectively.

5 Conclusion

In this paper, we propose Conflict-Aware Complementary Balancing (CACB) for attributed graph clustering. CACB evaluates the natural aggregate encoder gradient through objective-wise compatibility constraints. Once incompatibility is detected, it minimally redistributes the relational gradient contributions while keeping the base coefficient and total relational budget fixed. The resulting repair restores joint first-order compatibility when feasible and provides quantitative guarantees on aggregate descent. Extensive experiments on six benchmark datasets demonstrate the effectiveness of CACB across different graph domains. Further ablation studies, optimization comparisons and trajectory analyses validate the contributions of complementary supervision and compatibility-aware gradient coordination. Overall, CACB extends complementary supervision in AGC from objective design to explicit gradient coordination, offering a principled way to regulate how multiple supervisory signals jointly optimize the shared representation. Future work will explore extending CACB to more complex settings such as multi-view and temporal graphs, where coordinating heterogeneous supervisory signals poses additional challenges.

AI USE STATEMENT

Generative AI tools were used for language polishing and readability improvements, as well as for retrieval and discovery of potentially relevant literature. They were not used for any tasks requiring disclosure under the ICLR 2027 policy. All AI-assisted edits were reviewed by the authors, and any literature identified with AI assistance was independently verified before inclusion. The authors take responsibility for the final content of this work.

REFERENCES

Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *Proceedings of the web conference 2020*, pp. 1400–1410, 2020.

Fnu Devvrit, Aditya Sinha, Inderjit Dhillon, and Prateek Jain. S3gc: Scalable self-supervised graph clustering. *Advances in Neural Information Processing Systems*, 35:3248–3261, 2022.

Karish Grover, Theodore Vasiloudis, Han Xie, Sixing Lu, Xiang Song, and Christos Faloutsos. Feedback control for multi-objective graph self-supervision. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=EuTiwQ5bgj.

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. URL https://openreview.net/forum?id=9jdQLmPUHW.

Xinying Lai, Dingming Wu, Christian S Jensen, and Kezhong Lu. A re-evaluation of deep learning methods for attributed graph clustering. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 1168–1177, 2023.

Wen-Zhi Li, Chang-Dong Wang, Hui Xiong, and Jian-Huang Lai. Homogcl: Rethinking homophily in graph contrastive learning. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1341–1352, 2023.

Yan Li and Xiaoyun Chen. Attributed graph subspace clustering with residual compensation guided by adaptive dual manifold regularization. *Expert Systems with Applications*, 255:124699, 2024.

Xiaoyang Lin, Renchi Yang, Haoran Zheng, and Xiangyu Ke. Spectral subspace clustering for attributed graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 789–799, 2025.

Hongtao Liu, Xianbin Lu, Kefei Cheng, and Xueyan Liu. A multi-embedding fusion network for attributed graph clustering. *Applied Soft Computing*, 165:112073, 2024a.

Yunfei Liu, Jintang Li, Yuehe Chen, Ruofan Wu, Ericbk Wang, Jing Zhou, Sheng Tian, Shuheng Shen, Xing Fu, Changhua Meng, et al. Revisiting modularity maximization for graph clustering: A contrastive learning perspective. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1968–1979, 2024b.

Yunhui Liu, Xinyi Gao, Tieke He, Tao Zheng, Jianhua Zhao, and Hongzhi Yin. Reliable node similarity matrix guided contrastive graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):9123–9135, 2024c.

Yunhui Liu, Yue Liu, Yongchao Liu, Tao Zheng, Stan Z Li, Xinwang Liu, and Tieke He. Beyond the academic monoculture: A unified framework and industrial perspective for attributed graph clustering. *arXiv preprint arXiv:2603.20829*, 2026.

Cuihua Ma, Chaosheng Tang, Ziqi Deng, Fuyun Li, Haixia Long, and Jieren Cheng. Prototype based contrastive graph clustering network for reducing false negatives. *Scientific Reports*, 15(1): 35584, 2025.

Liang Peng, Yixuan Ye, Cheng Liu, Hangjun Che, Man-Fai Leung, Si Wu, and Hau-San Wong. Trustworthy neighborhoods mining: Homophily-aware neutral contrastive learning for graph clustering. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Lianze Shan, Ningchong Wang, Jitao Zhao, Di Jin, and Dongxiao He. Revisiting positive samples in graph contrastive learning: From the perspective of message passing. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YkUsqcPYkU.

Wenxuan Tu, Sihang Zhou, Xinwang Liu, Xifeng Guo, Zhiping Cai, En Zhu, and Jieren Cheng. Deep fusion clustering network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9978–9987, 2021.

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rklz9iAcKQ.

Chun Wang, Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, and Chengqi Zhang. Attributed graph clustering: a deep attentional embedding approach. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 3670–3676, 2019.

Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 10834–10842, 2023.

Xihong Yang, Erxue Min, Ke Liang, Yue Liu, Siwei Wang, Sihang Zhou, Huijun Wu, Xinwang Liu, and En Zhu. Graphlearner: Graph node clustering with fully learnable augmentation. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 5517–5526, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Shuai Zhang, Shan Yang, Wenyu Zhang, Jiahao Nie, and Shan Ji. G2cl: Gradient-guided graph contrastive learning for eliminating the message contrastive conflict. *Neural Networks*, pp. 108605, 2026.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering*, 34(12):5586–5609, 2021.

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. *Advances in Neural Information Processing Systems*, 38: 115969–115994, 2026a.

Weizhi Zhao, Heng Liu, Shudong Hou, and Feng Chen. Topology-aware contrastive learning for attributed graph clustering. *Neurocomputing*, pp. 133213, 2026b.

Yimei Zheng, Caiyan Jia, and Jian Yu. Attributed graph clustering under the contrastive mechanism with cluster-preserving augmentation. *Information Sciences*, 681:121225, 2024.

Hao Zhu and Piotr Koniusz. Simple spectral graph convolution. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=CYO5T-YjWZV.

Pengfei Zhu, Jialu Li, Yu Wang, Bin Xiao, Shuai Zhao, and Qinghua Hu. Collaborative decision-reinforced self-supervision for attributed graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10851–10863, 2022.

Pengfei Zhu, Qian Wang, Yu Wang, Jialu Li, and Qinghua Hu. Every node is different: Dynamically fusing self-supervised tasks for attributed graph clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17184–17192, 2024.

A Additional Analysis

This appendix provides additional analysis for the two main components of CACB. We first characterize the objective-specific relational projections introduced in Section 3.3. We then analyze the minimum-intervention property, relaxed repair, and compatibility formulation in Section 3.4.

A.1 Projection-Induced Relational Geometries

Section 3.3 uses separate projection heads for neighborhood attraction and nonedge separation. The following result clarifies the precise form of this separation.

Proposition 1 (Separate parameterization of relational geometries) *For each $k \in \{n, s\}$, let $\mathbf{Z}_k = \mathbf{H}\mathbf{P}_k$ and $\mathbf{M}_k = \mathbf{P}_k\mathbf{P}_k^\top$. Then $\mathbf{M}_k \succeq 0$ and, whenever the projected vectors have nonzero norm,*

$$s(\mathbf{h}_i\mathbf{P}_k, \mathbf{h}_j\mathbf{P}_k) = \frac{\mathbf{h}_i\mathbf{M}_k\mathbf{h}_j^\top}{\sqrt{\mathbf{h}_i\mathbf{M}_k\mathbf{h}_i^\top}\sqrt{\mathbf{h}_j\mathbf{M}_k\mathbf{h}_j^\top}}. \tag{20}$$

Moreover, the two relational losses are directly parameterized by disjoint projection heads:

$$\nabla_{\mathbf{P}_s}\mathcal{L}_n = \mathbf{0}, \qquad \nabla_{\mathbf{P}_n}\mathcal{L}_s = \mathbf{0}. \tag{21}$$

They nevertheless remain coupled through the shared representation because

$$\nabla_{\mathbf{H}}\mathcal{L}_k = \nabla_{\mathbf{Z}_k}\mathcal{L}_k\mathbf{P}_k^\top, \qquad k \in \{n, s\}. \tag{22}$$

Proof. For any row vector $\mathbf{x}$,

$$\mathbf{x}\mathbf{M}_k\mathbf{x}^\top = \mathbf{x}\mathbf{P}_k\mathbf{P}_k^\top\mathbf{x}^\top = \|\mathbf{x}\mathbf{P}_k\|_2^2 \geq 0,$$

so $\mathbf{M}_k$ is positive semidefinite. Substituting $\mathbf{z}_{k,i} = \mathbf{h}_i\mathbf{P}_k$ into cosine similarity directly gives Eq. 20. Since $\mathcal{L}_n$ depends on $\mathbf{P}_n$ but not on $\mathbf{P}_s$, and $\mathcal{L}_s$ depends on $\mathbf{P}_s$ but not on $\mathbf{P}_n$, Eq. 21 follows. Applying the chain rule through $\mathbf{Z}_k = \mathbf{H}\mathbf{P}_k$ gives Eq. 22. □

Proposition 1 shows that the two heads separately parameterize the similarity geometries used by neighborhood attraction and nonedge separation, rather than forcing both relational signals to operate under the same metric. Their interaction is therefore removed at the projection-head parameterization level but remains present at the shared encoder, which motivates the gradient-level coordination in Section 3.4.

A.2 Minimum-Intervention Property and Degenerate Cases

The repair in Section 3.4 minimizes δ^2. We next show that this criterion corresponds directly to minimum deviation from the natural coefficient allocation and, except for a degenerate direction, minimum change of the aggregate gradient.

Let

$$\mathbf{c}(\delta) = \big(1, \beta(1+\delta), \beta(1-\delta)\big)$$

denote the coefficient vector after repair, and let $\mathbf{c}_0 = (1, \beta, \beta)$ denote the natural coefficients. Then

$$\|\mathbf{c}(\delta) - \mathbf{c}_0\|_2^2 = 2\beta^2\delta^2. \tag{23}$$

The corresponding gradient deviation satisfies

$$\|\mathbf{g}(\delta) - \mathbf{g}_0\|_2^2 = \beta^2\delta^2\|\mathbf{g}_n - \mathbf{g}_s\|_2^2. \tag{24}$$

Therefore, over any nonempty feasible interval, the solution in Eq. 14 uniquely minimizes coefficient deviation from the natural allocation. When $\mathbf{g}_n \neq \mathbf{g}_s$, it also uniquely minimizes the Euclidean distance between the repaired gradient and $\mathbf{g}_0$ over the same feasible set. This follows because both quantities are positive multiples of δ^2, whose unique minimizer on a closed interval is the projection of zero onto that interval.

Several degenerate cases follow directly. If $\mathbf{g}_n = \mathbf{g}_s$, then $\mathbf{g}(\delta) = \mathbf{g}_0$ for every δ, so relational redistribution cannot alter the encoder gradient and the minimum-intervention tie-break selects $\delta = 0$. If an individual objective gradient is zero, its compatibility constraint is automatically satisfied and it is omitted from normalized relaxation terms to avoid division by zero. If all three gradients vanish, CACB returns the zero gradient. The formulation assumes $\beta > 0$; when relational supervision is disabled, the aggregate reduces to $\mathbf{g}_b$ and no relational repair is required.

A.3 Exact Solution of the Minimum Compatibility Relaxation

When the compatibility interval is empty, Section 3.4 minimizes the worst normalized residual violation. For each $k \in \mathcal{K}_+$, define

$$\ell_k(\delta) = \frac{-a_k - b_k\delta}{\|\mathbf{g}_k\|_2} = u_k + v_k\delta, \qquad u_k = -\frac{a_k}{\|\mathbf{g}_k\|_2}, \qquad v_k = -\frac{b_k}{\|\mathbf{g}_k\|_2}, \tag{25}$$

and include $\ell_0(\delta) = 0$. The relaxation objective can then be written as

$$\phi(\delta) = \max_{j \in \{0\}\cup\mathcal{K}_+} \ell_j(\delta), \qquad \varepsilon^* = \min_{\delta\in[-1,1]} \phi(\delta). \tag{26}$$

Because ϕ is the pointwise maximum of finitely many affine functions, it is continuous, convex, and piecewise linear. Between two consecutive intersections of the affine functions, the active maximum is affine; hence its minimum over that segment is attained at an endpoint or throughout a flat segment. Consequently, at least one minimizer belongs to the finite candidate set

$$\mathcal{S} = \{-1, 1\} \cup \left\{ \frac{u_j - u_i}{v_i - v_j} : v_i \neq v_j,\ -1 \leq \frac{u_j - u_i}{v_i - v_j} \leq 1 \right\}, \tag{27}$$

with $i, j \in \{0\} \cup \mathcal{K}_+$. Thus,

$$\varepsilon^* = \min_{\delta\in\mathcal{S}} \phi(\delta). \tag{28}$$

For the resulting optimum, $\phi(\delta) \leq \varepsilon^*$ is equivalent to

$$a_k + b_k\delta + \varepsilon^*\|\mathbf{g}_k\|_2 \geq 0, \qquad k \in \mathcal{K}_+. \tag{29}$$

Applying the same interval construction as in Eq. 13 to these relaxed constraints yields the interval of optimal reallocations, denoted by $\mathcal{I}_{\varepsilon^*}$. CACB projects zero onto this interval, thereby selecting the smallest $|\delta|$ among all minimizers of the worst residual violation.

The selected relaxed gradient satisfies

$$\mathbf{g}_k^\top \mathbf{g}_{\mathrm{CACB}} \geq -\varepsilon^*\|\mathbf{g}_k\|_2, \qquad k \in \mathcal{K}_+. \tag{30}$$

Hence ε^* directly quantifies the residual first-order incompatibility that cannot be removed by the constrained relational reallocation.

A.4 Relaxed Aggregate Progress and Realized Displacements

Theorem 1 establishes aggregate progress for an exactly feasible repair. We first extend the result to the relaxed case.

For a selected reallocation δ, let

$$\mathbf{u} = \mathbf{g}(\delta), \qquad w = (1, \beta, \beta), \qquad v = (1, \beta(1+\delta), \beta(1-\delta)),$$

and define

$$c = \frac{1}{1+|\delta|}, \qquad B = \sum_{k\in\mathcal{K}} (w_k - cv_k)\|\mathbf{g}_k\|_2. \tag{31}$$

Since $\delta \in [-1, 1]$, we have $c \in [1/2, 1]$ and $w_k - cv_k \geq 0$ for every k, so $B \geq 0$.

Suppose

$$\mathbf{g}_k^\top \mathbf{u} \geq -\varepsilon\|\mathbf{g}_k\|_2, \qquad k \in \mathcal{K}, \tag{32}$$

where $\varepsilon = 0$ for an exactly feasible repair and $\varepsilon = \varepsilon^*$ for the minimum-relaxation case. Because $\mathbf{g}_0 = \sum_k w_k\mathbf{g}_k$ and $\mathbf{u} = \sum_k v_k\mathbf{g}_k$,

$$\begin{aligned} \mathbf{g}_0^\top \mathbf{u} &= c\|\mathbf{u}\|_2^2 + \sum_{k\in\mathcal{K}} (w_k - cv_k)\mathbf{g}_k^\top \mathbf{u} \\ &\geq c\|\mathbf{u}\|_2^2 - \varepsilon B. \end{aligned} \tag{33}$$

For $\varepsilon = 0$, Eq. 33 reduces to the feasible-repair result in Theorem 1. For the relaxed case, the term εB explicitly measures the loss of aggregate alignment caused by residual incompatibility.

If the aggregate objective $F = \mathcal{L}_b + \beta(\mathcal{L}_n + \mathcal{L}_s)$ has an L-Lipschitz gradient on the relevant segment, the descent lemma further gives

$$F(\theta - \eta\mathbf{u}) \leq F(\theta) - \eta(c - L\eta/2)\|\mathbf{u}\|_2^2 + \eta\varepsilon B. \tag{34}$$

Thus, the feasible case recovers the descent guarantee in the main text, whereas relaxed repair introduces an explicit residual term determined by ε.

Compatibility under an optimizer-transformed displacement. The compatibility conditions in Section 3.4 concern the gradient supplied to the encoder optimizer. To characterize a realized descent vector that may deviate from this direction, write

$$\mathbf{q} = \gamma \mathbf{u} + \mathbf{e}, \qquad \gamma > 0, \tag{35}$$

where $\mathbf{e}$ denotes the deviation from a scaled CACB direction. For every nonzero objective gradient,

$$\frac{\mathbf{g}_k^\top \mathbf{q}}{\|\mathbf{g}_k\|_2} \geq \gamma \frac{\mathbf{g}_k^\top \mathbf{u}}{\|\mathbf{g}_k\|_2} - \|\mathbf{e}\|_2. \tag{36}$$

This follows immediately from Cauchy–Schwarz. Consequently, if

$$m = \min_{k \in \mathcal{K}_+} \frac{\mathbf{g}_k^\top \mathbf{u}}{\|\mathbf{g}_k\|_2} > 0,$$

compatibility is preserved for the realized vector whenever $\|\mathbf{e}\|_2 \leq \gamma m$. For a boundary repair with $m = 0$, preservation depends directly on the displacement deviation. This result distinguishes the compatibility guarantee of the supplied CACB gradient from transformations introduced by the optimizer.

A.5 Why the Complete Aggregate Must Be Tested

Pairwise agreement between the two relational gradients does not determine whether the complete encoder gradient is compatible with all objectives. This can be seen from two simple examples.

First, let $\beta = 1$, $\mathbf{g}_n = (1, 0)$, $\mathbf{g}_s = (-1, 1)$, and $\mathbf{g}_b = (0, 1)$. Although $\mathbf{g}_n^\top \mathbf{g}_s = -1 < 0$, the natural aggregate is $\mathbf{g}_0 = (0, 2)$ and satisfies

$$\mathbf{g}_b^\top \mathbf{g}_0 \geq 0, \qquad \mathbf{g}_n^\top \mathbf{g}_0 \geq 0, \qquad \mathbf{g}_s^\top \mathbf{g}_0 \geq 0.$$

Thus, relational disagreement alone does not imply that the full aggregate requires repair.

Conversely, let $\mathbf{g}_n = (1, 0)$, $\mathbf{g}_s = (0.5, 1)$, and $\mathbf{g}_b = (-1.7, 0)$. Here $\mathbf{g}_n^\top \mathbf{g}_s = 0.5 > 0$, yet $\mathbf{g}_0 = (-0.2, 1)$ gives $\mathbf{g}_n^\top \mathbf{g}_0 = -0.2 < 0$. The natural aggregate is therefore incompatible despite pairwise relational agreement. In this example, $\delta = 0.4$ yields $\mathbf{g}(\delta) = (0, 0.6)$, which satisfies all three compatibility constraints.

These examples show that the base contribution can change the compatibility of the aggregate encoder gradient in either direction. CACB therefore tests the aggregate against all three objective gradients rather than using pairwise relational agreement as the repair criterion.

B Training Algorithm

Algorithm 1: CACB training step

Input: graph, encoder θ, heads $\mathbf{P}_n, \mathbf{P}_s$, relational weight β.

1. Sample two augmented views and nonedges; evaluate $\mathcal{L}_b, \mathcal{L}_n, \mathcal{L}_s$.
2. Compute the three encoder gradients and $\mathbf{g}_0$ in Eq. 8.
3. If $a_k = \mathbf{g}_k^\top \mathbf{g}_0 \geq 0$ for every k, set $\delta^* = 0$.
4. Otherwise, construct $\mathcal{I}$ using Eqs. 12–13. If nonempty, clip zero to $\mathcal{I}$; if empty, solve Eq. 15 and clip zero to $\mathcal{I}_{\varepsilon^*}$.
5. Supply $\mathbf{g}_{\mathrm{CACB}} = \mathbf{g}_0 + \beta\delta^*(\mathbf{g}_n - \mathbf{g}_s)$ to the encoder optimizer. Update each projection head through its corresponding relational objective; head gradients are excluded from the compatibility test.

Inference: encode the original input graph and apply K-means to $\mathbf{H}$.

The scalar decision uses current encoder gradients and is not differentiated through. The base participates in compatibility constraints while its coefficient remains fixed.

Table 4: Statistics of the datasets used in the experiments. Edge entries are counted after preprocessing; all processed graphs are loop-free and bidirectional, so each undirected edge contributes two entries.

Dataset	Nodes	Edge Entries	Features	Classes	Type
Cora	2,708	10,556	1,433	7	Citation
Citeseer	3,327	9,104	3,703	6	Citation
Photo	7,650	238,162	745	8	Co-purchase
Computers	13,752	491,722	767	10	Co-purchase
CS	18,333	163,788	6,805	15	Co-authorship
Physics	34,493	495,924	8,415	5	Co-authorship

Table 5: Summary of the compared baselines.

Method	Category	Main mechanism
DGI (Veličković et al., 2019)	Self-supervised representation learning	Mutual-information maximization
SSGC (Zhu & Koniusz, 2021)	Spectral graph filtering	Simplified spectral propagation
S3GC (Devvrit et al., 2022)	Self-supervised graph clustering	Scalable graph contrastive learning
NS4GC (Liu et al., 2024c)	Contrastive graph clustering	Reliable node-similarity supervision
MAGI (Liu et al., 2024b)	Contrastive graph clustering	Modularity-aware contrastive learning
MS2CAG (Lin et al., 2025)	Spectral subspace clustering	Spectral subspace learning
CoCo (Ju et al., 2026)	Deep graph clustering	Compactness and consistency learning
SPGCL (Shan et al., 2026)	Graph contrastive learning	Positive-sample-aware supervision

C Experimental Details

C.1 Dataset Details

We evaluate CACB on six widely used attributed graph clustering benchmarks following the unified preprocessing and evaluation protocol of PyAGC (Liu et al., 2026). Cora and Citeseer are citation networks, Photo and Computers are Amazon co-purchase networks, and CS and Physics are co-authorship networks. Table 4 summarizes the statistics of the processed graphs used in our experiments. Ground-truth labels are used only for clustering evaluation.

C.2 Baseline Details

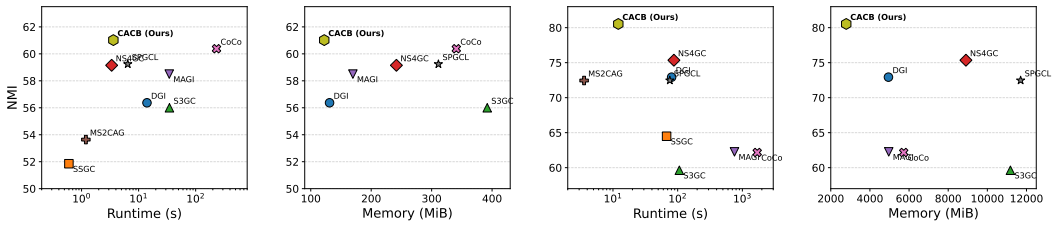
We compare CACB with eight representative graph representation and clustering methods covering spectral filtering, self-supervised representation learning, contrastive graph clustering, spectral subspace learning, and recent clustering-oriented objectives. Table 5 summarizes the main characteristics of the compared methods.

C.3 Implementation Details

All experiments are implemented within the PyAGC framework (Liu et al., 2026) following its unified preprocessing and evaluation protocol. For the baseline methods, we follow the implementations and experimental settings provided by PyAGC. For CACB, we follow the experimental settings of NS4GC (Liu et al., 2024c), using full-batch training with a two-layer GCN encoder. Here, p_m and p_d denote the feature-masking and edge-dropping rates, respectively, and d_{hid} denotes the hidden dimension of the GCN encoder. The learning rate and weight decay are set to 1×10^{-3} and 1×10^{-5}, respectively. All experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 3090 GPU (24 GB VRAM), a 12th Gen Intel Core i9-12900 CPU, and 32 GB RAM, using PyTorch 2.6.0 with CUDA 12.4. The two objective-specific projection heads are bias-free linear layers with $d_p = d/2$. The similarity margin m and temperature τ are fixed to 0.2 and 0.1, respectively. At each training step, qN directed nonedge pairs are randomly sampled after excluding observed edges and self-pairs. For a comprehensive overview of dataset-specific hyperparameters, please refer to Table 6.

Table 6: Hyperparameter settings of CACB on six benchmark datasets.

Dataset	p_m	p_d	q	β	d_{hid}	Epochs
Cora	0.1	0.2	16	0.10	64	200
Citeseer	0.0	0.8	32	0.20	256	50
Photo	0.1	0.8	64	0.05	256	200
Computers	0.1	0.5	16	0.10	256	400
CS	0.1	0.2	64	0.10	128	300
Physics	0.0	0.8	32	0.10	64	100



(a) Cora-Runtime (b) Cora-Memory (c) Physics-Runtime (d) Physics-Memory

Figure 6: **Performance–efficiency comparison on Cora and Physics.** Clustering performance is measured by NMI. Higher NMI together with lower runtime and GPU memory consumption is preferred. SSGC and MS2CAG do not require GPU computation and are therefore omitted from the memory panels.

D Additional Experimental Results

D.1 Computational Efficiency

We further evaluate the computational efficiency of CACB against the compared baselines on Cora and Physics, representing a commonly used small-scale benchmark and the largest graph in our experiments, respectively. Figure 6 compares clustering performance with training runtime and peak GPU memory consumption.

As shown in Fig. 6, CACB achieves a favorable performance–efficiency trade-off on both datasets. On Cora, CACB attains the highest NMI while requiring only 3.61 s of runtime and 122.3 MiB of peak GPU memory, the lowest memory consumption among the GPU-based methods. The advantage becomes more evident on Physics, where CACB achieves the highest NMI of 80.53 with a runtime of 12.09 s and peak GPU memory of 2789.8 MiB, giving a favorable empirical trade-off against most compared methods. These measurements include gradient evaluation and repair; the closed-form scalar decision alone should not be equated with total training cost.

D.2 Ablation Study

As shown in Table 7, CACB achieves the best performance in all 24 dataset–metric settings. Removing either $\mathcal{L}_n$ or $\mathcal{L}_s$ causes substantial and dataset-dependent degradation, reflecting their complementary relational roles. Removing the objective-specific projections also consistently reduces performance. Finally, w/o CACB retains both relational objectives and their projections but directly supplies the unrepaired natural gradient $\mathbf{g}_0$, leading to lower performance across all reported settings. This comparison isolates the additional contribution of compatibility-aware repair beyond relational supervision and projection design.

D.3 Additional Parameter Sensitivity Results

For completeness, we report sensitivity results on the remaining four datasets. Figures 7 and 8 show the effects of β and q. Changing β affects both the natural aggregate and the fixed relational budget 2β, while changing q affects the sampled separation signal and its encoder gradient.

Table 7: Ablation study on six benchmark datasets. Mean ± standard deviation is reported. Best results are in **bold**, and second-best results are underlined.

Metric	Variant	Cora	Citeseer	Photo	Computers	CS	Physics
ACC	w/o L_n	67.25±0.22	68.33±0.15	78.58±0.12	47.80±0.06	68.81±3.52	73.90±0.02
	w/o L_s	65.32±0.35	56.75±2.57	69.52±1.94	32.23±0.60	65.93±3.09	86.49±0.20
	w/o Proj.	76.18±0.92	69.94±0.04	79.81±0.11	59.34±1.95	80.79±0.53	91.06±0.04
	w/o CACB	75.30±0.16	68.98±0.01	76.30±0.62	59.70±0.97	78.51±2.31	84.58±0.02
	CACB	**76.95±0.43**	**70.94±0.05**	**80.09±0.27**	**61.61±0.03**	**84.98±1.09**	**93.83±0.01**
NMI	w/o L_n	55.34±0.03	42.42±0.11	70.00±0.16	54.38±0.02	72.90±1.30	68.88±0.03
	w/o L_s	53.79±0.22	31.21±2.04	59.04±2.34	33.65±1.10	73.50±1.45	68.89±0.22
	w/o Proj.	61.00±0.93	44.69±0.03	74.73±1.23	56.70±0.94	81.10±0.17	76.33±0.06
	w/o CACB	59.73±0.21	43.54±0.04	67.96±0.87	55.34±0.72	80.28±0.73	77.47±0.01
	CACB	**61.02±0.48**	**46.09±0.05**	**75.49±0.20**	**56.72±0.04**	**82.54±0.54**	**80.53±0.01**
ARI	w/o L_n	48.77±0.17	43.67±0.15	61.68±0.14	32.67±0.03	55.71±4.03	59.74±0.06
	w/o L_s	45.77±0.38	27.97±2.79	49.69±3.26	15.74±0.55	59.49±3.07	76.74±0.30
	w/o Proj.	58.34±0.42	46.69±0.04	65.80±2.42	45.05±1.39	76.58±2.35	85.81±0.03
	w/o CACB	56.15±0.24	45.01±0.04	58.38±1.15	44.08±0.76	74.53±0.95	79.50±0.04
	CACB	**58.51±0.67**	**47.74±0.08**	**66.82±0.28**	**45.15±0.05**	**82.61±1.84**	**88.16±0.01**
F1	w/o L_n	62.53±0.15	64.19±0.12	76.47±0.14	39.39±0.04	67.90±4.08	67.14±0.02
	w/o L_s	65.09±0.97	52.98±2.58	65.80±0.73	26.89±0.43	64.14±3.24	81.37±0.28
	w/o Proj.	74.57±0.03	65.02±0.05	76.78±2.84	47.75±3.38	74.88±0.18	86.62±0.08
	w/o CACB	73.24±0.11	64.41±0.03	71.72±1.93	48.66±0.91	71.20±4.48	79.75±0.00
	CACB	**74.83±0.41**	**65.70±0.08**	**77.71±0.32**	**50.72±0.04**	**81.10±2.83**	**91.44±0.02**

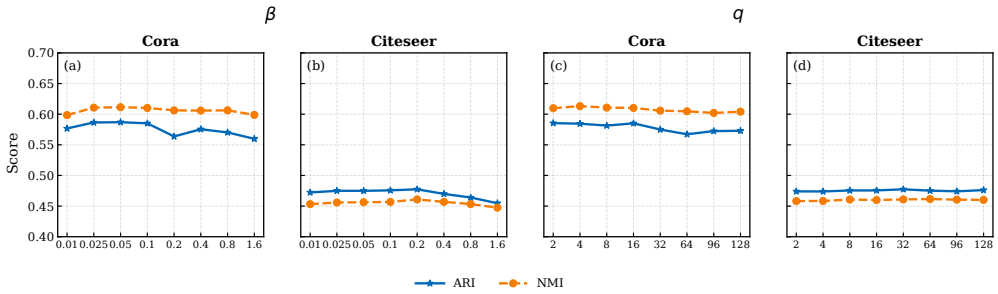


Figure 7: Sensitivity on Cora and Citeseer to β (a–b) and q (c–d).

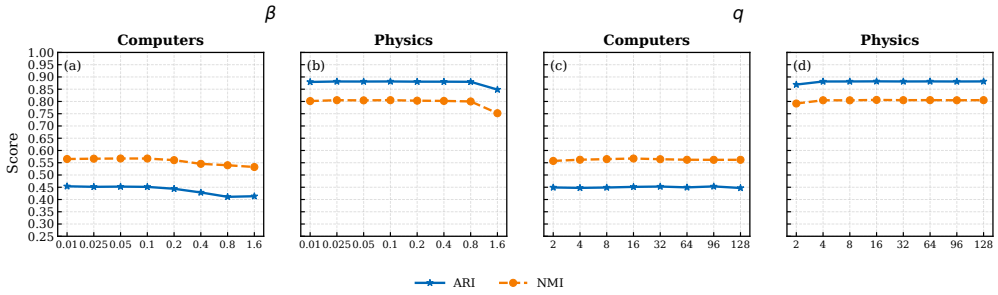


Figure 8: Sensitivity on Computers and Physics to β (a–b) and q (c–d).

D.4 Visualization of Learned Representations

Figure 9 provides a qualitative comparison of the learned representation spaces on Photo and CS. Compared with the heavily overlapping raw features, the learned representations exhibit clearer cluster structures. CACB produces more compact and better separated class regions while reducing the inter-class mixing observed in competing methods. These visualizations are consistent with the quantitative clustering improvements reported in Table 1.

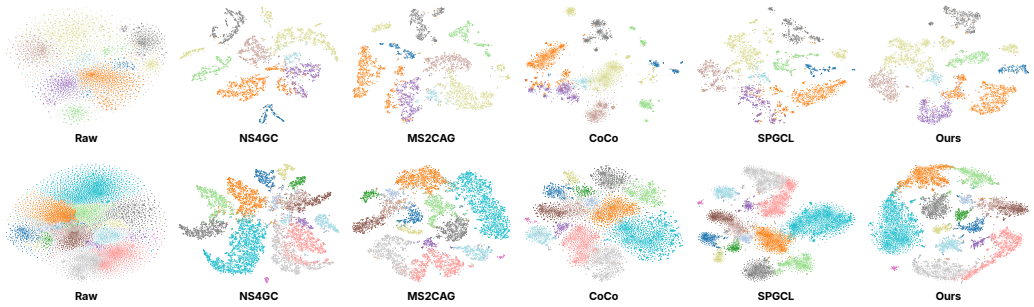


Figure 9: **Visualization of learned representations on Photo and CS.** The top and bottom rows correspond to Photo and CS, respectively. Colors indicate ground-truth classes and are used only for visualization.