



Breaking Structural Isolation: Scalable Graph Clustering via Community-Aware Sampling and Structural Entropy

Jingyun Zhang
Beihang University
Beijing, China
zhangjingyun@buaa.edu.cn

Hao Peng*
Beihang University
Beijing, China
Hangzhou Innovation Institute,
Beihang University
Hangzhou, China
penghao@buaa.edu.cn

Jianxin Li
Beihang University
Beijing, China
lijx@buaa.edu.cn

Angsheng Li
Beihang University
Beijing, China
angsheng@buaa.edu.cn

Philip S. Yu
University of Illinois at Chicago
Chicago, USA
psyu@uic.edu

ABSTRACT

Unsupervised graph clustering is a fundamental technique for uncovering underlying semantic patterns in large-scale networks. Although Graph Contrastive Learning has demonstrated promising performance, existing methods often suffer from the "structural isolation" issue during mini-batch training, making it challenging to capture cohesive community structures that characterize the global topological distribution. To address these challenges, we propose **SCISE**, a **S**calable unsupervised graph **C**lustering framework that preserves structural **I**ntegrity by synergizing community-aware sampling with constrained **S**tructural **E**ntropy. Specifically, we first introduce the Structural Entropy Community Constraint operator (SECC), which optimizes structural information within a constrained solution space to mitigate community fragmentation and enhance partition cohesion. Second, to prevent global information loss during batch training, we design a Community-Aware Sampling Expansion (CSampE) mechanism that incorporates the community context of target nodes into sampling batches, effectively breaking structural barriers and preserving topological integrity. Finally, we devise a Structural Contrastive Learning (StructCL) module that refines edge weights based on intra-batch structural similarity, guiding the encoder to learn representations in a higher-order structural space. Extensive experiments on six mainstream benchmark datasets demonstrate that SCISE significantly outperforms state-of-the-art algorithms, with ablation studies and robustness analyses further validating its effectiveness and reliability for real-world large-scale graphs.

PVLDB Reference Format:

Jingyun Zhang, Hao Peng, Jianxin Li, Angsheng Li, and Philip S. Yu.
Breaking Structural Isolation: Scalable Graph Clustering via Community-Aware Sampling and Structural Entropy. PVLDB, 19(11): 3497 - 3510, 2026.
doi:10.14778/3836663.3836704

*Corresponding author.

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit <https://creativecommons.org/licenses/by-nc-nd/4.0/> to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights

PVLDB Artifact Availability:

The source code, data, and/or other artifacts have been made available at <https://github.com/SELGroup/SCISE>.

1 INTRODUCTION

Graph-structured data has become pervasive in representing complex relational information across diverse domains, such as social media, citation networks, and biological systems [12, 16]. As a fundamental task in network analysis, unsupervised graph clustering aims to partition nodes with high topological and attribute affinities into communities without reliance on manual labeling, thereby revealing the underlying semantic patterns and structural motifs within complex networks [10, 43]. This capability is critical for numerous downstream applications, including community detection, targeted recommendation, and anomaly discovery [2, 48]. Given the exponential growth of graph data and the prohibitive cost of human annotation, developing effective and scalable unsupervised clustering frameworks is of paramount importance for extracting meaningful structural insights from unlabeled, large-scale relational data [36, 42].

The prevailing paradigm for unsupervised graph clustering has shifted towards integrating Graph Neural Networks (GNNs) with Graph Contrastive Learning (GCL) to learn discriminative semantic embeddings [3, 19]. Typically, these methods leverage message-passing mechanisms and consistency regularization to reinforce feature similarity among intra-cluster nodes [56]. However, as the scale of real-world networks grows exponentially, these frameworks encounter a formidable bottleneck: the trade-off between computational scalability and structural integrity [11, 12].

To handle massive graphs, existing methods predominantly adopt mini-batch training via various stochastic sampling strategies [50]. Although GCL has demonstrated remarkable performance in representation learning, its efficacy on large-scale data is severely hampered by the "structural isolation" issue induced by batch-wise operations [32]. Since traditional samplers primarily focus on local

licensed to the VLDB Endowment.

Proceedings of the VLDB Endowment, Vol. 19, No. 11 ISSN 2150-8097.
doi:10.14778/3836663.3836704

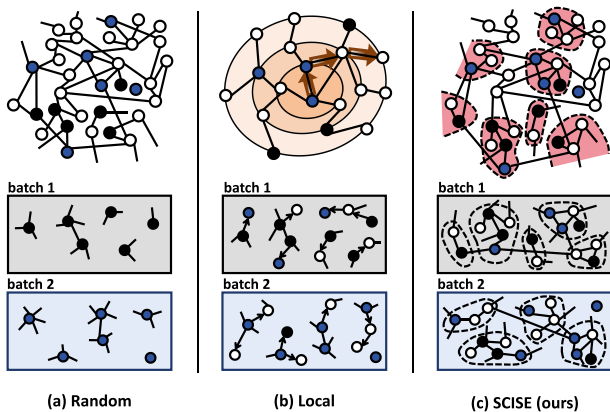


Figure 1: Comparison of sampling strategies. (a) Random: fragments communities into batches. (b) Local: limited to neighborhood context. (c) SCISE (Ours): incorporates global community context to preserve structural integrity within mini-batches.

neighborhood reachability, they inevitably sever the global community context, sequestering nodes of the same semantic cluster into disparate batches. Such limitations are visually depicted in Figure 1, where conventional random sampling (a) and local neighborhood sampling (b) fail to maintain community coherence within mini-batches. This structural isolation inherently erodes global topological awareness, a problem further exacerbated by the inherent limitations of current technological paths: contrastive-based frameworks [17, 25, 31, 46] typically lack mechanisms to compensate for such sampling-induced fragmentation, while structure-aware methods [22, 34, 35] often rely on noisy, unrefined graph statistics. Consequently, the absence of global topological awareness makes it challenging for models to capture cohesive community structures, leading to fragmented representations that fail to reflect the underlying semantic organization.

To address the aforementioned challenges, we propose **SCISE**, a Scalable unsupervised graph Clustering framework designed to preserve structural Integrity through the synergy of community-aware sampling and constrained Structural Entropy. SCISE reconciles the inherent conflict between computational scalability and community cohesion through three synergistic components. First, we introduce the Structural Entropy Community Constraint (SECC) operator to explicitly quantify and reinforce community compactness. Unlike traditional structural entropy metrics that often lead to excessive partitioning in unlabeled scenarios, SECC optimizes structural information within a bounded solution space constrained by the target community number. This allows the model to maintain minimal entropy while effectively mitigating community fragmentation. Second, to counteract the information loss inherent in mini-batch training, we design a Community-Aware Sampling Expansion (CSampE) mechanism. By identifying and incorporating the global community context of target nodes into each sampling batch, CSampE dismantles structural barriers between batches and restores the topological continuity required for stable optimization. Finally, we devise a Structural Contrastive Learning (StructCL)

method to guide the encoder toward a more discriminative representation space. Diverging from conventional paradigms that rely on noisy raw topology, StructCL refines higher-order edge weights based on latent structural similarities among intra-batch nodes. These refined affinities serve as structural guidance to maximize consistency among similar nodes, ultimately yielding high-quality embeddings that are partitioned into final clusters.

To evaluate the effectiveness of the proposed SCISE, we conduct extensive experiments on three medium-scale datasets (Photo, Computer, Pubmed) and three large-scale datasets (Ogbn-arxiv, Reddit, Ogbn-products). Quantitative results yield several critical insights: First, comprehensive performance analysis demonstrates that SCISE achieves overall superior performance compared to ten state-of-the-art baselines, confirming its consistent efficacy across both medium and large-scale scenarios. Second, rigorous ablation studies dissect the synergistic effects of our modules, validating that SECC, CSampE, and StructCL are each indispensable for preserving structural integrity. Finally, the stability analysis assesses the model’s invariance to hyper-parameter variations, underscoring its robustness and deployment in real-world, large-scale systems. The main contributions of this work are summarized as follows.

- **Scalable Framework (SCISE):** We propose SCISE, an unsupervised graph clustering framework that effectively harmonizes the conflict between computational scalability and structural integrity through the synergy of community-aware sampling and constrained structural entropy.
- **Community-Aware Sampling Expansion (CSampE):** We develop the CSampE mechanism to bridge "structural isolation" by adaptively incorporating global community contexts into mini-batches, thereby restoring topological continuity in mini-batch training.
- **Structural Entropy Community Constraint (SECC):** We introduce the SECC operator to explicitly quantify community compactness within a constrained space, significantly mitigating partition fragmentation and ensuring global community integrity.
- **Extensive Validation and Insights:** Through extensive evaluations on six benchmark datasets, we demonstrate the overall superior performance and robustness of SCISE, while identifying the synergistic and complementary effects of its core components.

2 RELATED WORK

2.1 Unsupervised Graph Clustering

Unsupervised graph clustering evolves from traditional statistical methods to sophisticated deep learning paradigms. Systematic surveys [24, 28] establish comprehensive taxonomies, categorizing the field into GNN-based and contrastive learning-based approaches while analyzing the trade-offs between generative and discriminative models. Within the GNN paradigm, DMoN [35] achieves end-to-end community detection via modularity maximization, while DeepCut [1] and RGC [21] extend these techniques to unsupervised image segmentation and reinforcement learning-based automated cluster number determination, respectively. Recently, contrastive learning has emerged as a dominant trend to enhance discriminative power; for instance, SCGC [17] simplifies contrastive architectures through neighbor-oriented loss, while CCGC [45] and

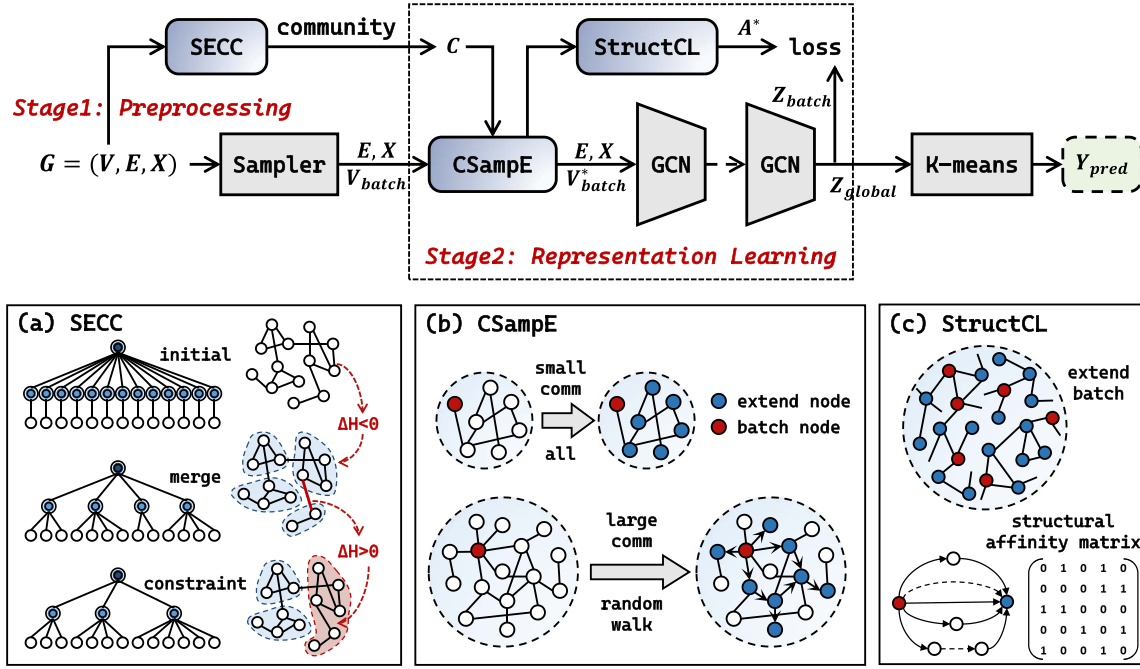


Figure 2: Overall framework of SCISE.

DCGL [7] leverage clustering-oriented guidance to refine representation learning. Parallel to these algorithmic advancements, structural entropy introduces a robust information-theoretic perspective for capturing graph hierarchies, with methods like USER [37] and the Lorentz-based LSEnet [34] pushing the boundaries of accuracy and parameter-free clustering. Furthermore, applying structural entropy to unsupervised graph clustering [54] opens new avenues for synergizing structural information theory with neural networks.

2.2 Structural Information Theory

Structural Information Theory has been validated in many applications. Introducing structural entropy in neural networks captures the underlying topological connectivity and reduces random interference [37, 54]. The hierarchical nature of the structure entropy encoding tree provides new methods for hierarchical structure pooling in graph neural network [40], unsupervised image segmentation [14, 44], dimension estimation [47], state abstraction [52, 53] in reinforcement learning, social bot detection [26, 51], unsupervised social event detection [5], and recommendation systems [55]. Additionally, reconstructing the graph structure on the hierarchical encoding tree suppresses edge noise and enhances the learning ability of the graph structure [57]. Furthermore, modifying the network structure based on minimizing structural entropy achieves maximum deception of community structure [23]. Similarly, SEGA [39] and SEPC [15] improve the performance of graph contrastive learning, guided by the principle of structural entropy minimization. Based on the homogeneous graph structure entropy, studys of multi-relational graph SE [4] and hypergraph further SE [49, 55] extend the structural information theory, making it suitable for more complex scenarios.

2.3 Large Graph Learning

Large-scale graph learning has evolved to address the scalability bottlenecks of traditional Graph Neural Networks (GNNs). As foundational works, GraphSAGE [12] introduces neighbor sampling for inductive representation learning, while VR-GCN [6] adopts variance reduction to stabilize stochastic training. To overcome the memory explosion in deep GNNs, Cluster-GCN [8] leverages graph clustering to restrict neighborhood search within dense subgraphs. Further refinements in sampling strategies, such as GraphSAINT [50], utilize subgraph sampling to decouple the number of layers from sampling complexity. Parallel to these developments, model simplification has become a key trend; SGC [38] removes non-linearities to fold GCN into a linear filter. Recently, the research frontier has shifted toward scaling Graph Transformers to large graphs. GraphGPS [27] decouples local and global mechanisms, while SGFormer [41] and Exphormer [33] employ simplified global attention or sparse frameworks to mitigate quadratic complexity.

3 PRELIMINARY

Definition 3.1 (Unsupervised Graph Clustering). The primary objective of unsupervised graph clustering is to partition nodes into cohesive groups by exclusively exploiting intrinsic topological structures and nodal attributes, in the absence of any ground-truth labels. Formally, let $G = (V, E, X)$ denote an undirected homogeneous graph, where $V = \{v_1, v_2, \dots, v_N\}$ is the set of N nodes and $E \subseteq V \times V$ represents the set of M edges. The nodal properties are encapsulated in a feature matrix $X \in \mathbb{R}^{N \times f}$, where f denotes the dimensionality of the attribute space. The connectivity patterns within G are characterized by a symmetric adjacency matrix $A \in \{0, 1\}^{N \times N}$, where $A_{ij} = 1$ indicates the existence of an edge

between nodes v_i and v_j . Given the graph G , the objective is to learn a mapping function $f : \{V, A, X\} \rightarrow Y$ that assigns each node to one of K disjoint clusters $Y \in \{1, \dots, K\}^N$, where K is a pre-defined or inferred number of clusters.

Definition 3.2 (Structural Entropy). Structural information theory [18] is originally proposed for measuring the structural information contained within a graph. Specifically, this theory aims to calculate the structural entropy of the homogeneous graph $G = (V, E)$, which reflects its uncertainty when undergoing hierarchical division. The structural information of the homogeneous graph G determined by the encoding tree $\mathcal{T}$ is defined as:

$$H^{\mathcal{T}}(G) = - \sum_{\alpha \in \mathcal{T}, \alpha \neq \lambda} \frac{g_{\alpha}}{\text{vol}(G)} \log \frac{\text{vol}(\alpha)}{\text{vol}(\alpha^{-})}, \quad (1)$$

where $\text{vol}(G)$ is the sum of the degrees of all nodes in the graph G . Each vertex in the encoding tree $\mathcal{T}$ corresponds to a node subset T_{α} in the graph G . $\text{vol}(\alpha)$ is the volume of T_{α} and is the sum of the degrees of all nodes in the subset T_{α} . α^{-} is the parent vertex of vertex α in the encoding tree. g_{α} is the sum of weights of all edges from node subset T_{α} to node subset V/T_{α} , which can be understood as the total weight of the edges from the nodes outside the node subset T_{α} to the nodes inside T_{α} , or the total weight of the cut edges. $\frac{g_{\alpha}}{\text{vol}(G)}$ represents the probability that the random walk enters T_{α} . The structural entropy $H(G)$ of graph G is the minimum $H^{\mathcal{T}}(G)$. Let $\mathcal{H}_h$ be encoding trees whose height is not greater than h , then the h -dimensional structural entropy of G is defined as $H_h(G) = \min\{H^{\mathcal{T}_h}(G)\}$.

4 METHODOLOGY

This section presents a detailed formalization of the proposed scalable unsupervised graph clustering framework, **SCISE**. As illustrated in Figure 2, SCISE comprises three strategic components: Structural Entropy Community Constraint (SECC), which optimizes structural entropy minimization under community-cardinality constraints to generate cohesive partitions; Community-aware Sampling Extension (CSampE), which mitigates the "structural isolation" of mini-batches by performing community-aware context expansion; and Structural Contrastive Learning (StructCL), which executes contrastive alignment on a higher-order topology reconstructed via structural similarity. Specifically, the pipeline begins with SECC (Section 4.1) as a preprocessing stage, leveraging a constrained structural entropy operator to partition the raw graph into dense community anchors. During the representation learning stage, CSampE (Section 4.2) augments sampled nodes with their corresponding intra-community neighbors to restore topological integrity. Subsequently, StructCL (Section 4.3) refines the batch-wise subgraph by re-weighting edges based on structural similarities approximated through random walks. Finally, the Overall Optimization (Section 4.4) integrates these modules into a unified objective to refine latent embeddings, followed by K-means clustering to produce the final partitions. A detailed complexity analysis of SCISE is provided in Section 4.5.

4.1 Structural Entropy Community Constraint

Applying structural information theory to graph structure learning has demonstrated substantial advantages across various downstream tasks, primarily due to its capacity to reveal the hierarchical topological semantics of nodes. However, current optimization paradigms for structural entropy predominantly rely on a greedy merge operator—iteratively merging node pairs until no further reduction in structural entropy is possible. This unconstrained, locally greedy approach is highly susceptible to over-segmentation and frequently traps the model in local optima. As illustrated in the case study in Section 5.7, conventional methods often yield a plethora of fragmented, micro-communities (e.g., sizes of 2, 3, or 4). In ideal real-world scenarios, however, community partitions are expected to be cohesive, characterized by an abundance of medium-sized communities and a minimal presence of extreme-sized (excessively large or small) communities.

To address this bottleneck, we impose a community cardinality constraint, denoted as N_{comm} , on the greedy structural entropy minimization process. Consequently, the termination criterion for the merge operator shifts from "the absence of entropy-decreasing node pairs" to "convergence to the target community count N_{comm} ". Specifically, the process initializes with all N nodes forming individual singleton communities, denoted as $C = \{c_1, c_2, \dots, c_N\}$, where inter-community connections correspond to the original edge set E . For each edge $e_{i,j} \in E$, we calculate the change in structure entropy, $\Delta H(i, j)$, resulting from the merger of c_i and c_j :

$$\begin{aligned} \Delta H(c_i, c_j) = & - \frac{g_{c_i \cup c_j}}{\text{vol}(G)} \log \frac{\text{vol}(c_i \cup c_j)}{\text{vol}(G)} + \sum_{v_t \in c_i \cup c_j} \left(- \frac{d_{v_t}}{\text{vol}(G)} \log \frac{d_{v_t}}{\text{vol}(c_i \cup c_j)} \right) \\ & - \left(- \frac{g_{c_i}}{\text{vol}(G)} \log \frac{\text{vol}(c_i)}{\text{vol}(G)} + \sum_{v_t \in c_i} \left(- \frac{d_{v_t}}{\text{vol}(G)} \log \frac{d_{v_t}}{\text{vol}(c_i)} \right) \right) \\ & - \left(- \frac{g_{c_j}}{\text{vol}(G)} \log \frac{\text{vol}(c_j)}{\text{vol}(G)} + \sum_{v_t \in c_j} \left(- \frac{d_{v_t}}{\text{vol}(G)} \log \frac{d_{v_t}}{\text{vol}(c_j)} \right) \right), \end{aligned} \quad (2)$$

where $\text{vol}(\cdot)$ denotes the volume of the community, $g(\cdot)$ represents the cut edges, and d_{v_t} signifies the node degree. Subsequently, we sort the entropy variations of all candidate edges in increasing order, retaining OP pairs where $\Delta H(c_i, c_j) < 0$. To control the pace and efficiency of the agglomeration, we introduce a hyperparameter p ($0 < p \leq 1$), executing the merge operation only on the top p proportion of these candidate pairs. Following one iteration, a new community set $C' = \{c_1, c_2, \dots, c_{N'}\}$ is formed, where $N' = N - p \times OP$. The core innovation of the constraint lies in its ability to circumvent premature convergence. So, before termination, if no node pairs satisfy $\Delta H(c_i, c_j) < 0$, yet the current number of communities has not reached N_{comm} , the algorithm relaxes the strict entropy-descent condition. It forces further merges—regardless of temporary structural entropy increases—until the total number of communities is strictly limited to N_{comm} (detailed in Algorithm 1). This mechanism effectively breaks out of local optima and yields highly compact partitions by compelling nodes into exactly N_{comm} cohesive groups, while ensuring global structural integrity. The unconstrained minimization algorithm can be used to estimate a feasible upper bound for the community range. As demonstrated in Section 5.4, SCISE is highly insensitive to the precise value of N_{comm} , and therefore no

exhaustive search for an optimal community number is required in practice.

Algorithm 1: Structural Entropy Community Constraint

Input: Graph $G = (V, E)$, target community count N_{comm} , merge speed $p \in (0, 1]$.
Output: Community partition $C = \{c_1, c_2, \dots, c_{N_{\text{comm}}}\}$.

```

1 Initialize  $C = \{\{v_1\}, \{v_2\}, \dots, \{v_N\}\}$ ;
2 Calculate  $\text{vol}(G) = \sum_{v \in V} d_v$ ;
3 while  $|C| > N_{\text{comm}}$  do
4    $\mathcal{E}_{\text{cand}} = \{(c_i, c_j) \mid c_i, c_j \in C, i < j, \exists e_{u,w} \in E \text{ s.t. } v_u \in c_i, v_w \in c_j\}$ ;
5   foreach  $(c_i, c_j) \in \mathcal{E}_{\text{cand}}$  do
6     Calculate  $\Delta H(c_i, c_j)$  via Eq. (2);
7   end
8   Sort  $\mathcal{E}_{\text{cand}}$  in descending order of  $\Delta H$ ;
9    $P_{\text{pos}} = \{(c_i, c_j) \in \mathcal{E}_{\text{cand}} \mid \Delta H(c_i, c_j) < 0\}$ ;
  // Forced Convergence
10  if  $P_{\text{pos}}$  is empty then
11     $P_{\text{pos}} = \mathcal{E}_{\text{cand}}$ ;
12  end
13   $OP = |P_{\text{pos}}|$ ;
14   $k = \min(p \cdot OP, |C| - N_{\text{comm}})$ ;
15  Select top  $k$  pairs from  $P_{\text{pos}}$  and merge them;
16  Update  $C$  and  $|C| = |C| - k$ ;
17 end
18 return  $C$ 

```

4.2 Community-aware Sampling Extension

To maintain computational efficiency on large-scale graphs, models typically rely on stochastic node sampling to construct mini-batches. However, this inherent randomness frequently severs deep-seated community ties. Nodes belonging to the same cohesive cluster may be partitioned into disparate batches, preventing the model from capturing the full topological evolution of a community within a single gradient update—a phenomenon we define as Structural Isolation. Conventional node-centric paradigms (e.g., neighbor sampling) often neglect long-range community constraints, leading to fragmented subgraphs where intra-batch nodes are topologically isolated. This fragmentation hinders the model from learning the cohesive hierarchical semantics intrinsic to the data.

CSampE bridges this gap by adaptively expanding the initial batch $\mathcal{V}_{\text{batch}}$ into structurally dense collectives based on underlying community memberships. By shifting the sampling paradigm from "node expansion" to "community alignment," CSampE identifies the corresponding community $c(v) \in C$ for each seed node $v \in \mathcal{V}_{\text{batch}}$ via the partitions generated by SECC. This mechanism pulls nodes from the same community and their critical topological structures into a unified sampling space, forcing the model to observe complete community semantics in each training step. Specifically, for each node $v \in \mathcal{V}_{\text{batch}}$ associated with community $c(v)$, we introduce a cardinality threshold θ to govern the expansion strategy:

Case I: Structural Restoration for Small Communities ($|c(v)| \leq \theta$). For small-scale communities, we execute a full-community expansion to ensure the local topology is entirely restored within a single training step. The expanded set $\mathcal{V}_{\text{ext}}$ is defined as:

$$\mathcal{V}_{\text{ext}} = \{u \mid \exists v \in \mathcal{V}_{\text{batch}}, u \in c(v)\}, \quad (3)$$

Case II: Backbone Extraction for Large Communities ($|c(v)| > \theta$). In expansive communities, full inclusion would trigger "batch explosion" and excessive memory overhead. To distill the most representative features, we execute w_l random walks of length w_l starting from node v within $c(v)$. Let $f(u; v)$ denote the frequency of node u being visited by node v within $c(v)$. We identify the structural backbone by filtering nodes that exceed the mean visitation frequency:

$$\mathcal{V}_{\text{ext}} = \{u \mid \exists v \in \mathcal{V}_{\text{batch}} \text{ s.t. } u \in c(v) \text{ and } f(u; v) > \bar{f}_v\}, \quad (4)$$

where $\bar{f}_v = \frac{1}{|\mathcal{R}_v|} \sum_{w \in \mathcal{R}_v} f(w; v)$ denotes the mean visitation frequency associated with seed v , and $\mathcal{R}_v$ represents the set of all nodes reachable from seed v within the community via random walks. The final expanded batch, $\mathcal{V}_{\text{batch}}^* = \mathcal{V}_{\text{batch}} \cup \mathcal{V}_{\text{ext}}$, transforms a collection of isolated points into a coherent structural field. This dual-track strategy enables the model to leverage dense community-aware signals while maintaining an effectively managed batch size, thereby enhancing the stability of gradient updates and the robustness of learned hierarchical representations.

4.3 Structural Contrastive Learning

While the CSampE module successfully assembles structurally dense mini-batches, efficiently translating these multi-hop topological patterns into the representation space remains a significant challenge. A naive approach would be to treat all node pairs within the same community as positive anchors. However, for large-scale graphs, even with the reduced scale of a mini-batch, this would lead to a quadratic explosion ($O(|\mathcal{V}_{\text{batch}}^*|^2)$) in the number of positive pairs, incurring prohibitive computational overhead. To quantify community-level structural relations without inflating complexity, we propose Structural Contrastive Learning (StructCL).

To efficiently capture community-level structural relations, we construct a structural affinity matrix A^* based on the visitation counts of localized random walks. For each node $v_i \in \mathcal{V}_{\text{batch}}^*$, we execute w_l random walks of length w_l within the graph. Let $\text{Count}(v_j; v_i)$ denote the total frequency of node v_j being visited in walks originating from v_i . The structural affinity matrix $A^* \in \mathbb{R}^{B \times B}$ (where $B = |\mathcal{V}_{\text{batch}}^*|$) is defined as:

$$A_{ij}^* = \begin{cases} \text{Count}(v_j; v_i), & \text{if } i \neq j \text{ and } v_j \in \text{RW}(v_i) \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

This sparse matrix effectively transforms the batch into a weighted structural subgraph. Within this subgraph, the weights A_{ij}^* serve as confidence scores for topological co-occurrence: a higher frequency indicates a more stable structural bond. By selectively focusing on these frequently visited "structural anchors," StructCL effectively captures the core community semantics while avoiding the computational redundancy of all-to-all community matching.

To align the neural similarity of embeddings with the distilled structural affinities, for any node pair (v_i, v_j) with $A_{ij}^* > 0$, we define the consistency loss as a multi-positive InfoNCE objective:

$$\mathcal{L}_{\text{struct}} = -\frac{1}{B} \sum_{v_i \in \mathcal{V}_{\text{batch}}^*} \frac{1}{|\mathcal{P}_i|} \sum_{v_j \in \mathcal{P}_i} \log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{v_k \neq v_i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}, \quad (6)$$

where $\mathcal{P}_i = \{v_j \mid A_{ij}^* > 0\}$ is the sparse set of structural neighbors for node v_i . By minimizing $\mathcal{L}_{struct}$, the model adaptively encodes dense intra-community signals into the representation space. This self-supervised mechanism ensures that the learned embeddings are inherently endowed with community-topology awareness, effectively bridging the "structural isolation" gap while maintaining a bounded computational complexity suitable for large-scale graph analysis.

Algorithm 2: Overall execution procedure of SCISE.

Input: Graph with feature $G = (V, E)$, features $\mathcal{X}$, adjacency matrix A , target community count N_{comm} , merge speed $p \in (0, 1]$, expansion threshold θ , random walk params (w_l, w_t) , batch size B , target clusters K .

Output: Cluster assignments $\mathcal{Y}$.

```

// Stage 1: Preprocessing community division
1  $C \leftarrow \text{SECC}(G, N_{comm}, p)$  via Algorithm 1;
2 Construct community lookup mapping  $c(v)$  for each  $v \in V$ 
  from  $C$ ;
// Stage 2: Structure-Aware Representation Learning
3 for  $epoch = 1, 2, 3, \dots$  do
4   foreach  $\text{mini-batch } \mathcal{V}_{batch} \subset V$  do
5     Initialize  $\mathcal{V}_{ext} \leftarrow \emptyset$ ;
6     foreach  $v \in \mathcal{V}_{batch}$  do
7       if  $|c(v)| \leq \theta$  then
8         // Case I: Full Expansion
9          $\mathcal{V}_{ext} \leftarrow \mathcal{V}_{ext} \cup \{u \mid u \in c(v)\}$  via Eq. 3;
10      end
11     else
12       // Case II: Backbone Extraction
13       Perform  $w_t$  random walks of length  $w_l$ 
14       within  $c(v)$ ;
15       Update  $\mathcal{V}_{ext}$  via Eq. 4;
16     end
17    $\mathcal{V}_{batch}^* = \mathcal{V}_{batch} \cup \mathcal{V}_{ext}$ ;
18   // StructCL Optimization
19   Compute structural affinity  $A^*$  via Eq. 5;
20    $Z_{batch} = \text{GCN}(\mathcal{X}[\mathcal{V}_{batch}^*], A)$ ;
21   calculate  $\mathcal{L}_{struct}$  via Eq. 6;
22    $\mathcal{L}_{struct}$  backward;
23 end
24  $Z_{global} = \emptyset$ ;
25 foreach  $\text{mini-batch } \mathcal{V}_{batch} \subset V$  do
26    $Z_{batch} = \text{GCN}(\mathcal{X}[\mathcal{V}_{batch}], A)$ ;
27    $Z_{global} = Z_{global} \cup Z_{batch}$ ;
28 end
29  $\mathcal{Y} = \text{K-means}(Z_{global}, K)$ ;
30 return  $\mathcal{Y}$ 

```

4.4 Overall Optimization

The overall execution procedure of SCISE is summarized in Algorithm 2. SCISE adopts a two-stage paradigm consisting of community division preprocessing and structure-aware representation learning. Specifically, the SECC is first invoked to obtain a stable community structure mapping C , which circumvents the high computational overhead of recalculating communities at each training step. During the training phase, for each mini-batch $\mathcal{V}_{batch}$, the CSampE adaptively expands the batch with community nodes $\mathcal{V}_{extend}$ based on the pre-acquired community scales. This enables

Table 1: Statistics of datasets.

Dataset	#Node	#Edge	#Feature	#Cluster	Degree
Photo	7,650	119,081	745	8	31.13
Computers	13,752	245,861	767	10	35.76
Pubmed	19,717	44,324	500	3	4.50
Ogbn-arxiv	169,343	1,166,243	128	40	13.77
Reddit	232,965	114,615,892	602	41	491.98
Ogbn-products	2,449,029	61,859,140	100	47	50.52

the subsequent StructCL objective to distill community-level semantic signals into the GCN encoder without incurring the "neighbor explosion" problem. Subsequently, the node features are processed through a GCN backbone to learn the node embeddings Z_{batch} . Finally, a single-pass K-means clustering is performed on the consolidated global embeddings Z_{global} to obtain the final partitions $\mathcal{Y}_{pred}$.

4.5 Complexity Analysis

To demonstrate the scalability of SCISE, we analyze its computational complexity across two primary stages: offline structural prior acquisition and structure-aware representation learning. First, the complexity of the SECC module is governed by the construction of the structural entropy encoding tree. Given a graph G with N nodes and M edges, when the parallel merge rate is p , the maximum number of rounds required for hierarchical merging is $\log_{1-p} \frac{1-p}{N-1}$. Consequently, the time complexity for constructing the structural prior is $O(M \cdot \log_{1-p} \frac{1-p}{N-1})$, which can be simplified to $O(M \log N)$. Since SECC is performed as a one-time preprocessing step, it does not add to the iterative overhead of the training phase.

In the online training phase, the complexity for each mini-batch $\mathcal{V}_{batch}$ consists of three components: CSampE expansion, neural encoding, and StructCL optimization. For each batch, the adaptive expansion via either full community retrieval or localized random walks incurs a complexity of $O(B \cdot w_l \cdot w_t)$, where B is the initial batch size, and (w_l, w_t) are the random walk parameters. The forward pass through the GCN has a complexity of $O(|\mathcal{V}_{batch}^*| \cdot f \cdot d)$, where f and d denote the input and embedding dimensions, respectively. Calculating the structural contrastive loss involves pairwise similarity computations within the expanded batch, resulting in $O(|\mathcal{V}_{batch}^*| \cdot w_l \cdot w_t)$. Given that $|\mathcal{V}_{batch}^*|$ is bounded within a small constant multiple of B , the total time complexity per iteration during the representation learning stage is $O(B \cdot (w_l w_t + f d))$. Extending this to a full training epoch, the overall complexity remains $O(N \cdot (w_l w_t + f d))$.

5 EXPERIMENTS

In this section, we conduct empirical experiments to demonstrate the effectiveness of the proposed framework SCISE. We aim to answer five research questions as follows: **Q1: Effectiveness.** How does SCISE perform in unsupervised graph clustering tasks compared with state-of-the-art baselines (Section 5.2)? **Q2: Ablation Study.** How do the SECC, CSampE, and StructCL modules influence the performance of SCISE, and how do they interact with each other (Section 5.3)? **Q3: Stability Analysis.** How do key hyperparameters impact the performance of SCISE, and what is the degree of model sensitivity to these parameters (Section 5.4)? **Q4:**

Table 2: Comparison of the NMI, ARI, ACC, and F1 across different methods on six datasets. The best results are bolded, and the second-best results are underlined.

Method	Photo				Computers				Pubmed			
	NMI	ARI	ACC	F1	NMI	ARI	ACC	F1	NMI	ARI	ACC	F1
DGI	33.70 $\pm$ 0.85	22.10 $\pm$ 0.55	43.00 $\pm$ 1.10	35.20 $\pm$ 0.90	42.00 $\pm$ 1.05	30.60 $\pm$ 0.85	47.90 $\pm$ 1.20	39.00 $\pm$ 1.00	32.20 $\pm$ 0.80	29.20 $\pm$ 0.75	65.70 $\pm$ 1.65	65.40 $\pm$ 1.60
S3GC	59.80 $\pm$ 0.47	56.10 $\pm$ 0.73	75.20 $\pm$ 0.58	72.90 $\pm$ 0.69	56.00 $\pm$ 0.62	43.80 $\pm$ 0.41	58.80 $\pm$ 0.76	47.50 $\pm$ 0.55	33.30 $\pm$ 0.79	34.50 $\pm$ 0.64	71.30 $\pm$ 0.48	70.30 $\pm$ 0.71
SUBLIME	56.91 $\pm$ 0.43	45.11 $\pm$ 0.98	64.95 $\pm$ 0.61	61.24 $\pm$ 0.25	45.55 $\pm$ 0.73	24.89 $\pm$ 0.15	40.94 $\pm$ 0.88	34.64 $\pm$ 0.49	18.68 $\pm$ 0.32	16.16 $\pm$ 0.67	58.33 $\pm$ 0.59	58.69 $\pm$ 0.91
CONVERT	63.04 $\pm$ 0.34	55.20 $\pm$ 1.18	74.24 $\pm$ 0.67	69.13 $\pm$ 1.05	48.44 $\pm$ 1.42	34.25 $\pm$ 0.53	51.30 $\pm$ 0.91	42.17 $\pm$ 0.28	28.19 $\pm$ 0.79	26.59 $\pm$ 1.36	63.29 $\pm$ 0.45	62.29 $\pm$ 1.12
SCGC	54.96 $\pm$ 3.49	45.66 $\pm$ 3.58	64.90 $\pm$ 3.33	52.02 $\pm$ 5.26	47.41 $\pm$ 1.18	46.06 $\pm$ 2.91	61.16 $\pm$ 2.22	38.07 $\pm$ 3.24	26.05 $\pm$ 2.21	25.27 $\pm$ 3.51	65.12 $\pm$ 2.21	61.49 $\pm$ 3.40
DMoN	63.38 $\pm$ 0.46	52.41 $\pm$ 0.73	73.83 $\pm$ 0.58	71.05 $\pm$ 0.79	49.30 $\pm$ 0.64	31.82 $\pm$ 0.41	49.13 $\pm$ 0.77	39.29 $\pm$ 0.52	29.80 $\pm$ 0.71	24.52 $\pm$ 0.55	61.99 $\pm$ 0.69	62.17 $\pm$ 0.44
Dink-Net	74.36 $\pm$ 0.35	68.40 $\pm$ 0.98	81.71 $\pm$ 0.61	73.92 $\pm$ 1.14	42.10 $\pm$ 0.78	24.85 $\pm$ 1.64	45.17 $\pm$ 1.12	40.38 $\pm$ 1.93	29.93 $\pm$ 1.47	29.29 $\pm$ 0.56	67.29 $\pm$ 1.89	66.96 $\pm$ 1.03
LSEnet	58.94 $\pm$ 0.63	47.80 $\pm$ 0.88	65.46 $\pm$ 0.54	45.80 $\pm$ 0.91	55.03 $\pm$ 0.79	42.15 $\pm$ 1.02	53.27 $\pm$ 0.47	22.41 $\pm$ 0.68	29.67 $\pm$ 0.42	27.90 $\pm$ 0.36	58.98 $\pm$ 0.58	59.00 $\pm$ 0.61
MAGI	71.60 $\pm$ 0.76	61.50 $\pm$ 0.33	79.00 $\pm$ 1.07	72.90 $\pm$ 0.88	59.20 $\pm$ 1.19	46.20 $\pm$ 0.54	62.00 $\pm$ 0.90	57.40 $\pm$ 0.37	29.37 $\pm$ 0.25	29.25 $\pm$ 0.40	66.23 $\pm$ 0.23	65.11 $\pm$ 0.20
DeSE	68.94 $\pm$ 0.28	62.49 $\pm$ 0.12	<u>80.54</u> $\pm$ 0.12	73.67 $\pm$ 0.03	52.10 $\pm$ 0.15	45.64 $\pm$ 0.16	58.78 $\pm$ 0.31	43.17 $\pm$ 0.23	30.82 $\pm$ 0.17	29.47 $\pm$ 0.15	67.47 $\pm$ 0.21	67.36 $\pm$ 0.19
SCISE	74.53 $\pm$ 0.14	<u>64.51</u> $\pm$ 0.29	79.71 $\pm$ 0.15	76.22 $\pm$ 0.12	<u>58.46</u> $\pm$ 0.24	51.78 $\pm$ 0.80	67.32 $\pm$ 0.36	59.48 $\pm$ 0.84	35.77 $\pm$ 0.02	36.33 $\pm$ 0.02	72.38 $\pm$ 0.01	71.83 $\pm$ 0.01
Improv.(%)	$\uparrow$ 0.23%	-	-	$\uparrow$ 3.11%	-	$\uparrow$ 12.07%	$\uparrow$ 8.58%	$\uparrow$ 3.63%	$\uparrow$ 7.42%	$\uparrow$ 5.31%	$\uparrow$ 1.52%	$\uparrow$ 2.18%
Method	Ogbn-arxiv				Reddit				Ogbn-products			
	NMI	ARI	ACC	F1	NMI	ARI	ACC	F1	NMI	ARI	ACC	F1
DGI	41.20 $\pm$ 1.20	22.30 $\pm$ 0.80	31.40 $\pm$ 0.95	23.00 $\pm$ 0.70	30.60 $\pm$ 0.90	17.00 $\pm$ 0.60	22.40 $\pm$ 0.75	18.30 $\pm$ 0.55	46.70 $\pm$ 1.40	17.40 $\pm$ 0.65	32.00 $\pm$ 1.00	19.20 $\pm$ 0.60
S3GC	46.30 $\pm$ 0.58	27.00 $\pm$ 1.42	35.00 $\pm$ 0.91	23.00 $\pm$ 1.16	80.70 $\pm$ 1.27	74.50 $\pm$ 0.63	73.60 $\pm$ 1.08	56.00 $\pm$ 0.75	53.60 $\pm$ 0.84	<u>23.00</u> $\pm$ 0.35	40.20 $\pm$ 0.56	25.00 $\pm$ 0.92
SUBLIME			OOM				OOM				OOM	
CONVERT			OOM				OOM				OOM	
SCGC			OOM				OOM				OOM	
DMoN	35.60 $\pm$ 0.26	12.70 $\pm$ 0.97	25.00 $\pm$ 0.58	19.00 $\pm$ 1.14	62.80 $\pm$ 1.08	50.20 $\pm$ 0.34	52.90 $\pm$ 0.86	26.00 $\pm$ 0.41	42.80 $\pm$ 0.72	13.90 $\pm$ 1.19	30.40 $\pm$ 0.29	21.00 $\pm$ 0.95
Dink-Net	43.73 $\pm$ 0.23	<u>35.22</u> $\pm$ 0.41	<u>43.68</u> $\pm$ 0.35	<u>26.92</u> $\pm$ 0.48	78.91 $\pm$ 0.39	71.34 $\pm$ 0.26	76.03 $\pm$ 0.44	67.95 $\pm$ 0.31	50.78 $\pm$ 0.50	21.08 $\pm$ 0.34	41.09 $\pm$ 0.28	25.15 $\pm$ 0.42
LSEnet			OOM				OOM				OOM	
MAGI	<u>46.90</u> $\pm$ 0.29	31.00 $\pm$ 1.12	38.80 $\pm$ 0.68	26.60 $\pm$ 1.34	<u>87.50</u> $\pm$ 1.46	<u>90.70</u> $\pm$ 0.57	<u>91.10</u> $\pm$ 0.91	<u>85.30</u> $\pm$ 0.24	<u>55.10</u> $\pm$ 0.83	21.50 $\pm$ 1.27	<u>42.50</u> $\pm$ 0.42	<u>27.60</u> $\pm$ 1.05
DeSE			OOM				OOM				OOM	
SCISE	48.42 $\pm$ 0.08	37.66 $\pm$ 0.47	44.13 $\pm$ 0.59	31.52 $\pm$ 0.41	87.92 $\pm$ 0.12	90.97 $\pm$ 0.08	91.75 $\pm$ 0.24	86.48 $\pm$ 0.41	55.61 $\pm$ 0.02	25.65 $\pm$ 0.67	43.88 $\pm$ 0.39	27.82 $\pm$ 0.10
Improv.(%)	$\uparrow$ 3.25%	$\uparrow$ 6.92%	$\uparrow$ 1.04%	$\uparrow$ 17.08%	$\uparrow$ 0.48%	$\uparrow$ 0.30%	$\uparrow$ 0.71%	$\uparrow$ 1.39%	$\uparrow$ 0.93%	$\uparrow$ 11.50%	$\uparrow$ 3.24%	$\uparrow$ 0.78%

Robustness Against Noise. How robust is SCISE to noisy community priors (Section 5.5)? **Q5: Robustness Against Sparsity.** Can SCISE effectively alleviate structural isolation and maintain clustering performance when graph connectivity is severely degraded (Section 5.6)? **Q6: Case Study.** How do the community number (N_{comm}) constraints and merge speed (p) balance structural entropy and community cohesion (Section 5.7)?

5.1 Experiment Setup

Datasets. We conduct experiments on six benchmark datasets: Photo [30], Computers [30], Pubmed [29], Ogbn-arxiv [13], Reddit [12], and Ogbn-products [13]. Details of datasets are summarized in Table 1. **Baselines.** For graph clustering, we mainly compare SCISE with ten state-of-the-art baselines, including DGI [36], S3GC [9], SUBLIME [25], CONVERT [46], DMoN [35], Dink-Net [22], LSEnet [34], MAGI [20], DeSE [54], and SCGC [17]. **Evaluation Metrics.** We evaluate the accuracy and consistency of graph clustering with four metrics. NMI (Normalized Mutual Information) evaluates how well the predicted clusters match the true clusters in terms of information shared. ARI (Adjusted Rand Index) assesses the similarity between the predicted and true cluster assignments, adjusting for random chance. ACC (Accuracy) measures the proportion of nodes correctly assigned to their true clusters. F1 Score evaluates the balance between precision and recall in cluster assignments.

5.2 Overall Performance

Table 2 presents the comprehensive performance evaluation, where SCISE dominates across all benchmarks, achieving state-of-the-art (SOTA) results in 21 out of 24 metrics. On the Ogbn-arxiv dataset, our framework significantly outperforms the strongest baselines by 3.25% in NMI and a remarkable 17.08% in F1-score. We attribute this localized and global success to three synergistic strengths: **First, Countering Structural Drifting and Fragmentation:** Unlike existing structural methods (e.g., DMoN, DeSE) that often suffer from uncontrolled cluster merging or excessive fragmentation, SCISE employs the SECC operator to enforce explicit community-cardinality constraints. This yields partitions that are not only cohesive but also strictly aligned with the network’s inherent semantic distribution. **Second, Bridging Structural Isolation:** While conventional self-supervised methods (e.g., DGI, SUBLIME) rely on node-local consistency, they frequently lose global community context during batching. SCISE mitigates this via CSampE, which proactively restores community-level connectivity, and StructCL, which recalibrates higher-order dependencies. This combination ensures that the encoder captures holistic topological patterns rather than isolated neighborhood signals. **Third, Robust Scalability on Massive Graphs:** On large-scale benchmarks like Reddit and Ogbn-products, many high-performance competitors encounter Out-of-Memory (OOM) errors due to dense graph operations. In

Table 3: Ablation study on six datasets. Red cells indicate a significant performance degradation compared to the full model.

Variation	Photo				Computers				Pubmed			
	NMI	ARI	ACC	F1	NMI	ARI	ACC	F1	NMI	ARI	ACC	F1
SCISE	74.53 $\pm$ 0.14	64.51 $\pm$ 0.29	79.71 $\pm$ 0.15	76.22 $\pm$ 0.12	58.46 $\pm$ 0.24	51.78 $\pm$ 0.80	67.32 $\pm$ 0.36	59.48 $\pm$ 0.84	35.77 $\pm$ 0.02	36.33 $\pm$ 0.02	72.38 $\pm$ 0.01	71.83 $\pm$ 0.01
w/o SECC	74.41 $\pm$ 0.14	64.51 $\pm$ 0.10	79.74 $\pm$ 0.07	76.16 $\pm$ 0.29	58.11 $\pm$ 0.23	51.07 $\pm$ 0.38	67.05 $\pm$ 0.26	58.97 $\pm$ 0.13	35.59 $\pm$ 0.01	36.20 $\pm$ 0.02	72.31 $\pm$ 0.01	71.72 $\pm$ 0.01
w/o CSampE	73.62 $\pm$ 0.71	63.79 $\pm$ 0.43	79.48 $\pm$ 0.14	75.44 $\pm$ 0.93	49.00 $\pm$ 0.04	45.12 $\pm$ 0.08	64.67 $\pm$ 0.16	58.41 $\pm$ 0.06	35.87 $\pm$ 0.02	36.12 $\pm$ 0.01	72.19 $\pm$ 0.01	71.68 $\pm$ 0.02
w/o StructCL	74.79 $\pm$ 0.13	64.72 $\pm$ 0.09	79.82 $\pm$ 0.05	76.31 $\pm$ 0.06	55.69 $\pm$ 0.06	39.12 $\pm$ 0.10	55.72 $\pm$ 0.06	52.03 $\pm$ 0.02	28.59 $\pm$ 0.02	30.32 $\pm$ 0.03	68.73 $\pm$ 0.02	67.48 $\pm$ 0.04
w/o SECC & StructCL	74.77 $\pm$ 0.13	64.79 $\pm$ 0.14	79.88 $\pm$ 0.09	76.37 $\pm$ 0.06	56.45 $\pm$ 0.01	40.61 $\pm$ 0.02	57.14 $\pm$ 0.01	49.75 $\pm$ 0.01	25.62 $\pm$ 0.01	24.89 $\pm$ 0.01	65.90 $\pm$ 0.02	66.96 $\pm$ 0.02
w/o CSampE & StructCL	70.39 $\pm$ 0.60	62.66 $\pm$ 0.60	79.45 $\pm$ 0.37	77.10 $\pm$ 0.42	50.24 $\pm$ 0.59	35.18 $\pm$ 0.76	54.57 $\pm$ 0.22	55.82 $\pm$ 0.82	31.07 $\pm$ 0.02	36.76 $\pm$ 0.01	72.34 $\pm$ 0.01	70.57 $\pm$ 0.01
Variation	Ogbn-arxiv				Reddit				Ogbn-products			
	NMI	ARI	ACC	F1	NMI	ARI	ACC	F1	NMI	ARI	ACC	F1
SCISE	48.42 $\pm$ 0.08	37.66 $\pm$ 0.47	44.13 $\pm$ 0.59	31.52 $\pm$ 0.41	87.92 $\pm$ 0.12	90.97 $\pm$ 0.08	91.75 $\pm$ 0.24	86.48 $\pm$ 0.41	55.61 $\pm$ 0.02	25.65 $\pm$ 0.67	43.88 $\pm$ 0.39	27.82 $\pm$ 0.10
w/o SECC	48.15 $\pm$ 0.07	36.95 $\pm$ 0.37	44.09 $\pm$ 0.21	31.14 $\pm$ 0.37	87.45 $\pm$ 0.04	90.72 $\pm$ 0.11	91.14 $\pm$ 0.18	84.51 $\pm$ 0.52	55.33 $\pm$ 0.01	24.27 $\pm$ 0.01	42.89 $\pm$ 0.01	27.63 $\pm$ 0.01
w/o CSampE	45.31 $\pm$ 0.33	21.15 $\pm$ 1.11	32.73 $\pm$ 0.95	26.10 $\pm$ 0.61	85.97 $\pm$ 0.16	80.19 $\pm$ 1.37	85.66 $\pm$ 1.10	83.40 $\pm$ 0.57	53.21 $\pm$ 0.67	21.76 $\pm$ 1.17	39.40 $\pm$ 1.06	26.13 $\pm$ 0.45
w/o StructCL	47.55 $\pm$ 0.17	32.97 $\pm$ 0.31	41.42 $\pm$ 0.45	30.34 $\pm$ 0.73	88.20 $\pm$ 0.03	91.27 $\pm$ 0.08	91.57 $\pm$ 0.13	85.82 $\pm$ 0.19	54.78 $\pm$ 0.24	22.59 $\pm$ 0.26	40.62 $\pm$ 0.18	26.71 $\pm$ 0.11
w/o SECC & StructCL	46.91 $\pm$ 0.21	28.08 $\pm$ 1.77	38.35 $\pm$ 0.65	29.82 $\pm$ 1.01	87.46 $\pm$ 0.11	90.36 $\pm$ 0.27	90.69 $\pm$ 0.28	83.42 $\pm$ 0.90	54.80 $\pm$ 0.01	22.35 $\pm$ 0.01	40.51 $\pm$ 0.01	26.73 $\pm$ 0.02
w/o CSampE & StructCL	41.49 $\pm$ 0.37	15.57 $\pm$ 0.10	28.25 $\pm$ 0.77	22.95 $\pm$ 0.58	85.44 $\pm$ 0.04	80.09 $\pm$ 0.54	85.71 $\pm$ 0.84	82.61 $\pm$ 0.50	48.53 $\pm$ 0.22	15.83 $\pm$ 1.65	33.92 $\pm$ 1.74	23.07 $\pm$ 0.26

Table 4: End-to-end runtime analysis of SCISE.

Dataset	SECC	epoch	clustering	total
Photo	0.30 s	1.37 s	3.34 s	14.58 min
Computers	0.33 s	4.92 s	3.48 s	5.10 min
Pubmed	0.35 s	2.73 s	3.55 s	2.87 min
Ogbn-arxiv	1.24 s	73.73 s	73.26 s	1.64 h
Reddit	2.58 s	40.97 s	96.59 s	1.80 h
Ogbn-products	10.75 s	1028.96 s	245.21 s	7.54 h

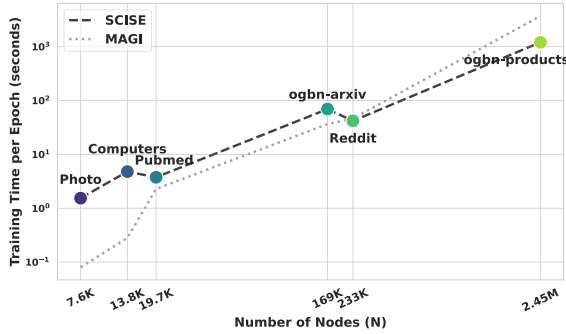


Figure 3: Runtime comparison between SCISE and MAGI.

contrast, SCISE scales efficiently to millions of nodes by decoupling the community-aware expansion from the backpropagation pipeline. This allows our framework to maintain global topological awareness even under strict memory constraints—a critical advantage for real-world deployment. Notably, this demonstrates that our framework preserves global topological awareness under memory constraints—a capability lacking in existing scalable methods. Collectively, these results demonstrate that the synergy between structural entropy constraints and community-aware sampling is fundamental for high-performance graph clustering at scale.

Furthermore, Figure 3 reports the average per-epoch runtime of SCISE and MAGI. As the graph size increases from thousands to millions of nodes, the runtime of SCISE exhibits a near-linear growth trend, demonstrating good scalability on large-scale graphs.

Table 5: Memory usage breakdown of SCISE (GPU A and GPU R denote allocated and reserved GPU memory, respectively).

Dataset	SECC			Train		
	GPU A	GPU R	CPU	GPU A	GPU R	CPU
Photo	41.77 MB	78.00 MB	1.07 MB	4.30 GB	14.62 GB	62.46 MB
Computers	87.88 MB	136.00 MB	1.30 MB	3.43 GB	13.00 GB	30.34 MB
Pubmed	18.24 MB	36.00 MB	1.50 MB	5.79 GB	16.29 GB	23.28 MB
Ogbn-arxiv	424.40 MB	591.99 MB	17.19 MB	19.01 GB	46.84 GB	356.35 MB
Reddit	11.30 GB	11.90 GB	2.46 MB	11.03 GB	31.17 GB	475.02 MB
Ogbn-products	18.71 GB	24.66 GB	113.06 MB	39.43 GB	46.79 GB	316.83 MB

Although SCISE incurs a moderate overhead on small datasets due to community-aware batch expansion, it becomes increasingly competitive on larger graphs and achieves lower per-epoch runtime than MAGI on Reddit and Ogbn-products. To provide a complete end-to-end efficiency analysis, Table 4 further reports the preprocessing (SECC), training, clustering, and total runtime costs. The results show that SECC introduces only a negligible one-time overhead, requiring less than 3 seconds on all datasets except Ogbn-products (10.75 s), which is insignificant compared with the overall training time. Even on Ogbn-products with 2.45 million nodes, the entire pipeline can be completed within 7.54 hours. Table 5 reports the memory consumption of both SECC and the training stage. Despite operating on large-scale graphs, the memory footprint of SECC remains substantially lower than that of representation learning, confirming that the proposed community discovery step is not the computational bottleneck of the framework. These results demonstrate that SCISE achieves strong clustering performance while maintaining practical end-to-end efficiency and scalability for million-scale graph clustering.

5.3 Ablation Study: Dissecting the Synergistic and Complementary Effects

To evaluate the contribution of each component, we construct three primary variants ("w/o SECC", "w/o CSampE", and "w/o StructCL") and two joint variants ("w/o SECC & StructCL" and "w/o CSampE & StructCL"). Specifically, "w/o SECC" replaces community-constrained structural entropy minimization with unconstrained optimization, while "w/o StructCL" computes contrastive loss directly on the raw

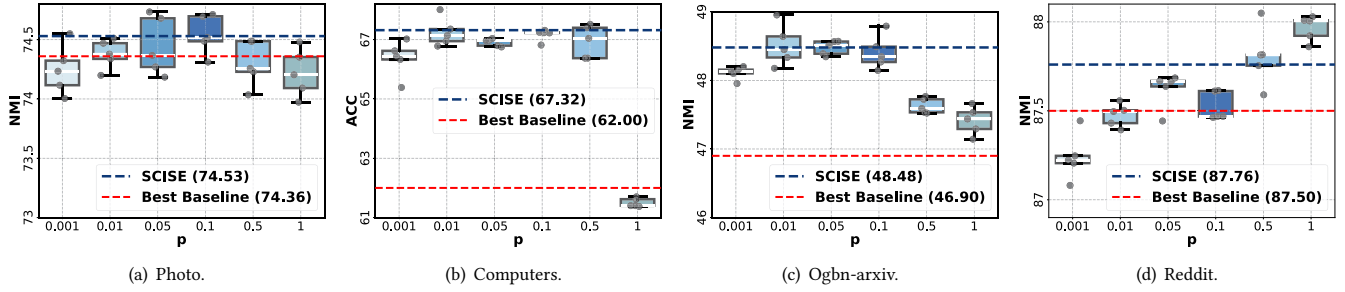


Figure 4: Sensitivity analysis of merge speed p on Photo, Computers, Ogbn-arxiv, and Reddit datasets.

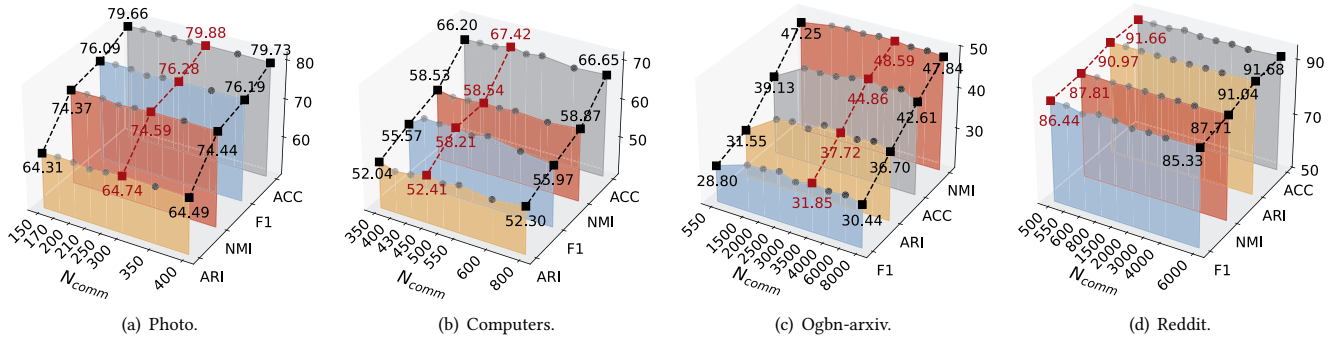


Figure 5: Sensitivity analysis of community number N_{comm} on Photo, Computers, Ogbn-arxiv, and Reddit datasets.

Table 6: Sensitivity analysis of learning rate lr with NMI.

lr	0.05	0.01	0.005	0.001	0.0005	0.0001
Photo	71.65	74.31	74.44	72.69	72.38	72.50
Computers	57.58	58.02	56.34	58.62	58.40	57.89
Pubmed	23.75	34.86	35.62	27.97	27.35	28.61
Ogbn-arxiv	48.40	48.35	48.18	48.40	48.24	47.77
Reddit	88.31	88.34	88.06	88.26	88.36	87.91
Ogbn-products	55.33	55.19	55.62	55.35	55.49	55.85

adjacency matrix. The results in Table 3 show that all components contribute positively to the final performance. (1) Effect of CSampE. Removing CSampE leads to the largest degradation on large-scale graphs. For example, the ARI on Ogbn-arxiv drops from 37.66% to 21.15%, indicating that community-aware batch expansion is crucial for alleviating structural isolation and preserving long-range topology during mini-batch training. (2) Effect of StructCL. StructCL consistently improves clustering quality, particularly on medium-scale datasets. On Pubmed, removing StructCL decreases ARI from 36.33% to 30.32%, demonstrating the benefit of incorporating refined structural affinities into contrastive learning. (3) Effect of SECC. Although the impact of SECC is generally smaller than that of CSampE, it provides a stable structural prior for downstream learning. Comparing "w/o StructCL" and "w/o SECC & StructCL", the additional removal of SECC further reduces Pubmed NMI from 28.59% to 25.62%, suggesting that community-constrained structural entropy helps maintain globally consistent partitions.

5.4 Stability Analysis: Assessing the Invariance to Hyper-parameter Variations

To further evaluate the robustness of SCISE, we analyze the sensitivity of seven key hyperparameters, including the merge speed p , community number N_{comm} , learning rate lr , random walk parameters (w_t , w_l), expansion threshold θ , and initial batch size B . Overall, SCISE exhibits stable performance across a wide range of settings, indicating low hyperparameter sensitivity and minimal tuning effort.

Merge speed p : Figure 4 shows that the performance remains stable across $p \in \{0.001, 0.01, 0.05, 0.1, 0.5, 1\}$. Only extreme values lead to noticeable fluctuations on a few datasets, while most settings consistently outperform the strongest baseline.

Community number N_{comm} : As shown in Figure 5, the performance curves remain relatively flat across a broad range of community numbers. Except for a localized drop on Ogbn-arxiv, the model maintains stable clustering quality, suggesting that SCISE is largely insensitive to the exact choice of N_{comm} .

Learning rate lr : As shown in Table 6, SCISE demonstrates stable convergence across typical learning rate intervals, which further confirms that the framework is not prone to overfitting specific parameter combinations.

Random walker w_t and w_l : Tables 7 and 8 show only minor performance variations when changing the number and length of random walks, suggesting that the structural affinity construction is robust to different walk configurations.

Table 7: Sensitivity analysis of hyper w_t with NMI.

w_t	10	20	30	40	50	60	80	100
Photo	73.87	74.00	74.21	74.12	74.40	74.42	74.18	74.15
Computers	56.57	58.40	57.82	56.50	56.54	56.60	56.42	56.50
Pubmed	35.36	35.58	35.62	35.54	35.73	35.21	35.73	35.35
Arxiv	48.28	48.34	48.35	48.50	47.98	48.53	48.35	48.27
Reddit	86.31	87.46	88.35	88.13	88.15	88.08	87.75	87.72
Products	54.63	55.34	55.52	OOM	OOM	OOM	OOM	OOM

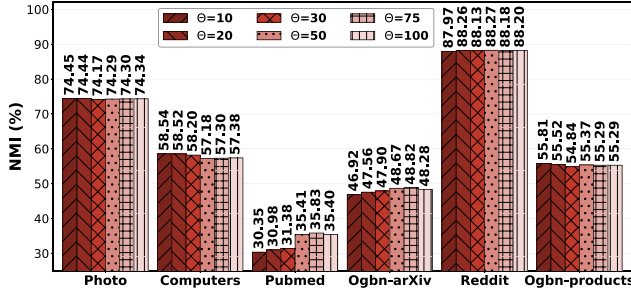


Figure 6: Sensitivity analysis of threshold θ with NMI.

Community size threshold θ : Figure 6 shows that SCISE maintains stable performance for $\theta \in [10, 100]$ on most datasets. Only Pubmed exhibits a modest improvement with larger thresholds, while the overall trend remains consistent.

Initial Batch size B : Figure 7 indicates that SCISE consistently outperforms the strongest baseline under different batch sizes. This suggests that the framework is not strongly dependent on a specific batch configuration and can adapt to diverse hardware constraints.

Overall, SCISE achieves consistently strong performance across a wide range of hyperparameter settings. These results indicate that the framework requires little parameter tuning in practice, making it suitable for large-scale graph clustering scenarios where prior knowledge is limited.

5.5 Robustness against Noisy Community Priors

Since SCISE relies on the community partitions generated by SECC, we further evaluate its robustness under imperfect community priors. Specifically, we inject two types of perturbations: **random noise**, which randomly reassigns a fraction of nodes to different communities, and **neighbor noise**, which reassigns nodes according to the communities of their neighbors. Table 9 reports the results under varying noise ratios using the aggregated score (NMI+ARI+ACC+F1). Overall, SCISE exhibits strong robustness to both perturbation strategies. Even when 20% of the nodes are assigned to incorrect communities, the performance degradation remains marginal on most datasets. In particular, the three large-scale benchmarks (Ogbn-arxiv, Reddit, and Ogbn-products) remain highly stable under both random and neighbor noise, indicating that moderate inaccuracies in community assignments have little impact on the learned representations.

We further consider an extreme setting with completely random community assignments (100% random noise). Although performance decreases on all datasets, SCISE remains competitive and

Table 8: Sensitivity analysis of hyper w_l with NMI.

w_l	2	3	4	5	6	7	8	10
Photo	74.49	74.01	73.43	72.57	71.95	71.28	70.88	69.94
Computers	56.20	56.81	57.13	57.26	59.02	58.03	58.52	58.24
Pubmed	35.62	33.67	31.34	30.87	33.25	33.10	33.54	33.99
Ogbn-arxiv	48.60	48.40	48.46	48.01	47.93	47.97	48.02	47.98
Reddit	87.56	88.25	88.28	87.99	87.72	87.34	87.04	86.58
products	54.18	55.24	55.27	55.60	55.77	55.96	OOM	OOM

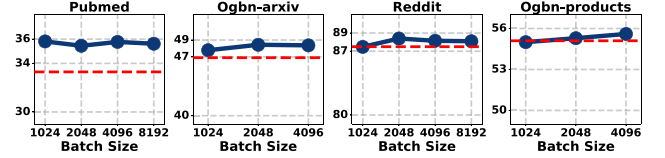


Figure 7: Sensitivity analysis of initial batch size B with NMI.

still achieves performance comparable to or better than MAGI on several benchmarks. This observation suggests that community priors serve as beneficial structural guidance rather than a strict prerequisite for effective clustering. The relatively small degradation on Photo can be attributed to its limited graph size, where the expanded batches already cover a substantial portion of the graph and therefore preserve sufficient structural context even under random community assignments. Overall, these results demonstrate that SCISE benefits from community-aware priors but does not critically depend on highly accurate SECC outputs. Combined with the hyperparameter sensitivity analysis in Figures 4 and 5, which shows that SCISE remains stable across a wide range of SECC configurations, we conclude that the effectiveness of SCISE is robust to both imperfect community partitions and variations in the SECC preprocessing stage.

5.6 Robustness against Sparsity

To further evaluate whether SCISE can alleviate structural isolation under sparse graph conditions, we progressively remove 10%–80% of the edges from the original graph and report the results in Table 10. In addition, Table 11 summarizes the statistics of mini-batches before and after CSampE expansion. The results reveal two important observations. **First, CSampE substantially enriches the structural context available to each mini-batch.** Across all datasets, the expanded batches contain significantly more nodes and edges than the original sampled batches. For example, on Ogbn-arxiv and Ogbn-products, the average batch size increases from 2,048 nodes to 28,549 and 22,788 nodes, respectively, while the average degree increases by more than 6 $\times$ and 12 $\times$. This demonstrates that CSampE effectively reconnects fragmented local neighborhoods and recovers structural information that would otherwise be lost during mini-batch sampling. **Second, SCISE exhibits strong robustness under severe graph sparsification.** Even after removing up to 80% of the edges, the performance degradation remains relatively moderate on most datasets. Compared with MAGI, SCISE consistently maintains higher NMI and ACC scores, with the advantage becoming particularly evident on sparse graphs such as Pubmed, where MAGI suffers a dramatic collapse while SCISE

Table 9: Performance comparison under varying community noise ratios (NMI+ARI+ACC+F1).

Dataset	SCISE	MAGI	random noise								neighbor noise							
Rate	0%	0%	1%	2%	4%	6%	8%	10%	20%	100%	1%	2%	4%	6%	8%	10%	20%	
Photo	294.96	285.00	290.44	289.06	290.61	294.60	293.80	288.44	292.33	291.29	289.94	290.39	292.08	294.73	294.03	290.84	288.12	
Computers	237.03	224.80	236.71	244.05	234.89	234.47	236.23	231.30	227.53	209.67	235.22	236.31	236.47	234.46	239.82	237.28	231.89	
Pubmed	216.32	189.96	186.69	182.17	195.14	188.09	195.83	198.05	189.93	208.10	196.47	181.42	196.02	183.79	204.18	195.00	208.01	
Ogbn-arxiv	161.73	143.30	160.24	160.91	159.38	159.05	158.88	159.27	159.76	143.42	159.18	161.59	157.73	158.21	161.54	158.12	158.30	
Reddit	357.12	354.60	359.11	358.43	357.86	357.38	357.58	357.04	357.00	330.97	356.96	357.73	358.21	357.64	357.84	356.84	356.63	
Ogbn-products	152.95	146.70	151.82	150.50	152.31	152.06	151.27	150.23	151.81	136.34	148.87	151.62	149.11	149.82	148.34	150.03	151.27	

Table 10: Performance comparison (NMI and ACC) under varying edge deletion ratios.

		NMI										ACC									
Rate		0%	10%	20%	30%	40%	50%	60%	70%	80%	0%	10%	20%	30%	40%	50%	60%	70%	80%		
Photo	SCISE	74.53	74.42	74.29	73.90	74.09	74.33	74.14	73.87	72.82	79.71	79.78	79.77	79.67	79.61	79.69	79.73	79.54	78.62		
	w/o SECC	74.41	74.03	74.15	74.27	74.14	74.24	74.12	73.93	72.83	79.74	79.76	79.67	79.67	79.74	79.72	79.57	79.57	78.76		
	MAGI	71.60	70.06	70.03	68.76	69.67	68.68	68.86	70.17	68.22	79.00	79.06	78.80	77.93	78.46	77.97	78.07	79.14	77.98		
Computers	SCISE	58.46	58.24	57.84	58.22	57.97	57.67	57.08	56.95	56.72	67.32	66.79	66.98	66.99	66.84	66.37	65.98	66.06	65.10		
	w/o SECC	58.11	58.37	58.11	58.27	57.66	57.46	57.13	56.19	56.24	67.05	67.26	66.96	67.16	66.71	66.05	66.09	65.41	64.95		
	MAGI	59.20	59.01	59.40	59.50	59.60	58.71	58.79	59.36	59.91	62.00	62.02	61.73	61.44	62.62	61.52	61.93	61.67	61.08		
Pubmed	SCISE	35.77	32.91	32.36	32.02	32.35	32.32	31.66	30.74	24.29	72.38	70.62	70.04	70.22	70.30	70.03	69.87	69.24	63.34		
	w/o SECC	35.59	32.00	32.48	32.71	32.47	32.25	31.18	31.61	31.59	72.31	69.96	70.66	70.57	70.27	70.37	69.61	69.70	69.15		
	MAGI	29.37	28.98	28.97	29.14	28.69	26.17	23.47	20.00	4.36	66.23	65.94	66.14	65.90	65.59	62.14	57.53	51.61	43.96		
arxiv	SCISE	48.42	48.11	48.09	46.12	45.52	45.15	44.24	43.34	42.16	44.13	43.43	43.33	38.21	34.70	34.99	33.69	33.54	34.14		
	w/o SECC	48.15	47.87	47.55	47.42	46.28	45.99	45.13	43.30	41.46	44.09	43.82	43.19	43.12	41.15	39.67	37.51	34.46	30.57		
	MAGI	46.90	45.86	46.35	46.17	45.66	44.93	43.80	42.72	40.69	38.80	37.62	37.09	37.01	35.59	34.22	33.00	31.90	29.74		
Reddit	SCISE	87.92	87.62	87.47	87.51	87.45	87.41	87.29	86.98	86.82	91.75	91.41	91.29	91.46	91.20	91.00	90.77	90.86	90.78		
	w/o SECC	87.45	85.33	84.94	84.64	85.58	85.40	84.43	84.54	84.35	91.14	83.38	82.70	81.14	84.64	82.20	82.35	82.14	82.35		
	MAGI	87.50	87.84	87.34	87.29	87.09	87.44	86.81	87.15	87.12	91.10	91.79	90.40	89.44	89.97	90.31	88.80	90.14	90.45		
products	SCISE	55.61	55.88	55.75	55.48	55.33	55.95	55.69	55.11	55.43	43.88	43.77	42.96	44.21	42.07	42.84	42.90	42.64	41.96		
	w/o SECC	55.33	56.49	55.64	55.59	55.51	55.76	55.12	55.52	55.20	42.89	44.05	42.86	43.86	42.42	42.57	42.26	42.21	42.86		
	MAGI	55.10	55.66	55.52	56.05	54.64	55.12	54.68	54.94	54.43	42.50	41.08	40.59	41.62	39.63	40.54	41.54	40.59	38.65		

Table 11: Statistics of the batch before and after CSampE.

Dataset	AVG. Node	AVG. Edge	AVG. Degree
Photo	2,048 → 6,609	19,475 → 226,000	9.51 → 34.20
Computers	2,048 → 9,713	12,868 → 374,281	6.28 → 8.53
Pubmed	2,048 → 14,863	3,002 → 72,280	1.47 → 4.86
Ogbn-arxiv	2,048 → 28,549	2,384 → 204,633	1.16 → 7.17
Reddit	2,048 → 12,127	10,906 → 1,442,950	5.33 → 118.99
Ogbn-products	2,048 → 22,788	2,135 → 284,661	1.04 → 12.49

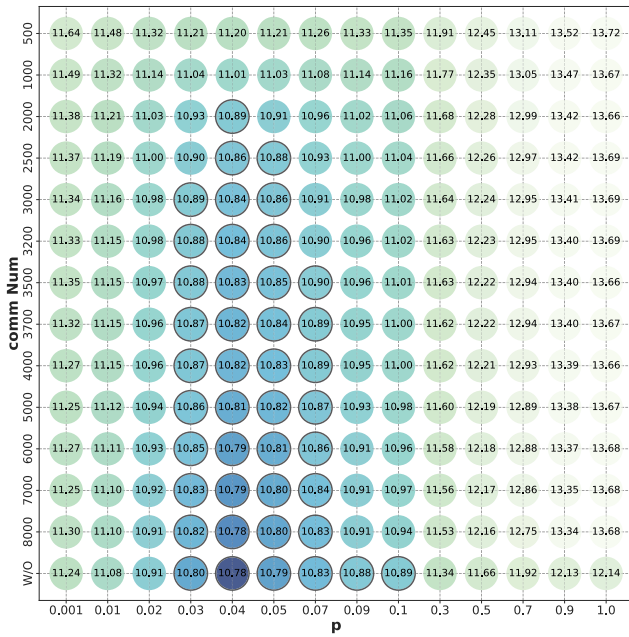
preserves substantially better clustering quality. These results indicate that the community-aware structural expansion introduced by SCISE effectively mitigates the loss of connectivity caused by both mini-batch sampling and graph sparsification, thereby improving resilience against structural fragmentation. Interestingly, SCISE also remains consistently more stable than its w/o SECC variant on several sparse graphs (e.g., Reddit and Ogbn-arxiv), suggesting that the structural prior provided by SECC further improves resilience against graph fragmentation.

5.7 Case Study: Balancing SE Minimization and Community Cohesion

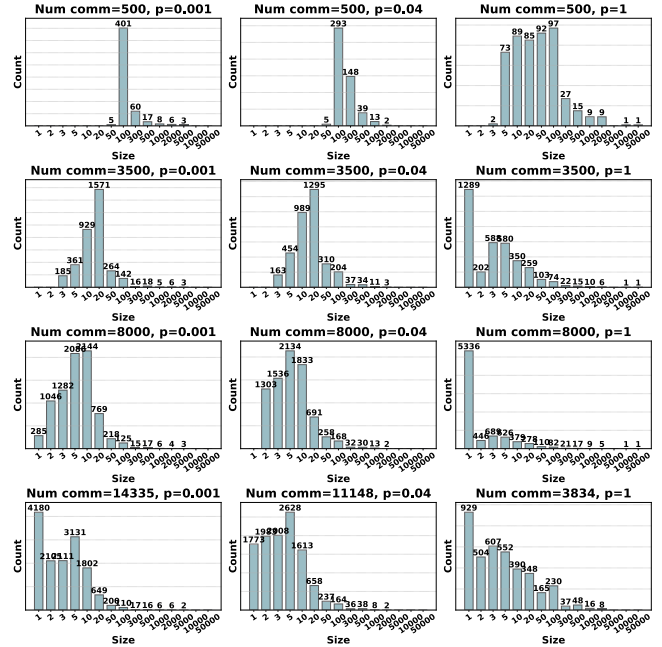
To intuitively understand how SECC minimizes structural entropy while enforcing global constraints, we conduct a comprehensive case study on the Ogbn-arxiv dataset. Figure 8 illustrates the structural entropy landscapes (Figure 8(a)) alongside the corresponding community size distributions (Figure 8(b)) under varying N_{comm} and merge speeds p . Specifically, for Ogbn-arxiv, N_{comm} ranges

from 500 to 8,000, including a "w/o" setting representing the unconstrained scenario. As shown in Figure 8(b), without SECC constraints, the number of formed communities can skyrocket to 14,335, 11,148, or 3,834 for different p values. Notably, the minimum constraint of 136 is dictated by the 115 inherent isolated nodes in the dataset and the connectivity between communities, which define the lower bound of partitionable structures.

The heatmaps visualize the structural entropy values, where deeper shades represent lower entropy, indicating a more stable and cohesive partition. A key observation is that structural entropy remains remarkably low and stable across a wide range of N_{comm} when modulated by p . Specifically, on the Ogbn-arxiv dataset, when p is tuned between 0.03 and 0.07, $N_{comm} > 3,000$ consistently maintains the structural entropy below 10.9. This quantitative evidence proves that by executing the SECC module, SCISE sacrifices a negligible amount of structural entropy precision to secure substantial community-level cohesion. By constraining the solution space, SECC guides the optimization process toward partitions that are both structurally stable and semantically meaningful. The grid of histograms further reveals how community boundaries are formed under different configurations. **(1) Over-fragmentation without Constraints:** As depicted in the bottom rows of Figure 8(b), unconstrained optimization continuously favors local entropy reduction, producing a large number of tiny communities (size < 5). In this case, community boundaries are excessively fragmented and fail to reflect meaningful mesoscopic structures. **(2) Community Collapse under Excessive Constraint:** Conversely, overly strict



(a) Heatmap of Structural Entropy on Ogbn-arxiv.



(b) Histogram of community size on Ogbn-arxiv.

Figure 8: Evolution of structural entropy and community distribution on Ogbn-arxiv.

constraints (top rows) force many structurally distinct regions to be merged together. The resulting boundaries become overly coarse, yielding a few dominant communities that obscure the intrinsic organization of the graph. **(3) Balanced Boundary Formation:** In contrast, the partitions generated by SCISE (middle rows) exhibit a much healthier community size distribution, with few fragmented micro-communities and few oversized giant clusters. This indicates that the resulting boundaries emerge near structural bottlenecks where further merging provides little entropy reduction, while preserving sufficient intra-community connectivity. Consequently, SECC achieves a balance between entropy minimization and community cohesion, producing partitions that better align with the underlying graph structure.

6 CONCLUSION

In this paper, we propose SCISE, a scalable unsupervised graph clustering framework designed to alleviate structural isolation in large-scale graph learning. By incorporating the Structural Entropy Community Constraint (SECC), Community-Aware Sampling Expansion (CSampE), and Structured Contrastive Learning (StructCL), SCISE effectively preserves both community cohesion and global topological information during mini-batch training. Extensive experiments on six benchmark datasets demonstrate that SCISE consistently achieves state-of-the-art clustering performance while maintaining strong scalability on graphs containing millions of nodes. Beyond accuracy improvements, our analyses show that SCISE is highly robust to noisy community priors, graph sparsification, and hyperparameter variations. Furthermore, the additional time and memory evaluations confirm that the proposed SECC

preprocessing introduces only a negligible overhead relative to the overall training pipeline, making SCISE practical for real-world large-scale deployments. Overall, our findings highlight the importance of integrating community-level structural information into scalable graph clustering and demonstrate that reliable structural priors can effectively mitigate sampling-induced information loss without sacrificing efficiency or robustness. As future work, we plan to investigate more scalable structural entropy optimization techniques and extend the proposed framework to emerging ultra-large graph benchmarks, such as Ogbn-papers100M. Another promising direction is to develop adaptive community-aware sampling strategies that automatically adjust random-walk length and sampling budgets according to graph structural characteristics.

ACKNOWLEDGMENTS

This work is supported by the NSFC through grants U25B2029, 62322202, and 62432006, Beijing Natural Science Foundation through grant L253021, the Pioneer and Leading Goose R&D Program of Zhejiang through grant 2025C02044, Local Science and Technology Development Fund of Hebei Province Guided by the Central Government of China through grant 254Z9902G, Science Research Project of Hebei Higher Education Institutions under grant CYZD2026005, Shijiazhuang Science and Technology Plan Project through grant 2511301807A, Major Science and Technology Special Projects of Yunnan Province through grants 202502AD080012 and 202502AD080006, and the Academic Excellence Foundation of BUAA for PhD Students.

REFERENCES

- [1] Amit Aflalo, Shai Bagon, Tamar Kashti, and Yonina Eldar. 2023. Deepcut: Unsupervised segmentation using graph neural networks clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 32–41.
- [2] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery* 29, 3 (2015), 626–688.
- [3] Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. 2020. Structural deep clustering network. In *Proceedings of the web conference 2020*. 1400–1410.
- [4] Yuwei Cao, Hao Peng, Angsheng Li, Chenyu You, Zhifeng Hao, and Philip S Yu. 2024. Multi-Relational Structural Entropy. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. 1–15.
- [5] Yuwei Cao, Hao Peng, Zhengtao Yu, and Philip S Yu. 2024. Hierarchical and Incremental Structural Entropy Minimization for Unsupervised Social Event Detection. In *Proceedings of the AAAI conference on artificial intelligence*. 1–13.
- [6] Jianfei Chen, Jun Zhu, and Le Song. 2018. Stochastic Training of Graph Convolutional Networks with Variance Reduction. In *International Conference on Machine Learning*. PMLR, 942–950.
- [7] Mulin Chen, Bocheng Wang, and Xuelong Li. 2024. Deep contrastive graph learning with clustering-oriented guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 11364–11372.
- [8] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 257–266.
- [9] Fnu Devvrit, Aditya Sinha, Inderjit Dhillon, and Prateek Jain. 2022. S3GC: Scalable Self-Supervised Graph Clustering. In *Advances in Neural Information Processing Systems*, Vol. 35. Curran Associates, Inc., 3248–3261.
- [10] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3–5 (2010), 75–174.
- [11] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. 2018. Large-scale learnable graph convolutional networks. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1416–1424.
- [12] Will Hamilton, Zitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).
- [13] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* 33 (2020), 22118–22133.
- [14] Li Huang, Jingyun Zhang, Weikang Gong, Guangjie Zeng, Hao Peng, and Dongsheng Chen. 2026. SECTOR: structural entropy-based learning of spatiotemporal organisation in spatial transcriptomics. *Bioinformatics* 42, 6 (2026), btg367.
- [15] Xiang Huang, Hao Peng, Li Sun, Hui Lin, Chunyang Liu, Jiang Cao, and Philip S Yu. 2025. Structural entropy guided probabilistic coding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 17467–17475.
- [16] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*. OpenReview.net. <https://openreview.net/forum?id=SJU4ayYgl>
- [17] Gayan K Kulatilake, Marius Portmann, and Shekhar S Chandra. 2025. SGC: Self-supervised contrastive graph clustering. *Neurocomputing* 611 (2025), 128629.
- [18] Angsheng Li and Yicheng Pan. 2016. Structural information and dynamical complexity of networks. *IEEE TIT* 62, 6 (2016), 3290–3339.
- [19] Mengyao Li, Zhibang Yang, Xu Zhou, Yixiang Fang, Kenli Li, and Keqin Li. 2025. Clustering on attributed graphs: From single-view to multi-view. *Comput. Surveys* 57, 7 (2025), 1–36.
- [20] Yunfei Liu, Jintang Li, Yuehe Chen, Ruofan Wu, Ericbk Wang, Jing Zhou, Sheng Tian, Shuheng Shen, Xing Fu, Changhua Meng, et al. 2024. Revisiting modularity maximization for graph clustering: A contrastive learning perspective. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1968–1979.
- [21] Yue Liu, Ke Liang, Jun Xia, Xihong Yang, Sihang Zhou, Meng Liu, Xinwang Liu, and Stan Z Li. 2023. Reinforcement graph clustering with unknown cluster number. In *Proceedings of the 31st ACM international conference on multimedia*. 3528–3537.
- [22] Yue Liu, Ke Liang, Jun Xia, Sihang Zhou, Xihong Yang, Xinwang Liu, and Stan Z Li. 2023. Dink-net: Neural clustering on large graphs. In *International conference on machine learning*. PMLR, 21794–21812.
- [23] Yiwei Liu, Jiamou Liu, Zijian Zhang, Liehuang Zhu, and Angsheng Li. 2019. REM: From structural entropy to community structure deception. *Proceedings of the Advances in Neural Information Processing Systems* 32 (2019), 1–11.
- [24] Yue Liu, Jun Xia, Sihang Zhou, Xihong Yang, Ke Liang, Chenchen Fan, Yan Zhuang, Stan Z Li, Xinwang Liu, and Kunlun He. 2022. A survey of deep graph clustering: Taxonomy, challenge, application, and open resource. *arXiv preprint arXiv:2211.12875* (2022).
- [25] Yixin Liu, Yu Zheng, Daokun Zhang, Hongxu Chen, Hao Peng, and Shirui Pan. 2022. Towards unsupervised deep graph structure learning. In *Proceedings of the ACM Web Conference 2022*. 1392–1403.
- [26] Hao Peng, Jingyun Zhang, Xiang Huang, Zhifeng Hao, Angsheng Li, Zhengtao Yu, and Philip S Yu. 2024. Unsupervised Social Bot Detection via Structural Information Theory. *ACM Transactions on Information Systems* 42, 6 (2024), 42.
- [27] Ladislav Rampásek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. 2022. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems* 35 (2022), 14501–14515.
- [28] Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu, Philip S Yu, and Lifang He. 2024. Deep clustering: A comprehensive survey. *IEEE transactions on neural networks and learning systems* 36, 4 (2024), 5858–5878.
- [29] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.
- [30] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868* (2018).
- [31] Xiao Shen, Dewang Sun, Shirui Pan, Xi Zhou, and Laurence T Yang. 2023. Neighbor contrastive learning on learnable graph augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 9782–9791.
- [32] Zhihao Shi, Jie Wang, Fanghua Lu, Hanzhu Chen, Defu Lian, Zheng Wang, Jieping Ye, and Feng Wu. 2024. Label deconvolution for node representation learning on large-scale attributed graphs against learning bias. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 12 (2024), 11273–11286.
- [33] Hamed Shirzad, Ameya Velinger, Balaji Venkatchalam, Danica J Sutherland, and Ali Kemal Sinop. 2023. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*. PMLR, 31613–31632.
- [34] Li Sun, Zhenhao Huang, Hao Peng, Yujie Wang, Chunyang Liu, and Philip S Yu. 2024. LSEnet: Lorentz Structural Entropy Neural Network for Deep Graph Clustering. In *International Conference on Machine Learning*. PMLR, 47078–47104.
- [35] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. 2023. Graph clustering with graph neural networks. *Journal of Machine Learning Research* 24, 127 (2023), 1–21.
- [36] Petar Veličković, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep graph infomax. In *International Conference on Learning Representations*.
- [37] Yifei Wang, Yupan Wang, Zeyu Zhang, Song Yang, Kaiqi Zhao, and Jiamou Liu. 2023. User: Unsupervised structural entropy-based robust graph neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 10235–10243.
- [38] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. Pmlr, 6861–6871.
- [39] Junran Wu, Xueyuan Chen, Bowen Shi, Shangzhe Li, and Ke Xu. 2023. SEGA: Structural Entropy Guided Anchor View for Graph Contrastive Learning. In *Proceedings of the ICML*. PMLR, 1–20.
- [40] Junran Wu, Xueyuan Chen, Ke Xu, and Shangzhe Li. 2022. Structural entropy guided graph hierarchical pooling. In *Proceedings of the International Conference on Machine Learning*. PMLR, 24017–24030.
- [41] Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. 2023. Sgformer: Simplifying and empowering transformers for large-graph representations. *Advances in neural information processing systems* 36 (2023), 64753–64773.
- [42] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.
- [43] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*. PMLR, 478–487.
- [44] Minhui Xie, Hao Peng, Pu Li, Guangjie Zeng, Shuhai Wang, Jia Wu, Peng Li, and Philip S Yu. 2025. Hierarchical superpixel segmentation via structural information theory. In *Proceedings of the 2025 SIAM International Conference on Data Mining (SDM)*. SIAM, 242–251.
- [45] Xihong Yang, Yue Liu, Sihang Zhou, Siwei Wang, Wenxuan Tu, Qun Zheng, Xinwang Liu, Liming Fang, and En Zhu. 2023. Cluster-guided contrastive graph clustering network. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 10834–10842.
- [46] Xihong Yang, Cheng Tan, Yue Liu, Ke Liang, Siwei Wang, Sihang Zhou, Jun Xia, Stan Z Li, Xinwang Liu, and En Zhu. 2023. Convert: Contrastive graph clustering with reliable augmentation. In *Proceedings of the 31st ACM international conference on multimedia*. 319–327.
- [47] Zhenyu Yang, Ge Zhang, Jia Wu, Jian Yang, Quan Z Sheng, Hao Peng, Angsheng Li, Shan Xue, and Jianlin Su. 2023. Minimum entropy principle guided graph neural networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 114–122.
- [48] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale

- recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 974–983.
- [49] Guangjie Zeng, Hao Peng, Angsheng Li, Li Sun, Chunyang Liu, Shengze Li, Yicheng Pan, and Philip S Yu. 2026. Hyperbolic continuous structural entropy for hierarchical clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 40. 28094–28102.
 - [50] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. 2019. GraphSAINT: Graph Sampling Based Inductive Learning Method. In *International Conference on Learning Representations*.
 - [51] Xianghua Zeng, Hao Peng, and Angsheng Li. 2025. Proactive Bot Detection Based on Structural Information Principles. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).
 - [52] Xianghua Zeng, Hao Peng, Angsheng Li, Chunyang Liu, Lifang He, and Philip S. Yu. 2023. Hierarchical State Abstraction based on Structural Information Principles. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, Edith Elkind (Ed.). International Joint Conferences on Artificial Intelligence Organization, 4549–4557.
 - [53] Xianghua Zeng, Hao Peng, Yicheng Pan, Angsheng Li, and Guanlin Wu. 2026. Structural Information-based Hierarchical Diffusion for Offline Reinforcement Learning. *Advances in Neural Information Processing Systems* 38 (2026), 109539–109567.
 - [54] Jingyun Zhang, Hao Peng, Li Sun, Guanlin Wu, Chunyang Liu, and Zhengtao Yu. 2025. Unsupervised graph clustering with deep structural entropy. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 3752–3763.
 - [55] Jingyun Zhang, Hao Peng, Mingdai Yang, and Philip S Yu. 2025. Enhanced Pre-training for Recommendation via Hypergraph Structural Entropy. *ACM Transactions on Information Systems* 44, 2 (2025), 1–48.
 - [56] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *Proceedings of the web conference 2021*. 2069–2080.
 - [57] Dongcheng Zou, Hao Peng, and Chunyang Liu. 2024. A structural information guided hierarchical reconstruction for graph anomaly detection. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 4318–4323.