HYPERGRAPH STRUCTURAL ENTROPY NETWORK FOR ATTRIBUTED HYPERGRAPH CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Attributed hypergraph clustering leverages high-order relationship modeling to deliver discriminative vertex representations for data grouping. However, precise clustering is often hindered by the decoupling of the representation learning and grouping stages. Moreover, prevailing contrastive learning methods rely heavily on meticulously designed augmentations to achieve invariance, without explicitly guaranteeing a cluster-separable embedding space. To address these limitations, we propose the Hypergraph Structural Entropy Network (HSENet), a unified generative clustering framework that directly couples representation learning with precise data grouping. At its core, we formulate hypergraph structural entropy (HSE) to explicitly capture the hierarchical topology of hypergraphs, generalized for edge-dependent vertex weights (EDVW) and equipped with a continuous, almost-everywhere differentiable relaxation via the Lovász extension. Within a joint, end-to-end framework, HSENet couples two objectives: a masked generative self-supervised module to learn vertex semantics and structural topology, and an HSE-guided clustering assigner that optimizes a soft partitioning tree on an enhanced representation hypergraph. Extensive experiments on 8 benchmark datasets demonstrate that HSENet outperforms 10 competing methods, highlighting its clustering superiority.

1 INTRODUCTION

Hypergraphs model high-order, complex relationships utilizing hyperedges that can connect multiple vertices simultaneously Zhou et al. (2006). Hypergraph clustering seeks to partition vertices into groups based on these relationships, with applications ranging from circuit design Li et al. (2024) to spatial genome organization Gong et al. (2024). When vertex attributes are also available, attributed hypergraph clustering jointly exploits hypergraph topology and attribute information to identify clusters. Traditional approaches integrate these two sources of information through hypergraph random walk Li et al. (2023) or modularity maximization Feng et al. (2025b). Neural approaches further provide flexible representations for modeling nonlinear attribute-topology dependencies Lee & Shin (2023); Feng et al. (2025a). Beyond learning such representations, a central challenge is to couple representation learning with cluster assignment through an objective that explicitly accounts for high-order relations and their influence on how vertices are grouped.

Many existing deep approaches learn vertex representations through contrastive objectives and subsequently apply k-means to obtain cluster assignments Wang et al. (2025a;b). This separation presents two limitations. First, the encoder is trained without feedback from the downstream clustering objective, leaving representation learning insufficiently aligned with the final partitioning task. Contrastive learning can also be sensitive to augmentations that alter cluster-relevant information. Second, although the embeddings may encode high-order relations, the k-means objective does not explicitly account for hyperedge relationships during assignment. Although CAHC Ni et al. (2026) couples representation learning with clustering through assignment sharpening, its clustering loss does not explicitly account for how hyperedges are partitioned. These limitations motivate a joint framework in which an explicit hypergraph partitioning objective guides both representation learning and cluster assignment by evaluating how hyperedges are split across candidate clusters.

To address the limitations of existing methods, we propose the Hypergraph Structural Entropy Network (HSENet), a unified generative approach guided by hypergraph structural entropy (HSE) for

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.

attributed hypergraph clustering. Building on graph structural entropy Li & Pan (2016), we formulate HSE to evaluate the hierarchical organization of hypergraphs. Minimizing HSE guides cluster assignments according to the overall hypergraph structure. We formulate HSE for hypergraphs with edge-dependent vertex weights (EDVW), allowing it to account for varying vertex membership strengths within hyperedges. Using the Lovász extension, we derive a continuous relaxation that is differentiable almost everywhere with respect to soft assignments, enabling gradient-based optimization of the partitioning objective over multiple levels of the hypergraph hierarchy.

Within HSENet, a shared encoder learns vertex semantics and high-order co-membership patterns through masked feature reconstruction and hyperedge filling. To incorporate the learned attribute-topology dependencies into the partitioning objective, these representations induce weighted hyperedges whose incidence weights reflect learned vertex-hyperedge affinities. Fusing these hyperedges with the observed topology yields an enhanced EDVW hypergraph. A hierarchical assigner learns a soft partitioning tree by minimizing the relaxed HSE on this enhanced hypergraph. The generative and HSE objectives jointly optimize the shared encoder and hierarchical assigner, allowing explicit high-order structural information to guide both representation learning and cluster assignment.

We summarize the main contributions as follows:

• **Hypergraph Structural Entropy:** We formulate HSE as a unified objective for hierarchical hypergraph partitioning while accommodating edge-dependent vertex weights. We further derive a continuous Lovász relaxation, enabling gradient-based partition learning (Section 4).

• **Unified Architecture:** We propose HSENet, a generative framework with coupled representation learning, topology enhancement, and HSE-guided hierarchical assignment. Joint optimization allows this objective to guide both representation learning and cluster assignment (Section 5).

• **Empirical Evaluation:** Experiments on 8 datasets against 10 baselines demonstrate that HSENet outperforms state-of-the-art models, validating the effectiveness of HSENet [1] (Section 6).

2 Related Work

Attributed Hypergraph Clustering. Without node attributes, hypergraph clustering partitions vertices using high-order connectivity through spectral cuts Zhou et al. (2006), ratio cuts Veldt et al. (2020), PageRank Takai et al. (2020), block models Ruggeri et al. (2023), or modularity Chodrow et al. (2021). Attributed hypergraph clustering further incorporates vertex features to obtain clusters that are both structurally cohesive and semantically consistent. Traditional methods combine topology and attributes through joint matrix factorization Du et al. (2019); Whang et al. (2020), KNN-augmented random walks Li et al. (2023), or integrated modularity maximization Feng et al. (2025b). Recent deep methods instead learn clustering-oriented vertex representations: HCN Wang et al. (2025b) aligns semantic similarities with hypergraph structure, while HCN-PAI Wang et al. (2025a) additionally imputes missing attributes and performs node- and hyperedge-level contrastive learning. Both obtain final clusters using k-means. CAHC Ni et al. (2026) jointly optimizes representations and cluster assignments through centroid-based self-training. Its assignment objective, however, does not explicitly evaluate hyperedge partition costs. HSENet introduces an HSE-based partitioning objective that guides both representation and assignment learning.

Self-Supervised Hypergraph Learning. Contrastive and generative learning are two major paradigms for self-supervised hypergraph representation learning. HyperGCL learns augmentations for contrastive learning Wei et al. (2022), while TriCL aligns node-, hyperedge-, and membership-level representations Lee & Shin (2023). HyFi preserves topology during augmentation and exploits shared-group information to define weak positive pairs Roh et al. (2024). SE-HSSL combines correlation-based node- and group-level objectives with hierarchical membership contrast Li et al. (2026). On the generative side, HypeBoy uses masked feature reconstruction for encoder warm-up, followed by hyperedge filling Kim et al. (2024). SS-HT combines masked feature reconstruction with structure-aware contrastive learning Feng et al. (2025a). Whereas these methods primarily target representation learning, HSENet couples generative self-supervision with an HSE objective that explicitly evaluates hyperedge splitting costs to guide representation and assignment learning.

[1] Source code is provided at https://anonymous.4open.science/r/HSENet-380D.

Structural Entropy. Structural entropy quantifies the uncertainty of a graph under a hierarchical encoding tree and has become a principled objective for revealing community structure Li & Pan (2016); Su et al. (2025). Subsequent extensions go beyond homogeneous graphs, including multi-relational structural entropy Cao et al. (2024) and incremental structural entropy on dynamic graphs Yang et al. (2024). On graphs, SE has been made differentiable for neural clustering: LSEnet learns a partitioning tree in hyperbolic space by minimizing differentiable structural information Sun et al. (2024); DeSE couples soft hierarchical assignments with topology enhancement Zhang et al. (2025a); and ASIL further integrates augmented structural information with conductance analysis Sun et al. (2026). SCISE scales this line to large graphs by constraining SE minimization with community-aware sampling Zhang et al. (2026). EPRHSE applies structural entropy to a clique-expanded representation for recommendation pretraining Zhang et al. (2025b). Our formulation instead evaluates EDVW hyperedge splitting costs within a hierarchical partitioning objective.

3 PRELIMINARIES

Hypergraph Notation. We use an edge-dependent vertex weight (EDVW) formulation Chitra & Raphael (2019) to represent attributed hypergraphs as $H = (V, E, \mathbf{X}, \omega, \gamma)$, where $V = \{v_i\}_{i=1}^n$ and $E = \{e_j\}_{j=1}^m$ are the vertex and hyperedge sets, $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the vertex feature matrix, $\omega : E \to \mathbb{R}_{>0}$ assigns hyperedge weights, and $\gamma_e(v) > 0$ specifies the contribution of $v \in e$ to e. The weighted incidence matrix $\mathbf{H} \in \mathbb{R}_{\geq 0}^{n\times m}$ has entries $\mathbf{H}_{ij} = \gamma_{e_j}(v_i)$ if $v_i \in e_j$ and 0 otherwise. Setting $\gamma_e(v) = 1$ for every incidence recovers the standard hyperedge-weighted hypergraph model with binary incidence. Let $\Gamma_j = \sum_{i=1}^n \mathbf{H}_{ij}$ be the total EDVW mass of e_j. The vertex degree matrix $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n)$ and the volume of $S \subseteq V$ are:

$$d_i = \sum\nolimits_{j=1}^m \omega(e_j)\mathbf{H}_{ij}, \quad \mathrm{vol}(S) = \sum\nolimits_{v_i\in S} d_i, \quad \mathrm{vol}(H) = \mathbf{1}_n^\top \mathbf{D}\mathbf{1}_n. \tag{1}$$

Hypergraph Cut and Conductance. For $S \subseteq V$, let $x_j(S) = \sum_{v_i\in S} \mathbf{H}_{ij}$ be the EDVW mass of e_j contained in S. We assume all hyperedges are nonempty, so $\Gamma_j > 0$ and $0 \leq x_j(S) \leq \Gamma_j$. A nonnegative splitting function $\Phi(x, \Gamma)$ satisfies $\Phi(0, \Gamma) = \Phi(\Gamma, \Gamma) = 0$ and assigns a penalty according to how a hyperedge is divided Veldt et al. (2020; 2023). Representative choices are:

$$\begin{aligned} &\Phi_{\mathrm{AON}}(x, \Gamma) = \mathbb{I}(0 < x < \Gamma), && \Phi_{\mathrm{Lin}}(x, \Gamma) = \min\{x, \Gamma - x\}, \\ &\Phi_{\delta\text{-Lin}}(x, \Gamma) = \min\{\delta, x, \Gamma - x\}, && \Phi_{\mathrm{Quad}}(x, \Gamma) = [x(\Gamma - x)]/\Gamma. \end{aligned} \tag{2}$$

The generalized cut induced by Φ and its associated conductance are:

$$g_\Phi(S) = \sum\nolimits_{j=1}^m \omega(e_j)\Phi(x_j(S), \Gamma_j), \qquad \mathrm{cond}_\Phi(S) = \frac{g_\Phi(S)}{\min\{\mathrm{vol}(S), \mathrm{vol}(V \setminus S)\}}. \tag{3}$$

The hypergraph conductance $\mathrm{Cond}_\Phi(H)$ is the minimum of $\mathrm{cond}_\Phi(S)$ over all $S \subset V$ such that both S and its complement have positive volume.

Set-Based Structural Entropy. Structural entropy was originally introduced for graphs Li & Pan (2016). We formulate it with cuts and volumes to accommodate both graphs and hypergraphs.

Definition 3.1 (Set-Based Structural Entropy)**.** Let $\mathcal{T}$ be a rooted partitioning tree with $T_\rho = V$ and singleton leaves, where the vertex sets associated with the children of each internal node partition its vertex set. For each non-root node α, let T_α denote its vertex set and α^- its parent. Let g_α be the cut of T_α: the total weight of edges crossing from T_α to $V \setminus T_\alpha$ for graphs, and $g_\Phi(T_\alpha)$ for hypergraphs.

$$\mathcal{H}^{\mathcal{T}}(H) = -\frac{1}{\mathrm{vol}(H)} \sum_{\alpha\in\mathcal{T}\setminus\{\rho\}} g_\alpha \log_2 \frac{\mathrm{vol}(T_\alpha)}{\mathrm{vol}(T_{\alpha^-})}. \tag{4}$$

Its K-dimensional structural entropy is the minimum over partitioning trees of depth at most K.

4 HYPERGRAPH STRUCTURAL ENTROPY

We first express discrete structural entropy through level-wise assignments and establish its equivalence to Eq. 4, then introduce a continuous relaxation for neural optimization.

Level-Wise Discrete HSE. Let $\mathcal{T}_h$ be the set of n_h tree nodes at depth h, where $\mathcal{T}_0 = \{\rho\}$ and $\mathcal{T}_K$ contains the $n_K = n$ vertex leaves, after padding shallower leaves with single-child chains without changing the entropy. For $h = 1, \ldots, K$, let $\mathbf{C}^h \in \{0,1\}^{n_h \times n_{h-1}}$ be row-stochastic, where $\mathbf{C}^h_{\alpha\beta} = 1$ indicates that $\beta \in \mathcal{T}_{h-1}$ is the parent of $\alpha \in \mathcal{T}_h$. Ordering the vertex leaves as $v_1, \ldots, v_n$ and setting $\mathbf{C}^{K+1} = \mathbf{I}_n$, the vertex-to-node assignment at level $h = 0, \ldots, K$ is $\mathbf{S}^h = \mathbf{C}^{K+1}\mathbf{C}^K \cdots \mathbf{C}^{h+1} \in \{0,1\}^{n \times n_h}$.

Definition 4.1 (Discrete Hypergraph Structural Entropy)**.** For an EDVW hypergraph H and a discrete partitioning tree $\mathcal{T}$ of depth K with positive non-root module volumes, the discrete HSE is:

$$\mathcal{H}^{\mathcal{T}}_{\Phi}(H) = -\frac{1}{\mathrm{vol}(H)} \sum_{h=1}^{K} \sum_{\alpha \in \mathcal{T}_h} \sum_{j=1}^{m} \omega(e_j) \Phi\left(\sum_i \mathbf{S}^h_{i\alpha} \mathbf{H}_{ij}, \Gamma_j \right) \log_2 \frac{\sum_i d_i \mathbf{S}^h_{i\alpha}}{\sum_{\beta,i} d_i \mathbf{C}^h_{\alpha\beta} \mathbf{S}^{h-1}_{i\beta}}, \quad (5)$$

where i ranges over $1, \ldots, n$ and β over $\mathcal{T}_{h-1}$.

Theorem 4.1 (Representation Equivalence and Reduction Relations)**.** Let $\mathcal{T}$ be a discrete partitioning tree with positive non-root module volumes, encoded by binary row-stochastic assignments $\{\mathbf{C}^h\}_{h=1}^K$. **(i)** Eq. 5 is exactly the level-wise scalar representation of the set-based structural entropy in Eq. 4 with boundary cost g_Φ. **(ii)** If $\gamma_e(v) = 1$ for every incidence, EDVW-HSE reduces to cardinality-based hypergraph structural entropy. **(iii)** If, in addition, the hypergraph is 2-uniform, then $\mathcal{H}^{\mathcal{T}}_{\Phi}(H) = c_\Phi \mathcal{H}^{\mathcal{T}}(G)$, where G is the induced weighted graph and $c_\Phi = \Phi(1,2)$.

Theorem 4.2 (Connection to Hypergraph Conductance)**.** Let the degree entropy be $\mathcal{H}_{\deg}(H) = -\sum_{i=1}^n \frac{d_i}{\mathrm{vol}(H)} \log_2 \frac{d_i}{\mathrm{vol}(H)}$. The discrete HSE in Eq. 5 satisfies the following bounds. **(i)** For any discrete partitioning tree $\mathcal{T}$ whose non-root modules and their complements have positive volume,

$$\mathcal{H}^{\mathcal{T}}_{\Phi}(H) \geq \mathrm{Cond}_\Phi(H)\big(\mathcal{H}_{\deg}(H) - 1\big). \quad (6)$$

(ii) If every non-root module also satisfies $\mathrm{vol}(T_\alpha) \leq \mathrm{vol}(H)/2$, then the stronger bound holds:

$$\mathcal{H}^{\mathcal{T}}_{\Phi}(H) \geq \mathrm{Cond}_\Phi(H)\mathcal{H}_{\deg}(H), \qquad \text{equivalently} \qquad \mathcal{H}^{\mathcal{T}}_{\Phi}(H)/\mathcal{H}_{\deg}(H) \geq \mathrm{Cond}_\Phi(H). \quad (7)$$

Proofs of Theorems 4.1 and 4.2 are provided in Appendices B.1 and B.2, respectively. Theorem 4.1 establishes consistency with the set-based formulation and recovers cardinality-based hypergraph entropy and graph structural entropy, the latter up to the factor c_Φ. For the splitting functions considered here (with fixed $\delta > 0$), c_Φ is a positive constant depending only on Φ, so graph structural entropy is recovered by dividing by c_Φ. Theorem 4.2 connects discrete HSE to global hypergraph conductance. Under the half-volume condition in part (ii), normalized discrete HSE upper-bounds $\mathrm{Cond}_\Phi(H)$. Since $\mathcal{H}_{\deg}(H)$ is fixed for a given hypergraph, minimizing HSE over trees satisfying this condition minimizes this upper bound, providing a conductance-based motivation for the objective. We next introduce continuous HSE for optimization.

Continuous HSE via the Lovász Extension. To enable neural optimization while preserving discrete hyperedge cut costs, we adopt the Lovász extension, which provides a unified convex, piecewise-linear extension of the induced submodular cut functions Lovász (1983); Li & Milenkovic (2018). This also accommodates Φ_{AON}, whose direct evaluation on soft assignments yields zero gradients almost everywhere.

Specifically, we allow each $\mathbf{C}^h$ to take values in $[0,1]^{n_h \times n_{h-1}}$ while retaining the row-stochastic constraint, yielding fractional vertex-to-module assignments $\mathbf{S}^h$. For each hyperedge e_j, its splitting penalty defines the set function:

$$F_{j,\Phi}(A) = \Phi\big(\textstyle\sum_{v_i \in A} \gamma_{e_j}(v_i), \Gamma_j\big), \quad A \subseteq e_j. \quad (8)$$

The set function $F_{j,\Phi}$ assigns a cost to separating a subset of e_j from its complement within the hyperedge. To extend this cost to soft memberships, consider $\mathbf{x} \in [0,1]^{|e_j|}$ and order the vertices of e_j so that $x_{i_1} \geq \cdots \geq x_{i_{|e_j|}}$, breaking ties arbitrarily. Let $A_r = \{v_{i_1}, \ldots, v_{i_r}\}$. Since $F_{j,\Phi}(\emptyset) = F_{j,\Phi}(e_j) = 0$, its Lovász extension is:

$$\widehat{F}_{j,\Phi}(\mathbf{x}) = \sum\nolimits_{r=1}^{|e_j|-1} (x_{i_r} - x_{i_{r+1}}) F_{j,\Phi}(A_r). \quad (9)$$

This extension agrees with the discrete cut cost on indicator vectors: $\widehat{F}_{j,\Phi}(\mathbf{1}_A) = F_{j,\Phi}(A)$. For the splitting functions in Section 3, $F_{j,\Phi}$ is submodular, so its extension is convex and piecewise linear.

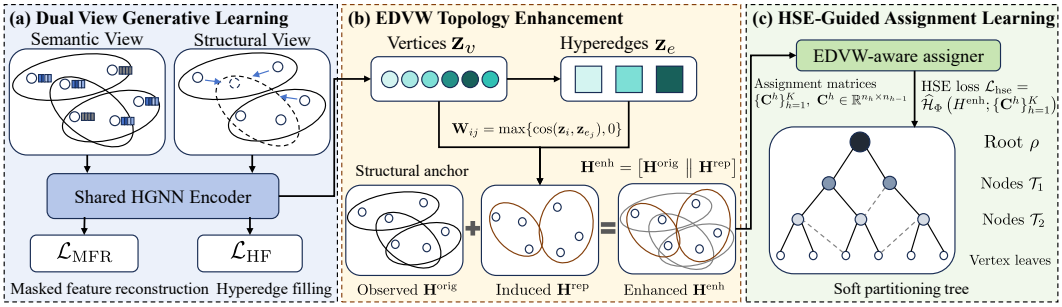


Figure 1: An overview of the HSENet framework.

For module α at level h, we evaluate the extension at $\mathbf{s}^h_{j,\alpha} = (\mathbf{S}^h_{i\alpha})_{v_i \in e_j}$. Replacing the splitting penalties in Eq. 5 with their Lovász extensions yields the relaxed HSE:

$$\widehat{\mathcal{H}}_\Phi\left(H; \{\mathbf{C}^h\}_{h=1}^K\right) = -\frac{1}{\text{vol}(H)} \sum_{h=1}^{K} \sum_{\alpha \in \mathcal{T}_h} \sum_{j=1}^{m} \omega(e_j) \widehat{F}_{j,\Phi}\left(\mathbf{s}^h_{j,\alpha}\right) \log_2 \frac{\sum_i d_i \mathbf{S}^h_{i\alpha}}{\sum_{\beta,i} d_i \mathbf{C}^h_{\alpha\beta} \mathbf{S}^{h-1}_{i\beta}}, \tag{10}$$

where i ranges over $1, \ldots, n$ and β over $\mathcal{T}_{h-1}$. For a fixed H with $\Gamma_j > 0$ and positive volume terms, Eq. 10 is continuous and differentiable almost everywhere in $\{\mathbf{C}^h\}_{h=1}^K$, and recovers Eq. 5 when all assignments are binary.

Theorem 4.3 (Bound of Relaxed HSE)**.** Fix H, Φ, and the level sizes. Let $\{\overline{\mathbf{C}}^h\}$ encode a discrete tree $\mathcal{T}$ with positive non-root module volumes. There exist $r_\mathcal{T}, L_\mathcal{T} > 0$ such that any row-stochastic assignments satisfying $\sum_{h=1}^K \|\mathbf{C}^h - \overline{\mathbf{C}}^h\|_F \le \epsilon \le r_\mathcal{T}$ obey:

$$\left|\widehat{\mathcal{H}}_\Phi(H; \{\mathbf{C}^h\}) - \mathcal{H}^\mathcal{T}_\Phi(H)\right| \le L_\mathcal{T} \epsilon = O(\epsilon). \tag{11}$$

The proof and a conditional bound relative to the optimal discrete tree are given in Appendix B.3.

5 HSENet

Building on the relaxed HSE objective, HSENet comprises three components: generative representation learning, EDVW topology enhancement, and HSE-guided assignment learning (Figure 1). The learned representations enhance the hypergraph topology used for assignment learning, while joint optimization allows HSE to guide both the encoder and the assigner.

5.1 Generative Representation Learning

Effective attributed hypergraph clustering requires integrating vertex features with hypergraph structure. Thus, we seek representations that retain node attributes while reflecting higher-order memberships. To capture both modalities, we adopt masked feature reconstruction Hou et al. (2022) and hyperedge filling Kim et al. (2024), which jointly train a shared HGNN encoder f_{enc}.

Masked Feature Reconstruction. We sample a nonempty vertex subset $\hat{V}$ and replace its features with a learnable mask token, obtaining $\bar{\mathbf{X}}$. The shared encoder and hypergraph decoder compute:

$$\mathbf{Z}_v^{\text{mask}} = f_{\text{enc}}(E, \bar{\mathbf{X}}), \qquad \mathbf{X}'_v = f_{\text{dec}}(E, \bar{\mathbf{Z}}_v^{\text{mask}}), \tag{12}$$

where $\bar{\mathbf{Z}}_v^{\text{mask}}$ replaces the representations at masked positions with a learnable decoder mask token, leaving the others unchanged. This re-masking encourages reconstruction from hypergraph context. We optimize masked feature reconstruction using the scaled cosine error on masked vertices: $\mathcal{L}_{\text{MFR}} = \mathbb{E}_{v_i \sim \mathcal{U}(\hat{V})}\left[(1 - \cos(\mathbf{x}_i, \mathbf{x}'_i))^\rho\right]$, where $\mathbf{x}'_i$ is the reconstructed feature, cos denotes cosine similarity, and $\rho \ge 1$ controls the emphasis on difficult reconstruction targets.

Hyperedge Filling. We randomly zero feature entries and drop hyperedges, obtaining $\mathbf{Z}_v^{\text{aug}} = f_{\text{enc}}(E^{\text{aug}}, \mathbf{X}^{\text{aug}})$ from the perturbed input. For each retained hyperedge e_j with $|e_j| \ge 2$, we treat

each member v_i as a target and mean-pool the remaining members into a query:

$$Q_{ij} = e_j \setminus \{v_i\}, \qquad \mathbf{q}_{ij} = |Q_{ij}|^{-1} \sum\nolimits_{v_k \in Q_{ij}} \mathbf{z}_k^{\mathrm{aug}}. \tag{13}$$

A cosine-similarity softmax over all vertices predicts the target:

$$p(v_i \mid Q_{ij}) = \left[\mathrm{softmax}\big((\cos(\mathbf{z}_k^{\mathrm{aug}}, \mathbf{q}_{ij}))_{k=1}^{n}\big)\right]_i. \tag{14}$$

We optimize hyperedge filling by minimizing the negative log-likelihood of the target vertex: $\mathcal{L}_{\mathrm{HF}} = -\mathbb{E}_{(i,j)\sim\mathcal{U}(\mathcal{P}^{\mathrm{aug}})}\big[\log p(v_i \mid Q_{ij})\big]$, where $\mathcal{P}^{\mathrm{aug}}$ contains all such target-hyperedge pairs and $\mathcal{U}$ denotes the uniform distribution. This encourages compatibility between vertices and their higher-order membership contexts.

Both tasks update the shared encoder through:

$$\mathcal{L}_{\mathrm{rep}} = \lambda_1 \mathcal{L}_{\mathrm{MFR}} + \lambda_2 \mathcal{L}_{\mathrm{HF}}. \tag{15}$$

At each training step, an additional forward pass on the uncorrupted input produces $\mathbf{Z}_v = f_{\mathrm{enc}}(E, \mathbf{X})$ for topology enhancement and hierarchical assignment.

5.2 EDVW Topology Enhancement

HSE evaluated on the observed topology does not explicitly encode representation-derived affinities. We therefore augment this topology with EDVW hyperedges that incorporate learned attribute–topology dependencies into the partitioning objective.

At each forward pass, we mean-pool the uncorrupted representations within each observed hyperedge e_j to obtain a hyperedge prototype $\mathbf{z}_{e_j}$. For each prototype, we compute $\mathbf{W}_{ij} = \max\{\cos(\mathbf{z}_i, \mathbf{z}_{e_j}), 0\}$ and select up to r vertices with the largest positive affinities across the full vertex set. These vertices form a new hyperedge, with their affinities serving as EDVW incidence weights. Discarding empty hyperedges yields the induced incidence matrix $\mathbf{H}^{\mathrm{rep}} \in \mathbb{R}_{\geq 0}^{n \times m_{\mathrm{rep}}}$. We retain the observed relations by concatenating the original and induced incidence matrices:

$$\mathbf{H}^{\mathrm{enh}} = \left[\mathbf{H}^{\mathrm{orig}} \,\middle\|\, \mathbf{H}^{\mathrm{rep}}\right]. \tag{16}$$

Original hyperedges retain their weights $\omega(e_j)$, while $\beta > 0$ controls the weight of each induced hyperedge. We denote the resulting EDVW hypergraph by H^{enh}, with incidence matrix $\mathbf{H}^{\mathrm{enh}}$ and the hyperedge weights specified above. Appendix C.2 provides construction details, and Appendix B.4 gives a conditional conductance guarantee for the fusion.

5.3 HSE-guided Assignment Learning

To make cluster assignments explicitly sensitive to hyperedge partition costs, we learn a soft partitioning tree guided by the relaxed HSE on the enhanced hypergraph H^{enh}, using the level-wise assignments defined in Section 4. The tree has depth $K \geq 2$, with $n_K = n$ vertex leaves, n_1 output clusters, and a single root at level 0. We initialize $\mathbf{Z}^K = \mathbf{Z}_v$ and $\mathbf{H}^K = \mathbf{H}^{\mathrm{enh}} \in \mathbb{R}_{\geq 0}^{n \times M}$, where $M = m + m_{\mathrm{rep}}$.

For $h = K, \ldots, 2$, an EDVW-aware assignment layer $\mathcal{A}^h$, inspired by UniGAT Huang & Yang (2021), predicts parent assignments:

$$\mathbf{C}^h = \mathrm{softmax}\left(\mathcal{A}^h(\mathbf{Z}^h, \mathbf{H}^h, \boldsymbol{\omega}^{\mathrm{enh}})\right) \in [0,1]^{n_h \times n_{h-1}}, \tag{17}$$

where $\boldsymbol{\omega}^{\mathrm{enh}}$ contains the structural hyperedge weights and softmax is applied over candidate parents. The layer $\mathcal{A}^h$ forms hyperedge representations by EDVW-weighted averaging of node features. Each assignment head produces a logit for one candidate parent. Details are in Appendix C.3.

After predicting $\mathbf{C}^h$, we coarsen the structural incidences and sparsify the resulting memberships, while aggregating node representations:

$$\mathbf{H}^{h-1} = \mathcal{S}_b\left((\mathbf{C}^h)^\top \mathbf{H}^h\right), \qquad \mathbf{Z}^{h-1} = \mathrm{diag}\left((\mathbf{C}^h)^\top \mathbf{1}\right)^{-1} (\mathbf{C}^h)^\top \mathbf{Z}^h. \tag{18}$$

Here $\mathcal{S}_b$ retains up to b incidences per node, where b approximates the observed hypergraph's average incidence count per vertex.

Table 1: Performance comparison with baselines. Results (%) are mean±std of 5 independent runs.

Dataset	Metric	JNMF	AHCKA	AHRC	GRAC	TriCL	HyFi	SS-HT	HypeBoy	SE-HSSL	CAHC	HSENet
Cora	NMI	36.5±3.2	53.0±0.0	53.1±0.0	50.9±1.9	55.2±1.3	55.2±2.9	54.3±1.8	42.4±0.6	53.9±2.5	56.4±0.9	**57.5±0.8**
	ARI	28.4±3.9	48.0±0.0	46.9±0.0	43.1±3.9	48.6±1.7	47.5±4.7	47.1±4.1	27.0±0.9	45.6±3.6	50.8±1.6	**52.9±0.9**
	ACC	54.4±4.7	69.7±0.0	71.8±0.0	66.7±4.9	69.2±1.5	70.8±3.3	68.7±3.8	57.4±1.0	67.9±3.9	71.4±1.7	**74.7±1.6**
	F1	50.2±4.6	63.3±0.0	56.1±0.0	59.8±5.1	64.5±2.5	63.8±3.5	62.2±3.9	56.3±0.6	62.0±5.3	69.1±2.8	**69.5±4.0**
Citeseer	NMI	32.8±3.6	42.3±0.0	43.1±0.0	43.1±0.2	44.1±0.3	44.2±0.4	42.8±0.7	40.4±0.8	44.1±1.8	**45.9±1.2**	45.6±0.6
	ARI	32.8±3.9	42.1±0.0	43.6±0.0	43.3±0.4	45.1±0.3	45.1±0.5	42.9±1.1	40.5±1.2	43.4±1.7	46.2±1.2	**47.9±0.7**
	ACC	59.9±4.2	66.5±0.0	70.2±0.0	68.6±0.3	67.3±1.5	69.4±0.2	68.3±0.8	66.8±0.9	68.4±1.4	69.8±0.8	**72.1±0.4**
	F1	55.7±4.3	61.4±0.0	53.9±0.0	63.8±0.3	61.6±2.2	64.2±0.3	63.6±0.6	61.7±0.8	62.7±1.3	**65.1±0.6**	61.7±0.4
Pubmed	NMI	1.6±0.9	22.1±0.0	20.1±0.0	28.8±0.1	30.0±0.8	30.8±3.0	30.0±0.2	31.5±0.5	29.0±3.9	34.4±0.4	**35.2±2.0**
	ARI	1.9±0.9	20.8±0.0	15.1±0.0	24.7±0.2	25.6±2.1	28.8±6.7	26.0±0.3	30.1±1.3	25.8±3.7	32.2±0.3	**40.1±3.0**
	ACC	40.6±1.2	63.3±0.0	42.5±0.0	63.3±0.2	63.5±2.4	67.1±5.5	63.8±0.3	68.7±0.8	65.2±2.6	69.4±0.2	**74.0±2.0**
	F1	38.5±1.1	63.5±0.0	35.8±0.0	62.6±0.2	62.5±2.5	66.0±4.9	63.0±0.3	67.5±0.7	64.6±2.6	69.0±0.2	**72.1±1.8**
DBLP	NMI	29.7±1.2	59.8±0.0	61.0±0.0	51.8±4.1	64.4±2.4	OOM	OOM	OOM	OOT	63.9±2.8	**65.8±2.6**
	ARI	21.2±1.8	53.2±0.0	52.8±0.0	41.7±6.4	55.3±7.0					58.2±5.9	**61.8±5.1**
	ACC	48.6±2.4	64.4±0.0	73.5±0.0	59.1±5.2	70.4±7.0					72.3±8.0	**78.6±5.5**
	F1	35.5±4.5	60.7±0.0	61.8±0.0	60.1±5.2	67.8±7.8					70.4±8.6	**78.5±4.9**
20News	NMI	22.5±2.9	40.9±0.0	27.1±0.0	6.9±0.2	36.4±6.0	31.0±0.7	40.6±1.6	37.6±0.9	39.1±1.6	39.5±1.1	**43.5±1.2**
	ARI	21.0±7.6	46.9±0.0	19.9±0.0	5.7±0.4	28.5±14.0	22.2±0.4	36.1±4.1	26.1±1.1	40.5±0.3	35.6±2.5	**50.3±1.2**
	ACC	54.9±5.4	71.2±0.0	37.4±0.0	39.1±0.7	62.1±5.9	55.9±0.7	65.4±3.5	59.3±1.3	70.2±0.7	64.6±3.1	**76.0±1.0**
	F1	40.0±5.2	65.8±0.0	28.5±0.0	30.6±0.4	55.9±4.7	56.3±0.7	64.1±2.9	59.8±1.1	64.8±1.1	63.2±2.8	**73.0±1.7**
Mushroom	NMI	0.6±0.5	1.8±0.0	32.2±0.0	15.0±0.5	3.1±0.0	15.5±6.1	30.5±11.3	44.7±1.2	28.7±12.0	30.3±4.2	**45.8±0.1**
	ARI	0.8±0.7	2.5±0.0	7.8±0.0	19.7±0.5	-0.1±0.0	13.5±6.8	35.8±16.2	53.4±1.0	34.4±16.4	37.5±5.9	**54.2±0.1**
	ACC	54.1±2.1	57.9±0.0	14.2±0.0	72.2±0.3	50.5±0.0	66.6±8.0	79.0±7.3	86.5±0.3	78.3±7.7	80.5±2.4	**86.8±0.0**
	F1	54.0±2.1	57.2±0.0	14.8±0.0	72.2±0.3	37.4±0.0	62.0±12.3	78.1±8.4	86.4±0.3	77.4±8.6	80.5±2.5	**86.7±0.0**
Aminer	NMI	2.0±0.2	36.2±0.0	39.1±0.0	23.9±0.3	38.4±0.5	29.8±0.5	OOM	42.1±0.3	39.6±0.2	42.5±0.8	**49.9±0.9**
	ARI	1.2±0.2	23.9±0.0	28.4±0.0	12.6±0.2	22.8±0.9	13.6±1.3		23.5±0.6	22.6±1.1	27.7±1.2	**35.4±0.5**
	ACC	13.1±0.1	32.8±0.0	38.9±0.0	26.7±0.2	34.4±0.6	27.0±1.5		33.1±1.0	31.8±0.9	37.5±1.0	**43.1±0.6**
	F1	11.5±0.3	27.1±0.0	20.0±0.0	23.3±0.1	**29.1±0.9**	24.2±1.1		28.3±0.8	27.6±1.2	29.0±0.6	22.1±1.4
Zoo	NMI	33.7±6.2	68.3±0.0	67.4±0.0	76.8±2.3	90.7±0.7	88.9±1.7	**92.8±0.8**	82.2±1.9	90.1±0.5	80.3±3.8	90.5±1.9
	ARI	18.8±6.4	47.3±0.0	52.2±0.0	53.7±3.4	86.8±0.4	87.5±1.0	89.3±0.9	73.8±5.0	85.6±2.1	74.1±8.6	**91.8±3.2**
	ACC	45.0±4.9	61.4±0.0	78.2±0.0	62.4±1.6	89.3±0.4	89.7±0.5	91.1±0.6	80.2±3.9	88.5±1.2	80.4±4.0	**93.1±0.6**
	F1	34.9±6.6	53.9±0.0	62.3±0.0	57.6±2.7	77.9±0.9	**79.6±2.0**	79.5±0.8	69.2±5.0	77.3±0.5	65.7±5.3	78.7±1.8

Composing the learned parent assignments gives the vertex-to-cluster membership matrix:

$$\mathbf{S}^1 = \mathbf{C}^K \cdots \mathbf{C}^2, \qquad \hat{y}_i = \arg\max_k \mathbf{S}^1_{ik}. \tag{19}$$

The root assignment is fixed as $\mathbf{C}^1 = \mathbf{1}_{n_1 \times 1}$. The resulting soft hierarchy is trained by minimizing the relaxed HSE on H^{enh}: $\mathcal{L}_{\text{hse}} = \widehat{\mathcal{H}}_\Phi(H^{\text{enh}}; \{\mathbf{C}^h\}_{h=1}^K)$ using Eq. 10.

5.4 Joint Optimization

To encourage hard, mutually exclusive assignments and discourage assignment collapse, we introduce a column-mass-normalized orthogonality regularizer as follows:

$$\mathcal{L}_{\text{orth}} = (K-1)^{-1} \sum\nolimits_{h=2}^{K} \left\| (\widetilde{\mathbf{C}}^h)^\top \widetilde{\mathbf{C}}^h - \mathbf{I}_{n_{h-1}} \right\|_F^2, \tag{20}$$

where $\widetilde{\mathbf{C}}^h_{ik} = \mathbf{C}^h_{ik} / \sqrt{\sum_{u=1}^{n_h} \mathbf{C}^h_{uk}}$. Every hard assignment with nonempty parent clusters attains zero regularization loss, irrespective of cluster sizes (Appendix B.5).

To align representation learning with hypergraph partitioning, we jointly optimize all trainable modules so that the shared encoder receives both generative and HSE-based clustering supervision:

$$\mathcal{L}_{\text{HSENet}} = \mathcal{L}_{\text{hse}} + \lambda \mathcal{L}_{\text{orth}} + \mathcal{L}_{\text{rep}}, \tag{21}$$

where λ controls the regularization strength and $\mathcal{L}_{\text{rep}}$ is defined in Eq. 15.

6 Experiments

6.1 Experimental Setup

We conduct experiments on 8 public hypergraph datasets. They include 3 co-citation datasets (Cora, Citeseer, and Pubmed Sen et al. (2008)), 2 co-author datasets (DBLP Rossi & Ahmed (2015), Aminer Zhang et al. (2019)), and 3 datasets from the UCI Machine Learning Repository (20News, Mushroom, and Zoo Kelly et al. (2026)). We compare HSENet against 4 traditional attributed hypergraph clustering methods and 6 deep learning-based self-supervised or clustering-specific hypergraph learning methods. For traditional methods, we consider JNMF Du et al. (2019), AHCKA

Table 2: Ablation study. Results (%) are mean±std of 5 independent runs.

Dataset	Metric	w/o Recon.	w/o Filling	w/o Enhan.	k-means	w/o Lovász	w/o Orth.	HSENet
Cora	NMI	54.2±2.1	46.6±2.8	52.8±2.1	56.6±0.6	55.8±2.3	56.9±1.2	**57.5±0.8**
	ACC	70.8±3.5	58.0±4.9	68.3±3.8	73.4±0.6	72.3±3.3	74.2±1.7	**74.7±1.6**
Citeseer	NMI	43.8±1.5	39.9±4.6	45.1±0.7	44.7±0.6	45.3±0.6	**45.6±0.6**	**45.6±0.6**
	ACC	70.5±1.6	62.7±6.3	71.3±0.6	68.1±2.1	71.9±0.5	**72.1±0.4**	**72.1±0.4**
Pubmed	NMI	33.6±1.1	22.2±1.4	35.1±1.9	34.2±0.1	**36.3±2.6**	33.7±2.4	35.2±2.0
	ACC	71.3±4.5	54.3±1.9	73.9±0.8	71.9±0.1	72.3±4.2	71.8±3.2	**74.0±2.0**
20News	NMI	43.7±1.0	38.7±3.0	41.2±1.0	**44.5±0.6**	43.0±0.8	38.9±1.0	43.5±1.2
	ACC	**76.3±0.7**	70.0±3.3	74.0±2.2	74.3±2.1	75.8±0.4	68.5±2.0	76.0±1.0
Mushroom	NMI	32.8±14.1	39.7±6.9	45.6±0.3	42.5±3.2	45.4±0.5	**45.8±0.2**	**45.8±0.1**
	ACC	80.7±8.4	84.6±2.7	86.7±0.1	85.8±1.0	86.7±0.1	**86.8±0.1**	**86.8±0.0**
Aminer	NMI	49.0±0.9	48.5±1.5	46.3±0.6	43.4±0.2	**50.2±0.8**	48.9±0.8	49.9±0.9
	ACC	42.3±0.5	41.7±1.9	41.1±0.9	36.3±0.4	**43.8±0.4**	42.5±0.9	43.1±0.6

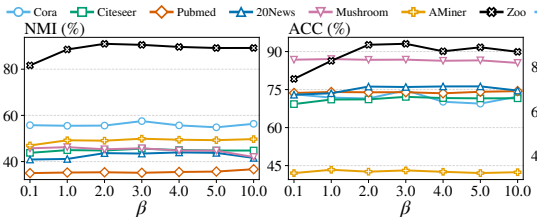


Figure 2: Sensitivity to hyperedge weight β.

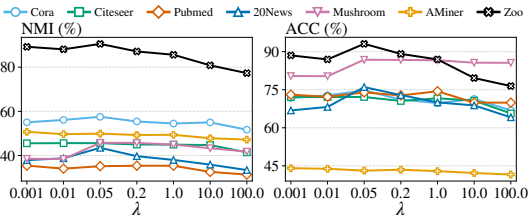


Figure 3: Sensitivity to regularizer weight λ.

Li et al. (2023), AHRC Feng et al. (2025b), and GRAC Fanseu Kamhoua et al. (2021). For deep learning methods, we consider TriCL Lee & Shin (2023), HyFi Roh et al. (2024), SS-HT Feng et al. (2025a), HypeBoy Kim et al. (2024), SE-HSSL Li et al. (2026), and CAHC Ni et al. (2026). Among them, AHCKA and AHRC are deterministic methods. We use 4 metrics to evaluate clustering performance following previous research Ni et al. (2026): Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), Accuracy (ACC), and Macro-F1 Score. By default, we set $\lambda = 0.05$, $\lambda_1 = 60$, $\lambda_2 = 120$, and $\beta = 3$. Additional implementation details are in Appendix D.3.

6.2 Experimental Results

Table 1 reports the attributed hypergraph clustering performance of all competing methods on 8 datasets. Based on these results, we draw the following key observations: **(1) Superiority of HSENet.** HSENet yields state-of-the-art results on most of the 8 datasets, securing the top spot across all 4 metrics on 5 of them. We attribute this empirical success primarily to our coupled representation and assignment learning framework, guided by a well-formulated HSE objective. Notably, HSENet demonstrates large margins over baselines on 20News, whose average hyperedge length is large at 655. This highlights the efficacy of the EDVW hypergraph enhancement, which may recover latent signals obscured within large hyperedges. Furthermore, HSENet achieves the best results on DBLP (41,302 vertices), the largest dataset evaluated, while several baselines encounter memory or time limits. **(2) Baseline Analysis.** Deep learning-based methods generally outperform traditional approaches, which can be credited to the superior representational capacity of neural networks. CAHC performs the best among the baselines, further validating the necessity of employing a clustering-oriented objective. Collectively, HSENet establishes a comprehensive advantage over competitors through its coupled learning design and HSE guidance.

6.3 Ablation Study

We evaluate 6 variants in Table 2: **w/o Recon.**, **w/o Filling**, and **w/o Enhan.** remove masked feature reconstruction, hyperedge filling, and EDVW topology enhancement, respectively. The remaining variants replace the HSE-guided assigner with k-means (**k-means**), omit the Lovász extension (**w/o Lovász**), or remove orthogonality regularization (**w/o Orth.**). Removing either reconstruction task generally reduces clustering performance, supporting the contributions of both semantic and structural supervision. Disabling EDVW enhancement also leads to substantial degradation on several

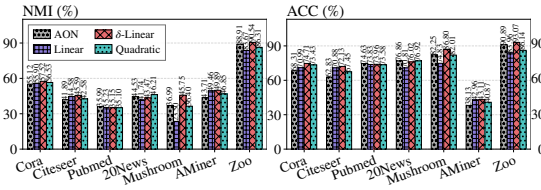


Figure 4: Comparison of splitting functions Φ.

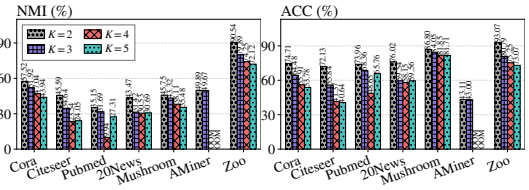


Figure 5: Comparison of tree heights K.

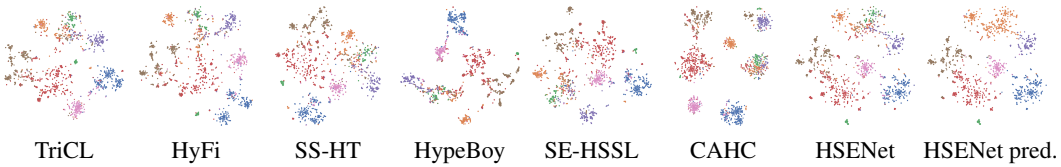


Figure 6: Cora t-SNE: panels 1–7 use ground-truth labels; panel 8 uses predicted clusters.

datasets, while the benefits of HSE-guided assignment, the Lovász extension, and orthogonality regularization vary across datasets. Overall, HSENet achieves the best or competitive performance across all evaluated datasets, supporting the combined design of representation learning, topology enhancement, and assignment optimization.

6.4 Sensitivity Analysis and Visualization

Coefficient sensitivity. We first examine the sensitivity to the representation-hyperedge weight β and the orthogonality regularization weight λ in Figures 2 and 3. For β, HSENet achieves stable performance over 2.0 to 4.0, while excessively small values slightly impede the clustering performance because the representation hypergraph is not emphasized enough. In contrast, overly large values degrade ACC, possibly because they overemphasize representation-derived hyperedges and weaken the original hypergraph's role as a structural anchor, thereby destabilizing model training. Regarding λ, moderate regularization consistently improves ACC scores, whereas a very large λ impedes clustering performance by imposing an overly restrictive assignment structure. Overall, HSENet remains insensitive to both coefficients within reasonably broad ranges.

Structural choices. We further investigate the hyperedge splitting function Φ and the partitioning-tree height K in Figures 4 and 5. Among the four splitting functions, δ-Linear splitting yields the strongest results across most datasets, suggesting that it is more suitable for HSE for hypergraph clustering. For the tree height, $K = 2$ consistently performs best or nearly best. Increasing K beyond 2 largely impedes clustering performance, suggesting that a shallow hierarchy is sufficient for these datasets. We therefore use $K = 2$ in all experiments unless otherwise specified.

Visualization. We visualize the learned vertex embeddings on Cora using t-SNE. Figure 6 shows 6 deep baselines and HSENet colored by ground-truth labels, together with HSENet embeddings colored by predicted assignment. Compared with other baselines, clustering-specific methods CAHC and HSENet form more compact and better-separated clusters. This indicates clustering-oriented designs help models learn separable embeddings. Moreover, HSENet produces fewer mixed clusters without a dominant class. Comparing the last 2 columns, HSENet's predicted partition closely matches the ground-truth labels, demonstrating the effectiveness of the HSE-guided assigner.

7 Conclusion

We present HSENet, a unified generative framework for attributed hypergraph clustering guided by hypergraph structural entropy (HSE). We formulate HSE for EDVW hypergraphs and derive a continuous Lovász relaxation for gradient-based partition learning. HSENet jointly optimizes generative representation learning, topology enhancement, and HSE-guided hierarchical assignment, allowing hypergraph structure to guide both representations and cluster assignments. Experiments on 8 benchmark datasets show that HSENet outperforms 10 baselines on most evaluated metrics.

REFERENCES

Yuwei Cao, Hao Peng, Angsheng Li, Chenyu You, Zhifeng Hao, and Philip S. Yu. Multi-relational structural entropy. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, volume 244, pp. 532–546, 2024.

Uthsav Chitra and Benjamin Raphael. Random walks on hypergraphs with edge-dependent vertex weights. In *International conference on machine learning*, pp. 1172–1181. PMLR, 2019.

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

Rundong Du, Barry Drake, and Haesun Park. Hybrid clustering based on content and connection structure using joint nonnegative matrix factorization. *Journal of Global Optimization*, 74(4): 861–877, 2019.

Barakeel Fanseu Kamhoua, Lin Zhang, Kaili Ma, James Cheng, Bo Li, and Bo Han. Hypergraph convolution based attributed hypergraph clustering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 453–463, 2021.

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

Yifan Feng, Shiquan Liu, Shihui Ying, Shaoyi Du, Zongze Wu, and Yue Gao. Self-supervised hypergraph training framework via structure-aware learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(11):10160–10176, 2025a.

Zijin Feng, Miao Qiao, Chengzhi Piao, and Hong Cheng. On graph representation for attributed hypergraph clustering. *Proceedings of the ACM on Management of Data*, 3(1):1–26, 2025b.

Haiyan Gong, Sichen Zhang, Xiaotong Zhang, and Yang Chen. A method for chromatin domain partitioning based on hypergraph clustering. *Computational and structural biotechnology journal*, 23:1584–1593, 2024.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 594–604, 2022.

Jing Huang and Jie Yang. Unignn: a unified framework for graph and hypergraph neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 2021.

Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository, 2026. URL `https://archive.ics.uci.edu/datasets`.

Sunwoo Kim, Shinhwan Kang, Fanchen Bu, Soo Yong Lee, Jaemin Yoo, and Kijung Shin. Hypeboy: Generative self-supervised representation learning on hypergraphs. In *The Twelfth International Conference on Learning Representations*, pp. 1–32, 2024.

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

Angsheng Li. Structural information learning machinery: Learning from observing, associating, optimizing, decoding, and abstracting. *arXiv preprint arXiv:2001.09637*, 2020.

Angsheng Li and Yicheng Pan. Structural information and dynamical complexity of networks. *IEEE Transactions on Information Theory*, 62(6):3290–3339, 2016.

Benzheng Li, Shunyang Bi, Hailong You, Zhongdong Qi, Guangxin Guo, Richard Sun, and Yuming Zhang. Mapart: An efficient multi-fpga system-aware hypergraph partitioning framework. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(10):3212–3225, 2024.

Fan Li, Xiaoyang Wang, Dawei Cheng, Ying Zhang, Wenjie Zhang, and Xuemin Lin. Fairness-aware hypergraph self-supervised learning with sampling-efficient signals. *IEEE Transactions on Knowledge and Data Engineering*, 38(6):3611–3625, 2026.

Pan Li and Olgica Milenkovic. Submodular hypergraphs: p-laplacians, cheeger inequalities and spectral clustering. In *International Conference on Machine Learning*, pp. 3014–3023. PMLR, 2018.

Yiran Li, Renchi Yang, and Jieming Shi. Efficient and effective attributed hypergraph clustering via k-nearest neighbor augmentation. *Proceedings of the ACM on Management of Data*, 1(2):1–23, 2023.

László Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art: Bonn 1982*, pp. 235–257. Springer, 1983.

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

Daeyoung Roh, Donghee Han, Daehee Kim, Keejun Han, and Mun Yi. Closer through commonality: Enhancing hypergraph contrastive learning with shared groups. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 640–649. IEEE, 2024.

Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 4292–4293, 2015.

Nicolò Ruggeri, Martina Contisciani, Federico Battiston, and Caterina De Bacco. Community detection in large hypergraphs. *Science Advances*, 9(28):eadg9159, 2023.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–106, 2008.

Dingli Su, Hao Peng, Yicheng Pan, and Angsheng Li. A survey of structural entropy: Theory, methods, and applications. In *IJCAI*, pp. 10660–10668, 2025.

Li Sun, Zhenhao Huang, Hao Peng, Yujie Wang, Chunyang Liu, and Philip S. Yu. LSEnet: Lorentz structural entropy neural network for deep graph clustering. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 47078–47104, 2024.

Li Sun, Zhenhao Huang, Yujie Wang, Hongbo Lv, Chunyang Liu, Hao Peng, and Philip S Yu. Asil: Augmented structural information learning for deep graph clustering in hyperbolic space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Yuuki Takai, Atsushi Miyauchi, Masahiro Ikeda, and Yuichi Yoshida. Hypergraph clustering based on pagerank. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1970–1978, 2020.

Nate Veldt, Austin R Benson, and Jon Kleinberg. Minimizing localized ratio cut objectives in hypergraphs. In *Proceedings of KDD*, pp. 1708–1718, 2020.

Nate Veldt, Austin R. Benson, and Jon Kleinberg. Augmented sparsifiers for generalized hypergraph cuts. *Journal of Machine Learning Research*, 24(207):1–50, 2023.

Qianqian Wang, Bowen Zhao, Zhengming Ding, Wei Feng, and Quanxue Gao. Hypergraph clustering network with partial attribute imputation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2697–2706, 2025a.

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

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

Joyce Jiyoung Whang, Rundong Du, Sangwon Jung, Geon Lee, Barry Drake, Qingqing Liu, Seonggoo Kang, and Haesun Park. Mega: Multi-view semi-supervised clustering of hypergraphs. *Proceedings of the VLDB Endowment*, 13(5):698–711, 2020.

Runze Yang, Hao Peng, Chunyang Liu, and Angsheng Li. Incremental measurement of structural entropy for dynamic graphs. *Artificial Intelligence*, 334:104175, 2024.

Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 793–803, 2019.

Jingyun Zhang, Hao Peng, Li Sun, Guanlin Wu, Chunyang Liu, and Zhengtao Yu. Unsupervised graph clustering with deep structural entropy. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 3752–3763, 2025a.

Jingyun Zhang, Hao Peng, Mingdai Yang, and Philip S Yu. Enhanced pre-training for recommendation via hypergraph structural entropy. *ACM Transactions on Information Systems*, 44(2):1–48, 2025b.

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. *arXiv preprint arXiv:2607.05469*, 2026.

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

A AI USE STATEMENT

In this work, we used generative AI tools to assist with grammar checking, wording refinement, readability improvements, and code development. We did not use generative AI tools to locate or verify references; all references were independently identified and verified by the authors. All AI-assisted revisions were carefully reviewed by the authors, and all AI-generated code was subjected to detailed manual review and validation for correctness before use. We take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

B THEORETICAL RESULTS

B.1 REDUCTION RELATIONS OF HSE

Theorem 4.1 (Representation Equivalence and Reduction Relations). Let $\mathcal{T}$ be a discrete partitioning tree with positive non-root module volumes, encoded by binary row-stochastic assignments $\{\mathbf{C}^h\}_{h=1}^K$. (i) Eq. 5 is exactly the level-wise scalar representation of the set-based structural entropy in Eq. 4 with boundary cost g_Φ. (ii) If $\gamma_e(v) = 1$ for every incidence, EDVW-HSE reduces to cardinality-based hypergraph structural entropy. (iii) If, in addition, the hypergraph is 2-uniform, then $\mathcal{H}_\Phi^{\mathcal{T}}(H) = c_\Phi \mathcal{H}^{\mathcal{T}}(G)$, where G is the induced weighted graph and $c_\Phi = \Phi(1, 2)$.

Proof. The proof draws on the structural-entropy framework of Li & Pan (2016) and the level-wise formulation of Sun et al. (2026), adapted to EDVW hypergraphs. **(i) Equivalence of the set-based and level-wise scalar forms.** Recall that $\mathcal{T}_K$ consists of the singleton vertex leaves ordered as $v_1, \ldots, v_n$, and $\mathbf{C}^{K+1} = \mathbf{I}_n$. Hence $\mathbf{S}^K = \mathbf{I}_n$ and

$$\mathbf{S}_{i\alpha}^K = \mathbb{I}(v_i \in T_\alpha), \qquad \alpha \in \mathcal{T}_K. \tag{22}$$

For $h = K-1, \ldots, 0$, the matrix products satisfy the recursion

$$\mathbf{S}^h = \mathbf{S}^{h+1}\mathbf{C}^{h+1}. \tag{23}$$

Suppose that $\mathbf{S}_{i\beta}^{h+1} = \mathbb{I}(v_i \in T_\beta)$ for every $\beta \in \mathcal{T}_{h+1}$. Because $\mathbf{C}^{h+1}$ is binary and row-stochastic, each β has exactly one parent in $\mathcal{T}_h$. Therefore,

$$\begin{aligned} \mathbf{S}_{i\alpha}^h &= \sum_{\beta \in \mathcal{T}_{h+1}} \mathbf{S}_{i\beta}^{h+1}\mathbf{C}_{\beta\alpha}^{h+1} \\ &= \mathbb{I}(v_i \in T_\alpha). \end{aligned} \tag{24}$$

Backward induction from $h = K$ thus shows that $\mathbf{S}^h$ is precisely the vertex-to-module indicator matrix at every level, including $\mathbf{S}_{i\rho}^0 = 1$ at the root.

For $\alpha \in \mathcal{T}_h$, the numerator of the volume ratio in Eq. 5 is therefore

$$\sum_{i=1}^n d_i \mathbf{S}_{i\alpha}^h = \sum_{v_i \in T_\alpha} d_i = \mathrm{vol}(T_\alpha). \tag{25}$$

Since $\mathbf{C}_{\alpha\beta}^h = 1$ precisely when $\beta = \alpha^-$, its denominator is

$$\sum_{\beta \in \mathcal{T}_{h-1}} \sum_{i=1}^n d_i \mathbf{C}_{\alpha\beta}^h \mathbf{S}_{i\beta}^{h-1} = \sum_{i=1}^n d_i \mathbf{S}_{i\alpha^-}^{h-1} = \mathrm{vol}(T_{\alpha^-}). \tag{26}$$

The argument of the splitting function is the EDVW mass of hyperedge e_j contained in T_α:

$$\sum_{i=1}^n \mathbf{S}_{i\alpha}^h \mathbf{H}_{ij} = \sum_{v_i \in T_\alpha} \mathbf{H}_{ij} = x_j(T_\alpha). \tag{27}$$

Summing the splitting penalties over hyperedges therefore gives the cut value:

$$\sum_{j=1}^m \omega(e_j)\Phi\left(\sum_{i=1}^n \mathbf{S}_{i\alpha}^h \mathbf{H}_{ij}, \Gamma_j\right) = \sum_{j=1}^m \omega(e_j)\Phi(x_j(T_\alpha), \Gamma_j) = g_\Phi(T_\alpha). \tag{28}$$

The logarithmic factor in Eq. 5 is independent of j. Substituting these identities and collecting the sum over hyperedges yield

$$\mathcal{H}_{\Phi}^{\mathcal{T}}(H) = -\frac{1}{\mathrm{vol}(H)} \sum_{h=1}^{K} \sum_{\alpha \in \mathcal{T}_h} g_{\Phi}(T_\alpha) \log_2 \frac{\mathrm{vol}(T_\alpha)}{\mathrm{vol}(T_{\alpha^-})}$$
$$= -\frac{1}{\mathrm{vol}(H)} \sum_{\alpha \in \mathcal{T} \setminus \{\rho\}} g_{\Phi}(T_\alpha) \log_2 \frac{\mathrm{vol}(T_\alpha)}{\mathrm{vol}(T_{\alpha^-})}. \tag{29}$$

The second equality holds because the level sets $\mathcal{T}_1, \ldots, \mathcal{T}_K$ form a disjoint enumeration of all non-root nodes. This is exactly Eq. 4; the assumed positive module volumes ensure that all logarithms are well-defined. Padding a shallower leaf with a single-child chain preserves its singleton module, so each added term has volume ratio 1 and contributes zero. Thus the equality also applies to trees of depth at most K represented after padding. This proves (i).

(ii) Reduction to cardinality-based hypergraph structural entropy. If $\gamma_e(v) = 1$ for every incidence, then $\mathbf{H}$ becomes the ordinary binary incidence matrix, $\Gamma_j = |e_j|$, and

$$x_j(S) = \sum_{v_i \in S} \mathbb{I}(v_i \in e_j) = |e_j \cap S|. \tag{30}$$

The EDVW degree and generalized cut consequently become

$$d_i = \sum_{j : v_i \in e_j} \omega(e_j), \qquad g_{\Phi}(S) = \sum_{j=1}^{m} \omega(e_j) \Phi(|e_j \cap S|, |e_j|), \tag{31}$$

which are precisely the ordinary weighted hypergraph degree and a cardinality-based splitting cut. Hence $\mathrm{vol}(S) = \sum_{v_i \in S} d_i$ is the ordinary hypergraph volume, and substituting these quantities into Eq. 5, using the scalar identities proved in (i), gives cardinality-based hypergraph structural entropy. This proves (ii).

(iii) Reduction to ordinary graph structural entropy. In addition to the binary-incidence assumption in (ii), suppose that every hyperedge contains exactly two vertices. Identify each hyperedge $e_j = \{u_j, v_j\}$ with an edge of the weighted graph $G = (V, E, \omega)$. For any $S \subset V$,

$$x_j(S) = |e_j \cap S| = \begin{cases} 0, & u_j, v_j \notin S, \\ 1, & |\{u_j, v_j\} \cap S| = 1, \\ 2, & u_j, v_j \in S. \end{cases} \tag{32}$$

The endpoint conditions $\Phi(0,2) = \Phi(2,2) = 0$ imply that an edge whose endpoints lie on the same side of the cut contributes zero. Every crossing edge has $x_j(S) = 1$ and therefore contributes $\omega(e_j) c_{\Phi}$, where $c_{\Phi} = \Phi(1,2)$. Thus

$$g_{\Phi}(S) = c_{\Phi} \sum_{\substack{e_j \in E: \\ |e_j \cap S| = 1}} \omega(e_j) = c_{\Phi} g_G(S). \tag{33}$$

Moreover, for every vertex v_i,

$$d_i = \sum_{j : v_i \in e_j} \omega(e_j) = d_i^G. \tag{34}$$

Therefore $\mathrm{vol}_H(S) = \mathrm{vol}_G(S)$ for every $S \subseteq V$, and in particular $\mathrm{vol}(H) = \mathrm{vol}(G)$. Applying (i) to Eq. 5 and substituting these cut and volume identities gives

$$\mathcal{H}_{\Phi}^{\mathcal{T}}(H) = -\frac{c_{\Phi}}{\mathrm{vol}(G)} \sum_{\alpha \in \mathcal{T} \setminus \{\rho\}} g_G(T_\alpha) \log_2 \frac{\mathrm{vol}_G(T_\alpha)}{\mathrm{vol}_G(T_{\alpha^-})}$$
$$= c_{\Phi} \mathcal{H}^{\mathcal{T}}(G). \tag{35}$$

This proves (iii). Finally,

$$\Phi_{\mathrm{AON}}(1,2) = \Phi_{\mathrm{Lin}}(1,2) = \Phi_{\delta\text{-Lin}}(1,2) = 1 \quad (\delta \geq 1), \qquad \Phi_{\mathrm{Quad}}(1,2) = \frac{1}{2}. \tag{36}$$

Hence the all-or-nothing, linear, and δ-linear penalties with $\delta \geq 1$ recover ordinary graph structural entropy exactly, whereas the unscaled quadratic penalty yields one half of it. □

B.2 Connection to Hypergraph Conductance

Theorem 4.2 (Connection to Hypergraph Conductance). Let the degree entropy be $\mathcal{H}_{\deg}(H) = -\sum_{i=1}^{n} \frac{d_i}{\mathrm{vol}(H)} \log_2 \frac{d_i}{\mathrm{vol}(H)}$. The discrete HSE in Eq. 5 satisfies the following bounds. (i) For any discrete partitioning tree $\mathcal{T}$ whose non-root modules and their complements have positive volume,

$$\mathcal{H}_{\Phi}^{\mathcal{T}}(H) \geq \mathrm{Cond}_{\Phi}(H)\big(\mathcal{H}_{\deg}(H) - 1\big). \tag{37}$$

(ii) If every non-root module additionally satisfies $\mathrm{vol}(T_\alpha) \leq \mathrm{vol}(H)/2$, then

$$\mathcal{H}_{\Phi}^{\mathcal{T}}(H) \geq \mathrm{Cond}_{\Phi}(H)\mathcal{H}_{\deg}(H), \qquad \text{equivalently} \qquad \frac{\mathcal{H}_{\Phi}^{\mathcal{T}}(H)}{\mathcal{H}_{\deg}(H)} \geq \mathrm{Cond}_{\Phi}(H). \tag{38}$$

Proof of Theorem 4.2. The proof draws on the structural-entropy and conductance analyses in Li & Pan (2016); Li (2020); Sun et al. (2026), adapted to EDVW cuts and volumes. By Theorem 4.1(i), the scalar sums in Eq. 5 give the module cuts, module volumes, and parent volumes appearing in Eq. 4. We use this equivalent set-based form below.

A tree identity for degree entropy. Write

$$V_H = \mathrm{vol}(H), \qquad V_\alpha = \mathrm{vol}(T_\alpha), \qquad \ell_\alpha = \log_2 \frac{V_{\alpha^-}}{V_\alpha} \tag{39}$$

where $V_\rho = V_H$ and ℓ_α is defined for every non-root node α. Since $T_\alpha \subseteq T_{\alpha^-}$ and all relevant volumes are positive, $V_\alpha \leq V_{\alpha^-}$ and hence $\ell_\alpha \geq 0$. Moreover, $V_\alpha = \sum_{v_i \in T_\alpha} d_i$. Therefore, changing the order of summation yields

$$\begin{aligned} \frac{1}{V_H} \sum_{\alpha \neq \rho} V_\alpha \ell_\alpha &= \frac{1}{V_H} \sum_{\alpha \neq \rho} \sum_{v_i \in T_\alpha} d_i \ell_\alpha \\ &= \frac{1}{V_H} \sum_{i=1}^{n} d_i \sum_{\alpha \in \mathrm{path}(v_i, \rho) \setminus \{\rho\}} \ell_\alpha. \end{aligned} \tag{40}$$

The modules containing v_i are exactly the nodes on its unique path from the leaf $\{v_i\}$ to the root. The logarithms telescope along this path, and the leaf volume is $\mathrm{vol}(\{v_i\}) = d_i$. Thus

$$\frac{1}{V_H} \sum_{\alpha \neq \rho} V_\alpha \ell_\alpha = \frac{1}{V_H} \sum_{i=1}^{n} d_i \log_2 \frac{V_H}{d_i} = \mathcal{H}_{\deg}(H). \tag{41}$$

(i) General lower bound. Partition the non-root nodes according to whether their modules contain more than half of the total volume:

$$\mathcal{A} = \{\alpha \neq \rho : V_\alpha > V_H/2\}, \qquad \mathcal{B} = \{\alpha \neq \rho : V_\alpha \leq V_H/2\}. \tag{42}$$

By the definitions of local and global EDVW conductance, every non-root module satisfies

$$\begin{aligned} g_\Phi(T_\alpha) &= \mathrm{cond}_\Phi(T_\alpha) \min\{V_\alpha, V_H - V_\alpha\} \\ &\geq \mathrm{Cond}_\Phi(H) \min\{V_\alpha, V_H - V_\alpha\}. \end{aligned} \tag{43}$$

The assumption that both T_α and its complement have positive volume ensures that this local conductance is well-defined. For $\alpha \in \mathcal{B}$, the minimum equals V_α; for $\alpha \in \mathcal{A}$, it equals $V_H - V_\alpha$.

Using the equivalent set-based form of Eq. 5, the nonnegativity of ℓ_α, and Eq. 41, we obtain

$$\begin{aligned} \mathcal{H}_{\Phi}^{\mathcal{T}}(H) &= \frac{1}{V_H} \sum_{\alpha \neq \rho} g_\Phi(T_\alpha) \ell_\alpha \\ &\geq \frac{\mathrm{Cond}_\Phi(H)}{V_H} \left[\sum_{\alpha \in \mathcal{B}} V_\alpha \ell_\alpha + \sum_{\alpha \in \mathcal{A}} (V_H - V_\alpha) \ell_\alpha \right] \\ &= \mathrm{Cond}_\Phi(H) \left(\mathcal{H}_{\deg}(H) - \Delta_{\mathcal{T}} \right), \end{aligned} \tag{44}$$

where

$$\Delta_{\mathcal{T}} = \sum_{\alpha \in \mathcal{A}} \left(\frac{2V_\alpha}{V_H} - 1 \right) \ell_\alpha. \tag{45}$$

The last equality follows by first replacing the two sums in brackets with $\sum_{\alpha \neq \rho} V_\alpha \ell_\alpha$ and then subtracting $(2V_\alpha - V_H)\ell_\alpha$ for each $\alpha \in \mathcal{A}$.

It remains to bound $\Delta_{\mathcal{T}}$. If $\mathcal{A}$ is empty, then $\Delta_{\mathcal{T}} = 0$. Otherwise, any two nodes in $\mathcal{A}$ must be comparable in the tree: if they were incomparable, their modules would be disjoint, while each would have volume greater than $V_H/2$, contradicting additivity of volume. Furthermore, every non-root ancestor of a node in $\mathcal{A}$ also belongs to $\mathcal{A}$. Hence all nodes in $\mathcal{A}$ form one consecutive root-to-leaf chain. Enumerate this chain as

$$\rho = \alpha_0, \alpha_1, \ldots, \alpha_r, \qquad \alpha_i^- = \alpha_{i-1}, \qquad V_i = \mathrm{vol}(T_{\alpha_i}). \tag{46}$$

For every $i \geq 1$, complement positivity and membership in $\mathcal{A}$ give

$$\frac{V_H}{2} < V_i < V_H, \qquad 0 < \frac{2V_i}{V_H} - 1 < 1. \tag{47}$$

Consequently, using the nonnegativity of the logarithms and then telescoping,

$$\begin{aligned} 0 \leq \Delta_{\mathcal{T}} &= \sum_{i=1}^{r} \left(\frac{2V_i}{V_H} - 1 \right) \log_2 \frac{V_{i-1}}{V_i} \\ &< \sum_{i=1}^{r} \log_2 \frac{V_{i-1}}{V_i} = \log_2 \frac{V_H}{V_r} < 1. \end{aligned} \tag{48}$$

The strict upper bound also holds when some parent and child volumes coincide, because $V_r < V_H$ guarantees that at least one logarithm on the chain is positive. Combining $\Delta_{\mathcal{T}} < 1$ with Eq. 44 and $\mathrm{Cond}_\Phi(H) \geq 0$ gives

$$\mathcal{H}_\Phi^{\mathcal{T}}(H) \geq \mathrm{Cond}_\Phi(H)\big(\mathcal{H}_{\deg}(H) - 1\big), \tag{49}$$

which proves part (i).

(ii) Sharper bound under the half-volume condition. Suppose now that $V_\alpha \leq V_H/2$ for every non-root node. Then $\mathcal{A}$ is empty and $\Delta_{\mathcal{T}} = 0$. Equation 44 therefore reduces to

$$\mathcal{H}_\Phi^{\mathcal{T}}(H) \geq \mathrm{Cond}_\Phi(H)\mathcal{H}_{\deg}(H). \tag{50}$$

Finally, every leaf is a positive-volume singleton and its complement has positive volume. Thus the normalized degree distribution has at least two positive entries, so $\mathcal{H}_{\deg}(H) > 0$. Dividing by $\mathcal{H}_{\deg}(H)$ yields

$$\frac{\mathcal{H}_\Phi^{\mathcal{T}}(H)}{\mathcal{H}_{\deg}(H)} \geq \mathrm{Cond}_\Phi(H), \tag{51}$$

which proves part (ii). □

B.3 Bound of Relaxed HSE

Theorem 4.3 (Bound of Relaxed HSE). Fix H, Φ, and the level sizes. Let $\{\overline{\mathbf{C}}^h\}$ encode a discrete tree $\mathcal{T}$ with positive non-root module volumes. There exist $r_{\mathcal{T}}, L_{\mathcal{T}} > 0$ such that any row-stochastic assignments satisfying $\sum_{h=1}^{K} \|\mathbf{C}^h - \overline{\mathbf{C}}^h\|_F \leq \epsilon \leq r_{\mathcal{T}}$ obey:

$$\left| \widehat{\mathcal{H}}_\Phi(H; \{\mathbf{C}^h\}) - \mathcal{H}_\Phi^{\mathcal{T}}(H) \right| \leq L_{\mathcal{T}} \epsilon = O(\epsilon). \tag{52}$$

Conventions and additional details. As in the main text, H is an EDVW hypergraph with $\Gamma_j > 0$ for every hyperedge, and Φ satisfies the conditions in the preliminaries. The fixed level sizes are $(n_0, \ldots, n_K)$ with $n_0 = 1$ and $n_K = n$. For each $h = 1, \ldots, K$, the discrete assignments $\overline{\mathbf{C}}^h \in \{0,1\}^{n_h \times n_{h-1}}$ and the relaxed assignments $\mathbf{C}^h \in [0,1]^{n_h \times n_{h-1}}$ are nonnegative and row-stochastic; $\mathbf{C}^{K+1} = \overline{\mathbf{C}}^{K+1} = \mathbf{I}_n$ remains fixed. The radius $r_{\mathcal{T}}$ can be chosen so that all module and parent reference volumes in Eq. 10 are positive throughout the stated neighborhood. The constants

$r_{\mathcal{T}}$ and $L_{\mathcal{T}}$ may depend on H, Φ, the level sizes, and $\mathcal{T}$, but are independent of ϵ and the admissible assignments. In particular, setting $\epsilon = 0$ gives exact agreement between the relaxed and discrete HSE at $\mathbf{C}^h = \overline{\mathbf{C}}^h$ for all h.

We first define the quantities used in the proof and establish this bound, then derive a conditional optimal-tree bound.

B.3.1 Proof of the Bound of Relaxed HSE

Proof of Theorem 4.3. The proof builds on the structural-entropy framework of Li & Pan (2016) and follows the approximation perspective of Sun et al. (2026). Write $\mathbf{C} = \{\mathbf{C}^h\}_{h=1}^K$ and $\overline{\mathbf{C}} = \{\overline{\mathbf{C}}^h\}_{h=1}^K$, and define

$$\Delta(\mathbf{C}, \overline{\mathbf{C}}) = \sum_{h=1}^{K} \|\mathbf{C}^h - \overline{\mathbf{C}}^h\|_F. \tag{53}$$

For each module α at level h, abbreviate its soft volume, parent reference volume, relaxed boundary cost, and logarithmic volume ratio as

$$\begin{aligned} v_\alpha^h(\mathbf{C}) &= \sum_i d_i \mathbf{S}_{i\alpha}^h, & p_\alpha^h(\mathbf{C}) &= \sum_\beta \mathbf{C}_{\alpha\beta}^h v_\beta^{h-1}(\mathbf{C}), \\ g_\alpha^h(\mathbf{C}) &= \sum_j \omega(e_j) \widehat{F}_{j,\Phi}(\mathbf{s}_{j,\alpha}^h), & \ell_\alpha^h(\mathbf{C}) &= \log_2 \frac{v_\alpha^h(\mathbf{C})}{p_\alpha^h(\mathbf{C})}. \end{aligned} \tag{54}$$

Thus Eq. 10 equals $-\operatorname{vol}(H)^{-1} \sum_{h,\alpha} g_\alpha^h(\mathbf{C}) \ell_\alpha^h(\mathbf{C})$. A bar on any of these quantities denotes evaluation at $\overline{\mathbf{C}}$.

Exact agreement at the discrete tree. At $\overline{\mathbf{C}}$, the vector $\bar{\mathbf{s}}_{j,\alpha}^h$ is the indicator of $e_j \cap T_\alpha$. The extension property gives

$$\widehat{F}_{j,\Phi}(\bar{\mathbf{s}}_{j,\alpha}^h) = F_{j,\Phi}(e_j \cap T_\alpha). \tag{55}$$

Moreover, $\overline{v}_\alpha^h = \operatorname{vol}(T_\alpha)$ and $\overline{p}_\alpha^h = \operatorname{vol}(T_{\alpha^-})$, because each row of $\overline{\mathbf{C}}^h$ selects its unique parent. Consequently,

$$\widehat{\mathcal{H}}_\Phi(H; \overline{\mathbf{C}}) = \mathcal{H}_\Phi^{\mathcal{T}}(H). \tag{56}$$

Local stability of the scalar terms. The entries of $\mathbf{S}^h$ are polynomials in the entries of $\mathbf{C}$. On the compact product of row-stochastic simplices, these maps and the volume maps in Eq. 54 are Lipschitz. For fixed H, each $\widehat{F}_{j,\Phi}$ is continuous and piecewise linear with finitely many ordering regions, so it is Lipschitz on $[0,1]^{|e_j|}$, including at sorting ties. Composition and finite summation therefore give constants $a_{h\alpha} < \infty$ such that

$$|v_\alpha^h - \overline{v}_\alpha^h| + |p_\alpha^h - \overline{p}_\alpha^h| + |g_\alpha^h - \overline{g}_\alpha^h| \le a_{h\alpha} \Delta(\mathbf{C}, \overline{\mathbf{C}}). \tag{57}$$

All discrete node and parent volumes are positive. By continuity, there exist $r_{\mathcal{T}} > 0$ and $\nu_{\min} > 0$ such that every v_α^h and p_α^h is at least $\nu_{\min}$ whenever $\Delta(\mathbf{C}, \overline{\mathbf{C}}) \le r_{\mathcal{T}}$. In particular, the relaxed objective is well-defined throughout this neighborhood. Applying the mean value theorem separately to $\log_2 v_\alpha^h$ and $\log_2 p_\alpha^h$ yields

$$|\ell_\alpha^h - \overline{\ell}_\alpha^h| \le \frac{|v_\alpha^h - \overline{v}_\alpha^h| + |p_\alpha^h - \overline{p}_\alpha^h|}{\nu_{\min} \ln 2} \le \frac{a_{h\alpha}}{\nu_{\min} \ln 2} \Delta(\mathbf{C}, \overline{\mathbf{C}}). \tag{58}$$

The closed neighborhood is compact, so there are finite bounds $|\ell_\alpha^h| \le M_{h\alpha}$ on it.

Objective bound. Combining these estimates with Eq. 56 gives

$$\begin{aligned} &\left| \widehat{\mathcal{H}}_\Phi(H; \mathbf{C}) - \mathcal{H}_\Phi^{\mathcal{T}}(H) \right| \\ &\quad \le \frac{1}{\operatorname{vol}(H)} \sum_{h,\alpha} \left(|g_\alpha^h - \overline{g}_\alpha^h| \, |\ell_\alpha^h| + |\overline{g}_\alpha^h| \, |\ell_\alpha^h - \overline{\ell}_\alpha^h| \right) \\ &\quad \le L_{\mathcal{T}} \Delta(\mathbf{C}, \overline{\mathbf{C}}), \qquad L_{\mathcal{T}} = \frac{1}{\operatorname{vol}(H)} \sum_{h,\alpha} a_{h\alpha} \left(M_{h\alpha} + \frac{|\overline{g}_\alpha^h|}{\nu_{\min} \ln 2} \right). \end{aligned} \tag{59}$$

Increasing $L_{\mathcal{T}}$ if necessary makes it strictly positive. Taking $\Delta(\mathbf{C}, \overline{\mathbf{C}}) \le \epsilon \le r_{\mathcal{T}}$ proves the claim. The constants may depend on H, Φ, the level sizes, and the reference tree, but not on ϵ. □

B.3.2 A Conditional Bound Relative to the Optimal Discrete Tree

To specify the comparison class, let $\mathfrak{T}$ be the nonempty, finite set of labeled discrete trees with the fixed level sizes and positive non-root module volumes. Define

$$\mathcal{H}^{\star}_{\Phi,\mathfrak{T}}(H) = \min_{\mathcal{T}\in\mathfrak{T}} \mathcal{H}^{\mathcal{T}}_{\Phi}(H). \tag{60}$$

The quantity $\mathcal{H}^{\star}_{\Phi,\mathfrak{T}}(H)$ is the minimum discrete HSE among trees with the prescribed number of nodes at each level. In contrast, $\mathcal{H}^{K}_{\Phi}(H)$ allows all valid trees of depth at most K, without fixing the level sizes. Hence, $\mathcal{H}^{\star}_{\Phi,\mathfrak{T}}(H) \ge \mathcal{H}^{K}_{\Phi}(H)$, with equality if a globally optimal tree can be represented with the prescribed level sizes. Here, shallower leaves may be padded with single-child chains without changing the HSE.

To compare soft assignments with discrete trees, define $R(\mathbf{C})$ by replacing each row of each $\mathbf{C}^h$ with a one-hot vector at a maximum entry, using a fixed rule to break ties. This selects one parent for each non-root node. However, rounding may leave some modules with zero volume. When all non-root modules have positive volume, the rounded assignments encode a tree in $\mathfrak{T}$, which we denote by $\mathcal{T}_{R(\mathbf{C})}$.

Lemma 1 (Uniform consistency near valid discrete trees)**.** *There exist $r, L > 0$, depending only on H, Φ, and the fixed level sizes, such that for every $\mathcal{T} \in \mathfrak{T}$ with encoding $\overline{\mathbf{C}}$, and every row-stochastic $\mathbf{C}$ with $\Delta(\mathbf{C}, \overline{\mathbf{C}}) \le \epsilon \le r$,*

$$\left|\widehat{\mathcal{H}}_{\Phi}(H; \mathbf{C}) - \mathcal{H}^{\mathcal{T}}_{\Phi}(H)\right| \le L\epsilon. \tag{61}$$

Proof. Apply Theorem 4.3 to each of the finitely many labeled trees in $\mathfrak{T}$ and take $r = \min\{1/4, \min_{\mathcal{T}\in\mathfrak{T}} r_{\mathcal{T}}\} > 0$ and $L = \max_{\mathcal{T}\in\mathfrak{T}} L_{\mathcal{T}} < \infty$. These choices are independent of ϵ and give the stated uniform bound. □

Proposition 1 (Conditional optimal-tree bound)**.** *Fix $0 < \epsilon \le r$ and a nonempty neural parameter family Θ_ϵ producing assignments $\mathbf{C}_\theta$. Assume:*

1. *Every $\theta \in \Theta_\epsilon$ satisfies $\mathcal{T}_\theta := \mathcal{T}_{R(\mathbf{C}_\theta)} \in \mathfrak{T}$ and $\Delta(\mathbf{C}_\theta, R(\mathbf{C}_\theta)) \le \epsilon$.*
2. *There exists $\theta_* \in \Theta_\epsilon$ whose decoded tree attains $\mathcal{H}^{\mathcal{T}_{\theta_*}}_{\Phi}(H) = \mathcal{H}^{\star}_{\Phi,\mathfrak{T}}(H)$.*
3. *The returned parameter $\theta_\eta \in \Theta_\epsilon$ is an η-optimal solution of the relaxed-HSE problem over this same family, for $\eta \ge 0$:*

$$\widehat{\mathcal{H}}_{\Phi}(H; \mathbf{C}_{\theta_\eta}) \le \inf_{\theta\in\Theta_\epsilon} \widehat{\mathcal{H}}_{\Phi}(H; \mathbf{C}_\theta) + \eta. \tag{62}$$

Then

$$0 \le \mathcal{H}^{\mathcal{T}_{\theta_\eta}}_{\Phi}(H) - \mathcal{H}^{\star}_{\Phi,\mathfrak{T}}(H) \le 2L\epsilon + \eta. \tag{63}$$

For an attained relaxed optimum, $\eta = 0$, and the gap is $O(\epsilon)$ with L independent of ϵ.

Proof. The first assumption and Lemma 1 imply

$$\left|\widehat{\mathcal{H}}_{\Phi}(H; \mathbf{C}_\theta) - \mathcal{H}^{\mathcal{T}_\theta}_{\Phi}(H)\right| \le L\epsilon, \qquad \theta \in \Theta_\epsilon. \tag{64}$$

Applying this bound at θ_η and θ_*, together with the other two assumptions, gives

$$\begin{aligned} \mathcal{H}^{\mathcal{T}_{\theta_\eta}}_{\Phi}(H) &\le \widehat{\mathcal{H}}_{\Phi}(H; \mathbf{C}_{\theta_\eta}) + L\epsilon \\ &\le \widehat{\mathcal{H}}_{\Phi}(H; \mathbf{C}_{\theta_*}) + \eta + L\epsilon \\ &\le \mathcal{H}^{\star}_{\Phi,\mathfrak{T}}(H) + 2L\epsilon + \eta. \end{aligned} \tag{65}$$

The lower bound follows from $\mathcal{T}_{\theta_\eta} \in \mathfrak{T}$ and the definition of the discrete optimum. □

Approximate representability. If the second assumption is weakened to $\mathcal{H}_{\Phi}^{\mathcal{T}_{\theta^*}}(H) \le \mathcal{H}_{\Phi,\mathfrak{T}}^{\star}(H)+\zeta$ for some $\zeta \ge 0$, the same proof gives the bound $\zeta + 2L\epsilon + \eta$. In particular, this bound is $O(\epsilon)$ when $\zeta, \eta = O(\epsilon)$ and the uniform rounding condition holds.

Compatibility with the output partition. Proposition 1 decodes the entire hierarchy by rounding each $\mathbf{C}^h$, whereas Section 5 assigns final cluster labels by row-wise argmax of $\mathbf{S}^1$. Under its first assumption these two procedures yield the same level-1 labels. Indeed, fix a leaf and follow its path in $R(\mathbf{C}_\theta)$. If the probability of taking the prescribed parent at level h is $1 - a_h$, then $0 \le a_h \le \|\mathbf{C}_\theta^h - R(\mathbf{C}_\theta^h)\|_F$. The corresponding entry of $\mathbf{S}^1$ is at least the probability of this path, $\prod_{h=2}^{K}(1-a_h) \ge 1 - \sum_{h=2}^{K} a_h \ge 1 - \epsilon > 1/2$. It is therefore the unique maximum in that row.

Scope. The parameter family Θ_ϵ may depend on ϵ, but the constants r, L do not. The uniform rounding, representability, and optimization conditions are additional assumptions; they are not guaranteed by softmax parameterization or orthogonality regularization alone. These results apply to a fixed hypergraph, including a frozen enhanced hypergraph, and do not by themselves establish convergence or global optimality of joint topology and representation learning under the composite HSENet loss.

B.4 Conductance under Topology Fusion

We now formalize when the representation-induced topology provably improves the generalized conductance of the observed topology. The result is an EDVW hypergraph counterpart of the graph-fusion analysis in Sun et al. (2026), but applies to any splitting function satisfying the conditions in the preliminaries.

For clarity, let $H_{\mathrm{o}} = (V, E_{\mathrm{o}}, \omega_{\mathrm{o}}, \gamma_{\mathrm{o}})$ and $H_{\mathrm{r}} = (V, E_{\mathrm{r}}, \omega_{\mathrm{r}}, \gamma_{\mathrm{r}})$ denote an observed and a representation-induced EDVW hypergraph, respectively, with disjoint indexed hyperedge families. For $\beta > 0$, define their fusion H_β on $E_{\mathrm{o}} \uplus E_{\mathrm{r}}$ by retaining ω_{o} on E_{o} and assigning $\beta\omega_{\mathrm{r}}$ on E_{r}. The construction in Section 5.2 is the special case $H_{\mathrm{o}} = H^{\mathrm{orig}}$, $H_{\mathrm{r}} = H^{\mathrm{rep}}$, $\omega_{\mathrm{r}} \equiv 1$, and $H_\beta = H^{\mathrm{enh}}$.

For $q \in \{\mathrm{o}, \mathrm{r}, \beta\}$ and $S \subset V$, write

$$g_q(S) = g_{\Phi,H_q}(S), \qquad v_q(S) = \mathrm{vol}_{H_q}(S), \qquad d_q(S) = \min\{v_q(S), v_q(V \setminus S)\}. \tag{66}$$

Whenever $d_q(S) > 0$, define the local generalized conductance

$$\phi_q(S) = \frac{g_q(S)}{d_q(S)}. \tag{67}$$

Proposition 2 (Generalized EDVW conductance under topology fusion)**.** *Fix $S \subset V$ with $d_{\mathrm{o}}(S) > 0$.*

1. *If $d_{\mathrm{r}}(S) > 0$, then*

$$\phi_\beta(S) \le \frac{d_{\mathrm{o}}(S)\phi_{\mathrm{o}}(S) + \beta d_{\mathrm{r}}(S)\phi_{\mathrm{r}}(S)}{d_{\mathrm{o}}(S) + \beta d_{\mathrm{r}}(S)} \le \max\{\phi_{\mathrm{o}}(S), \phi_{\mathrm{r}}(S)\}. \tag{68}$$

   Consequently, if $\phi_{\mathrm{r}}(S) \le \phi_{\mathrm{o}}(S)$, then

$$\phi_\beta(S) \le \phi_{\mathrm{o}}(S). \tag{69}$$

   If $\phi_{\mathrm{r}}(S) < \phi_{\mathrm{o}}(S)$, then the inequality is strict for every $\beta > 0$.

2. *If $d_{\mathrm{r}}(S) = 0$, then $g_{\mathrm{r}}(S) = 0$ and*

$$\phi_\beta(S) \le \phi_{\mathrm{o}}(S). \tag{70}$$

Therefore, if $\phi_{\mathrm{r}}(S) \le \phi_{\mathrm{o}}(S)$ for every subset with $d_{\mathrm{o}}(S), d_{\mathrm{r}}(S) > 0$, then

$$\mathrm{Cond}_\Phi(H_\beta) \le \mathrm{Cond}_\Phi(H_{\mathrm{o}}). \tag{71}$$

Moreover, the global inequality is strict if an optimal-conductance subset S_{o}^* of H_{o} satisfies $d_{\mathrm{r}}(S_{\mathrm{o}}^*) > 0$ and $\phi_{\mathrm{r}}(S_{\mathrm{o}}^*) < \phi_{\mathrm{o}}(S_{\mathrm{o}}^*)$.

Proof. Because the two hyperedge families are disjoint and hyperedge weights enter the generalized cut linearly, we have

$$g_\beta(S) = g_{\mathrm{o}}(S) + \beta g_{\mathrm{r}}(S). \tag{72}$$

The EDVW degree and volume are also linear in the hyperedge weights. Hence, for $A \in \{S, V \setminus S\}$,

$$v_\beta(A) = v_{\mathrm{o}}(A) + \beta v_{\mathrm{r}}(A). \tag{73}$$

For arbitrary nonnegative a, b, c, d and $\beta \geq 0$,

$$\min\{a + \beta c, b + \beta d\} \geq \min\{a, b\} + \beta \min\{c, d\}. \tag{74}$$

Indeed, both $a + \beta c$ and $b + \beta d$ are at least the right-hand side. Applying Eq. 74 to the two side-volumes in Eq. 73 gives

$$d_\beta(S) \geq d_{\mathrm{o}}(S) + \beta d_{\mathrm{r}}(S). \tag{75}$$

Suppose first that $d_{\mathrm{r}}(S) > 0$. Combining Eqs. 72 and 75 yields

$$\begin{aligned} \phi_\beta(S) &= \frac{g_{\mathrm{o}}(S) + \beta g_{\mathrm{r}}(S)}{d_\beta(S)} \\ &\leq \frac{g_{\mathrm{o}}(S) + \beta g_{\mathrm{r}}(S)}{d_{\mathrm{o}}(S) + \beta d_{\mathrm{r}}(S)} \\ &= \frac{d_{\mathrm{o}}(S)\phi_{\mathrm{o}}(S) + \beta d_{\mathrm{r}}(S)\phi_{\mathrm{r}}(S)}{d_{\mathrm{o}}(S) + \beta d_{\mathrm{r}}(S)}. \end{aligned} \tag{76}$$

The last expression is a weighted average of $\phi_{\mathrm{o}}(S)$ and $\phi_{\mathrm{r}}(S)$, proving Eq. 68 and the non-strict comparison. If $\phi_{\mathrm{r}}(S) < \phi_{\mathrm{o}}(S)$, both averaging weights are positive for every $\beta > 0$, so the weighted average is strictly smaller than $\phi_{\mathrm{o}}(S)$; the preceding upper bound therefore proves strict improvement.

If $d_{\mathrm{r}}(S) = 0$, one side of the cut has zero representation-induced volume. Since all retained incidences and hyperedge weights are positive, no representation-induced hyperedge can have positive mass on that side. Every representation-induced hyperedge is therefore unsplit by $(S, V \setminus S)$, and the endpoint conditions on Φ imply $g_{\mathrm{r}}(S) = 0$. Equations 72 and 75 then give

$$\phi_\beta(S) = \frac{g_{\mathrm{o}}(S)}{d_\beta(S)} \leq \frac{g_{\mathrm{o}}(S)}{d_{\mathrm{o}}(S)} = \phi_{\mathrm{o}}(S). \tag{77}$$

For the global statement, let S_{o}^* minimize the generalized conductance of H_{o}. Its two sides have positive observed volume and therefore also positive fused volume. Applying the established pointwise comparison at S_{o}^* gives

$$\mathrm{Cond}_\Phi(H_\beta) \leq \phi_\beta(S_{\mathrm{o}}^*) \leq \phi_{\mathrm{o}}(S_{\mathrm{o}}^*) = \mathrm{Cond}_\Phi(H_{\mathrm{o}}). \tag{78}$$

If the premise is strict at S_{o}^* and $d_{\mathrm{r}}(S_{\mathrm{o}}^*) > 0$, the pointwise strict result makes the global inequality strict as well. □

Corollary 1 (Cluster-pure representation-induced hyperedges)**.** *Fix $S \subset V$ with $d_{\mathrm{o}}(S), d_{\mathrm{r}}(S) > 0$. If every representation-induced hyperedge lies entirely in S or entirely in $V \setminus S$, then*

$$\phi_{\mathrm{r}}(S) = 0 \qquad \textit{and} \qquad \phi_\beta(S) \leq \phi_{\mathrm{o}}(S). \tag{79}$$

The second inequality is strict whenever $\phi_{\mathrm{o}}(S) > 0$.

Proof. No representation-induced hyperedge is split by $(S, V \setminus S)$, so the endpoint conditions on Φ give $g_{\mathrm{r}}(S) = 0$. Since $d_{\mathrm{r}}(S) > 0$, this implies $\phi_{\mathrm{r}}(S) = 0$. The conclusion follows from Proposition 2. □

Remark 1. Proposition 2 is conditional: the positive-affinity TopK construction does not by itself guarantee $\phi_{\mathrm{r}}(S) \leq \phi_{\mathrm{o}}(S)$. Rather, the proposition isolates the topology-quality condition under which fusion is provably safe. Because β scales the hyperedge weights rather than the EDVW memberships, the proof requires neither homogeneity nor differentiability of Φ and therefore applies to all splitting functions considered in the preliminaries. Combined with Theorem 4.2, the result explains why minimizing HSE on a sufficiently cluster-consistent enhanced topology aligns with low-conductance clustering. It does not claim that the HSE value itself is monotone under fusion, because fusion also changes the degree distribution and all module-volume ratios.

B.5 Orthogonality Regularization and Cluster Sizes

We justify that the regularizer in Eq. 20 encourages hard, mutually exclusive assignments without requiring equal cluster sizes. Specifically, we show that, under positive column masses, zero loss is attained exactly by hard assignments, and that every such assignment attains zero loss even when the parent clusters have different sizes.

Let $K \geq 2$ and fix a coarsening level $h \in \{2, \dots, K\}$ with $1 \leq n_{h-1} \leq n_h$. Write $\mathbf{C} = \mathbf{C}^h \in [0,1]^{n_h \times n_{h-1}}$ and assume that each row sums to one. For each parent cluster k, assume positive column mass $s_k = \sum_{i=1}^{n_h} \mathbf{C}_{ik} > 0$ and define

$$\widetilde{\mathbf{C}}_{ik} = \frac{\mathbf{C}_{ik}}{\sqrt{s_k}}, \qquad \mathcal{L}_h = \left\| (\widetilde{\mathbf{C}})^\top \widetilde{\mathbf{C}} - \mathbf{I}_{n_{h-1}} \right\|_F^2. \tag{80}$$

Proposition 3 (Zero-loss characterization of orthogonality regularization)**.** *Under the preceding assumptions,*

$$\mathcal{L}_h = 0 \quad \Longleftrightarrow \quad \textit{every row of } \mathbf{C}^h \textit{ is one-hot.} \tag{81}$$

If these assumptions hold at every coarsening level, then $\mathcal{L}_{\text{orth}} = 0$ if and only if every row of every $\mathbf{C}^h$, $h = 2, \dots, K$, is one-hot.

Proof. **Zero loss implies hard assignments.** If $\mathcal{L}_h = 0$, then $(\widetilde{\mathbf{C}})^\top \widetilde{\mathbf{C}} = \mathbf{I}_{n_{h-1}}$, so every normalized column has unit squared norm. For each column k,

$$\sum_{i=1}^{n_h} \widetilde{\mathbf{C}}_{ik}^2 = \frac{\sum_i \mathbf{C}_{ik}^2}{s_k} = 1 - \frac{\sum_i \mathbf{C}_{ik}(1 - \mathbf{C}_{ik})}{s_k} \leq 1. \tag{82}$$

Since $s_k > 0$ and each summand $\mathbf{C}_{ik}(1 - \mathbf{C}_{ik})$ is nonnegative, unit squared norm forces every entry in column k to belong to $\{0, 1\}$. This holds for every column. Each row sums to one and therefore contains exactly one 1.

Hard assignments imply zero loss. Conversely, suppose that every row of $\mathbf{C}$ is one-hot. Then $\mathbf{C}_{ik}^2 = \mathbf{C}_{ik}$, so the preceding identity gives unit squared norm for every normalized column. For distinct columns $a \neq b$, one-hot rows give $\mathbf{C}_{ia}\mathbf{C}_{ib} = 0$ for every i, hence

$$\sum_{i=1}^{n_h} \widetilde{\mathbf{C}}_{ia}\widetilde{\mathbf{C}}_{ib} = \frac{\sum_i \mathbf{C}_{ia}\mathbf{C}_{ib}}{\sqrt{s_a s_b}} = 0. \tag{83}$$

Thus $(\widetilde{\mathbf{C}})^\top \widetilde{\mathbf{C}} = \mathbf{I}_{n_{h-1}}$, and $\mathcal{L}_h = 0$.

Extension to all coarsening levels. Equation 20 gives $\mathcal{L}_{\text{orth}} = (K-1)^{-1} \sum_{h=2}^{K} \mathcal{L}_h$. Since every $\mathcal{L}_h \geq 0$, the average is zero exactly when all level-wise losses are zero. Applying the single-level equivalence at each level proves the final claim. □

Remark 2. Under a hard assignment, s_k counts the level-h nodes assigned to parent k, so the positive-column-mass assumption means that every parent cluster is nonempty. Every such assignment attains the minimum value $\mathcal{L}_h = 0$, since the loss is nonnegative and these assignments exist when $n_{h-1} \leq n_h$. The counts s_k need not be equal: column-mass normalization gives each hard-assignment column unit norm regardless of its size. These counts refer to level-h nodes, rather than the original vertices or hypergraph volumes they represent. The proposition characterizes zero-loss assignments; it does not guarantee that training reaches them or that rounding a soft assignment produces nonempty clusters.

C HSENet Details

C.1 Overall Algorithm

Algorithm 1 summarizes HSENet training and inference. At each training step, representations from the uncorrupted input are used for topology enhancement and hierarchical assignment, while masked feature reconstruction and hyperedge filling jointly supervise the shared encoder.

Algorithm 1 Training and inference of HSENet

Input: Observed incidences $\mathbf{H}^{\mathrm{orig}}$, features $\mathbf{X}$, and hyperedge weights $\boldsymbol{\omega}^{\mathrm{orig}}$;
tree depth K, level sizes $\{n_h\}_{h=1}^{K-1}$, and splitting function Φ;
edge weight β, loss weights $\lambda, \lambda_1, \lambda_2$, and training steps T.
Output: Vertex cluster labels $\{\hat{y}_i\}_{i=1}^{n}$.

1: Initialize the encoder, decoder, mask tokens, and assignment-layer parameters.
2: **for** $t = 1, \ldots, T$ **do**
// Generative representation learning
3: Mask vertex features, encode them, and re-mask the selected latent representations.
4: Decode the masked features and compute $\mathcal{L}_{\mathrm{MFR}}$ on masked vertices.
5: Sample feature corruption and hyperedge dropout.
6: Encode the perturbed input and compute $\mathcal{L}_{\mathrm{HF}}$ from valid target–hyperedge pairs.
7: $\mathcal{L}_{\mathrm{rep}} \leftarrow \lambda_1 \mathcal{L}_{\mathrm{MFR}} + \lambda_2 \mathcal{L}_{\mathrm{HF}}$.
// Dynamic EDVW topology enhancement
8: $\mathbf{Z}_v \leftarrow f_{\mathrm{enc}}(E, \mathbf{X})$ using the uncorrupted input.
9: Mean-pool each observed hyperedge to obtain $\mathbf{z}_{e_j}$.
10: $\mathbf{W}_{ij} \leftarrow \max\{\cos(\mathbf{z}_i, \mathbf{z}_{e_j}), 0\}$ for all i, j.
11: Retain up to r largest positive entries per column of $\mathbf{W}$.
12: Denote the resulting incidence matrix by $\mathbf{H}^{\mathrm{rep}}$, with m_{rep} columns.
13: $\mathbf{H}^{\mathrm{enh}} \leftarrow [\mathbf{H}^{\mathrm{orig}} \mid \mathbf{H}^{\mathrm{rep}}]$; $\boldsymbol{\omega}^{\mathrm{enh}} \leftarrow [(\boldsymbol{\omega}^{\mathrm{orig}})^\top, \beta \mathbf{1}_{m_{\mathrm{rep}}}^\top]^\top$.
// Hierarchical assignment and coarsening
14: $\mathbf{Z}^K \leftarrow \mathbf{Z}_v$, $\mathbf{H}^K \leftarrow \mathbf{H}^{\mathrm{enh}}$, and $\mathbf{C}^1 \leftarrow \mathbf{1}_{n_1 \times 1}$.
15: **for** $h = K, \ldots, 2$ **do**
16: $\mathbf{C}^h \leftarrow \mathrm{softmax}(\mathcal{A}^h(\mathbf{Z}^h, \mathbf{H}^h, \boldsymbol{\omega}^{\mathrm{enh}}))$. ▷ Appendix C.3
17: **if** $h > 2$ **then**
18: $\mathbf{H}^{h-1} \leftarrow \mathcal{S}_b((\mathbf{C}^h)^\top \mathbf{H}^h)$.
19: $\mathbf{Z}^{h-1} \leftarrow \mathrm{diag}((\mathbf{C}^h)^\top \mathbf{1})^{-1} (\mathbf{C}^h)^\top \mathbf{Z}^h$.
20: **end if**
21: **end for**
22: $\mathbf{S}^1 \leftarrow \mathbf{C}^K \cdots \mathbf{C}^2$.
// Joint optimization on the enhanced leaf hypergraph
23: $\mathcal{L}_{\mathrm{hse}} \leftarrow \widehat{\mathcal{H}}_\Phi(H^{\mathrm{enh}}; \{\mathbf{C}^h\}_{h=1}^K)$ via Eq. 10.
24: Compute $\mathcal{L}_{\mathrm{orth}}$ via Eq. 20.
25: $\mathcal{L}_{\mathrm{HSENet}} \leftarrow \mathcal{L}_{\mathrm{hse}} + \lambda \mathcal{L}_{\mathrm{orth}} + \mathcal{L}_{\mathrm{rep}}$.
26: Update all trainable parameters by backpropagation through $\mathcal{L}_{\mathrm{HSENet}}$.
27: **end for**
28: Repeat lines 8–22 with the trained model for evaluation.
29: **return** $\hat{y}_i = \arg\max_{k=1,\ldots,n_1} \mathbf{S}^1_{ik}$ for each vertex v_i.

Within each assignment layer, we temporarily add a unit-weight singleton hyperedge for each node to preserve its own representation during message passing (Appendix C.3).

C.2 Topology Enhancement Details

Representation-Induced Incidences. Let $\mathbf{Z}_v = f_{\mathrm{enc}}(E, \mathbf{X})$ denote the vertex representations from the uncorrupted input and $\mathbf{H}^{\mathrm{orig}} \in \{0,1\}^{n \times m}$ the observed incidence matrix used in our experiments, while the theory permits general EDVW observed incidences. For each observed hyperedge e_j, we mean-pool its vertex representations and compute affinities across the full vertex set:

$$\mathbf{z}_{e_j} = \frac{1}{|e_j|} \sum_{v_i \in e_j} \mathbf{z}_i, \qquad \mathbf{W}_{ij} = \max\{\cos(\mathbf{z}_i, \mathbf{z}_{e_j}), 0\}. \tag{84}$$

Let $\mathcal{I}_j$ contain the indices of the r largest strictly positive entries in $\mathbf{W}_{:j}$, or all such entries if fewer than r exist. We define the temporary incidence matrix

$$\widetilde{\mathbf{H}}^{\mathrm{rep}}_{ij} = \mathbf{W}_{ij}\, \mathbb{I}\{i \in \mathcal{I}_j\}, \qquad \widetilde{\mathbf{H}}^{\mathrm{rep}} \in \mathbb{R}^{n \times m}_{\geq 0}. \tag{85}$$

Removing all-zero columns yields $\mathbf{H}^{\mathrm{rep}} \in \mathbb{R}^{n \times m_{\mathrm{rep}}}_{\geq 0}$, whose nonzero entries are the EDVW memberships of the induced hyperedges. Columns generated from different observed hyperedges remain distinct even if their vertex supports coincide.

Topology Fusion. We concatenate the observed and induced incidences and their corresponding hyperedge weights:

$$\mathbf{H}^{\mathrm{enh}} = \left[\mathbf{H}^{\mathrm{orig}} \,\middle\|\, \mathbf{H}^{\mathrm{rep}}\right], \qquad \boldsymbol{\omega}^{\mathrm{enh}} = \begin{bmatrix} \boldsymbol{\omega}^{\mathrm{orig}} \\ \beta \mathbf{1}_{m_{\mathrm{rep}}} \end{bmatrix}, \qquad \beta > 0, \tag{86}$$

where $\boldsymbol{\omega}^{\mathrm{orig}} = (\omega(e_1), \ldots, \omega(e_m))^\top$. The columns of $\mathbf{H}^{\mathrm{enh}}$ define the indexed hyperedge family E^{enh}, with ω^{enh} given by the corresponding entries of $\boldsymbol{\omega}^{\mathrm{enh}}$. Thus, β sets the weight of each induced hyperedge, while its EDVW memberships remain the selected affinities.

The enhanced topology is recomputed at each forward pass. Gradients do not propagate through discrete support selection (positivity tests and TopK indices), but flow through the retained affinities to the encoder.

C.3 EDVW-Aware Assignment and Coarsening Details

The hierarchical assigner uses both learned representations and EDVW membership strengths to predict parent assignments at each level. We initialize $\mathbf{Z}^K = \mathbf{Z}_v$ and $\mathbf{H}^K = \mathbf{H}^{\mathrm{enh}}$, with $M = m + m_{\mathrm{rep}}$ structural hyperedge columns retained across levels.

EDVW-Aware Assignment Layer. For $h = K, \ldots, 2$, the assignment layer, inspired by Uni-GAT Huang & Yang (2021), incorporates EDVW weights into both node-to-hyperedge pooling and hyperedge-to-node attention. To retain direct access to each node's own representation during aggregation, we add temporary unit-weight singleton hyperedges:

$$\overline{\mathbf{H}}^h = \left[\mathbf{H}^h \,\middle\|\, \mathbf{I}_{n_h}\right], \qquad \overline{\boldsymbol{\omega}}^h = \begin{bmatrix} \boldsymbol{\omega}^{\mathrm{enh}} \\ \mathbf{1}_{n_h} \end{bmatrix}. \tag{87}$$

For each nonempty column j, we pool node representations according to their relative EDVW memberships:

$$\overline{\mathbf{e}}_j^h = \frac{\sum_{i=1}^{n_h} \overline{\mathbf{H}}_{ij}^h \mathbf{z}_i^h}{\sum_{i=1}^{n_h} \overline{\mathbf{H}}_{ij}^h}. \tag{88}$$

This weighted average gives stronger memberships greater influence, while normalization by the total membership mass makes the representation invariant to uniform rescaling of a column. For a singleton column, pooling returns the node's own representation; its presence also ensures that every node has a nonempty attention neighborhood.

Let $\overline{\mathcal{E}}_i^h = \{j \mid \overline{\mathbf{H}}_{ij}^h > 0\}$ denote the incident hyperedges of node i. Attention combines learned node–hyperedge compatibility with the structural prior $\overline{\omega}_j^h \overline{\mathbf{H}}_{ij}^h$, so that aggregation accounts for both hyperedge importance and the node's membership strength. Each head $k = 1, \ldots, n_{h-1}$ learns an aggregation for one candidate parent and projects it to a scalar assignment logit, using head-specific parameters $\mathbf{W}_{\mathrm{gat}}^{h,k}$, $\mathbf{a}^{h,k}$, and $\mathbf{w}_o^{h,k}$:

$$s_{ij}^{h,k} = \sigma\Big((\mathbf{a}^{h,k})^\top [\mathbf{W}_{\mathrm{gat}}^{h,k} \mathbf{z}_i^h \,\|\, \mathbf{W}_{\mathrm{gat}}^{h,k} \overline{\mathbf{e}}_j^h]\Big), \tag{89}$$

$$\alpha_{ij}^{h,k} = \frac{\overline{\omega}_j^h \overline{\mathbf{H}}_{ij}^h \exp(s_{ij}^{h,k})}{\sum_{l \in \overline{\mathcal{E}}_i^h} \overline{\omega}_l^h \overline{\mathbf{H}}_{il}^h \exp(s_{il}^{h,k})}, \qquad j \in \overline{\mathcal{E}}_i^h, \tag{90}$$

$$\ell_{ik}^h = (\mathbf{w}_o^{h,k})^\top \sigma\left(\sum_{j \in \overline{\mathcal{E}}_i^h} \alpha_{ij}^{h,k} \mathbf{W}_{\mathrm{gat}}^{h,k} \overline{\mathbf{e}}_j^h\right), \tag{91}$$

$$\mathbf{C}_{i:}^h = \mathrm{softmax}\big((\ell_{ik}^h)_{k=1}^{n_{h-1}}\big). \tag{92}$$

Here σ is LeakyReLU, $\|$ denotes concatenation, and softmax is taken over parent clusters. The final softmax converts the parent-specific logits into a row-stochastic soft assignment, allowing the hierarchy to be optimized with the relaxed HSE objective.

Hierarchical Coarsening. We coarsen structural memberships and node representations using Eq. 18. The intermediate matrix $\widetilde{\mathbf{H}}^{h-1} = (\mathbf{C}^h)^\top \mathbf{H}^h$ aggregates child memberships into parent-to-hyperedge incidences. Node representations are averaged with the same soft assignments and normalized by each parent's assignment mass to avoid scaling feature magnitudes with cluster size.

Soft assignments can make $\widetilde{\mathbf{H}}^{h-1}$ dense, so we apply row-wise Top-b sparsification to control the connectivity used by the next assignment layer. For each parent node c, let $\mathcal{J}_c$ contain the column indices of the largest $\min\{b, M\}$ entries in row c of $\widetilde{\mathbf{H}}^{h-1}$, where $b \geq 1$ is an integer and ties are broken arbitrarily. We retain these entries and set the others to zero:

$$\mathbf{H}^{h-1}_{cj} = \begin{cases} \widetilde{\mathbf{H}}^{h-1}_{cj}, & j \in \mathcal{J}_c, \\ 0, & \text{otherwise.} \end{cases} \tag{93}$$

This operation, denoted by $\mathcal{S}_b$ in Eq. 18, retains at most b nonzero incidences per parent node without renormalizing the retained weights. Column identities and the structural hyperedge-weight vector are preserved across levels; all-zero columns are excluded from message aggregation. At each forward pass, the selected indices are recomputed and treated as fixed during backpropagation, while gradients pass through retained values to $\mathbf{C}^h$ and $\mathbf{H}^h$.

The leaf input $\mathbf{H}^K = \mathbf{H}^{\text{enh}}$ is not subject to this coarsening sparsification. At subsequent assignment levels, message passing and further coarsening use the sparsified structural matrix. Temporary singleton columns are added only within each assignment layer and enter neither structural coarsening nor the HSE objective. The HSE objective is evaluated on the enhanced leaf hypergraph, while the coarsened matrices guide the construction of the assignment hierarchy.

C.4 Computational Complexity

We analyze one full-batch training step with sparse incidence operations and linear-time Top-k selection. Encoder/decoder depths, tree depth K, and internal level sizes $n_1, \ldots, n_{K-1}$ are treated as constants. Let $I_0 = \text{nnz}(\mathbf{H}^{\text{orig}})$ and $I = \text{nnz}(\mathbf{H}^{\text{enh}}) \leq I_0 + mr$, where $r \leq n$; d and d_z denote the input feature dimension and hidden-width bound.

Encoding, feature reconstruction, and hierarchical assignment/coarsening together cost $O(ndd_z + (n+m)d_z^2 + I_0 d + Id_z)$. Exhaustive hyperedge filling scores at most I_0 queries against all n vertices in $O(nI_0 d_z)$ time, reusing hyperedge embedding sums. Topology enhancement computes all vertex–hyperedge similarities before Top-r selection, costing $O(nmd_z)$. For the fixed internal level sizes, assignment composition and relaxed HSE evaluation cost $O(n + I \log n)$, with the logarithmic factor arising from sorting memberships within hyperedges for the Lovász extension; orthogonality regularization costs $O(n)$.

Combining these costs gives the per-step upper bound

$$\begin{aligned} O\Big(& ndd_z + (n+m)d_z^2 + I_0 d \\ & + n(I_0 + m)d_z + (I_0 + mr)\log n \Big). \end{aligned} \tag{94}$$

T training steps multiply the bound by T.

With sampling, scoring q filling queries against s candidates per query costs $O(qsd_z)$, while encoding and $O(I_0 d_z)$ query preprocessing remain necessary. Our sampling variants use at most 32 queries per epoch for large hyperedges and 257 candidates per query (256 negatives and the target) for large vertex sets. Inference requires one clean encoder pass, topology enhancement, and hierarchical assignment followed by argmax decoding, without reconstruction or training-loss evaluation.

D Experimental Details

D.1 Datasets

Table 3 summarizes the statistics of the 8 datasets used in our experiments.

Table 3: Statistics of datasets used in our experiments.

Statistic	Zoo	Cora	Citeseer	Pubmed	Mushroom	20News	Aminer	DBLP
# Vertices	101	1,434	1,458	3,840	8,124	16,242	20,201	41,302
# Hyperedges	43	1,579	1,079	7,963	298	100	8,052	22,363
# Incidences	1,717	4,786	3,453	34,629	40,620	65,451	32,028	99,561
Avg. hyperedge size	39.93	3.03	3.20	4.35	136.31	654.51	3.98	4.45
Avg. vertex degree	17.00	3.34	2.37	9.02	5.00	4.03	1.59	2.41
Max. hyperedge size	93	5	26	171	1,808	2,241	101	202
Max. vertex degree	17	145	88	99	5	44	18	18
# Features	16	1,433	3,703	500	22	100	500	1,425
# Classes	7	7	6	3	2	4	12	6

D.2 BASELINES

We compare HSENet with 10 baselines spanning classical attributed hypergraph clustering and neural hypergraph learning. The classical methods integrate attributes and topology through matrix factorization (JNMF), random walks (AHCKA), modularity maximization (AHRC), or hypergraph convolution followed by spectral clustering (GRAC). The neural methods include contrastive or correlation-based representation learning (TriCL, HyFi, and SE-HSSL), generative self-supervision (HypeBoy), and combined reconstruction and contrastive learning (SS-HT). CAHC further incorporates clustering-oriented supervision to jointly refine representations and cluster assignments.

• **JNMF** Du et al. (2019) jointly factorizes the attribute matrix using nonnegative matrix factorization and the structural similarity matrix using symmetric nonnegative matrix factorization, sharing a latent factor that directly encodes cluster assignments.

• **AHCKA** Li et al. (2023) augments the hypergraph with an attribute-based K-nearest-neighbor graph and optimizes a multi-hop conductance objective derived from a joint random walk over structural and attribute-based connections.

• **AHRC** Feng et al. (2025b) integrates hypergraph topology and vertex attributes into a graph representation, applies spanning-forest sparsification, and optimizes multi-hop modularity without requiring a predefined number of clusters.

• **GRAC** Fanseu Kamhoua et al. (2021) propagates vertex attributes through hypergraph convolution based on a nonlinear hypergraph Laplacian and applies spectral clustering to the resulting feature similarities.

• **TriCL** Lee & Shin (2023) learns hypergraph representations by contrasting two augmented views at the vertex, hyperedge, and vertex–hyperedge membership levels to capture both individual and group relations.

• **HyFi** Roh et al. (2024) preserves hypergraph topology through feature-noise augmentation and incorporates weak positive pairs weighted by shared-group information into vertex- and hyperedge-level contrastive learning.

• **SS-HT** Feng et al. (2025a) combines masked feature reconstruction with re-masking and structure-aware contrastive learning guided by an optimal-transport metric of local high-order structural changes.

• **HypeBoy** Kim et al. (2024) first warms up a hypergraph encoder through masked feature reconstruction and then learns representations through hyperedge filling, predicting a held-out vertex from the remaining members of a hyperedge.

• **SE-HSSL** Li et al. (2026) combines sampling-free canonical-correlation objectives at the vertex and hyperedge levels with a hierarchical membership-level contrastive objective that exploits hyperedge overlaps to reduce reliance on negative sampling.

• **CAHC** Ni et al. (2026) jointly optimizes vertex- and hyperedge-level contrastive learning with a clustering objective that aligns soft cluster assignments with centroid-based pseudo-labels, refining representations and cluster assignments together.

D.3 Implementation Details

We implement HSENet using PyTorch and PyTorch Geometric (PyG) and train it using the Adam optimizer with HGNN Feng et al. (2019) backbone. We conduct experiments on a server with an NVIDIA GeForce RTX A6000 GPU and 500 GB memory. Unless otherwise specified, we use the following settings. The partitioning-tree height is set to $K = 2$, yielding a hierarchy with vertex leaves at level 2, output clusters at level 1, and a single root at level 0. This setting learns vertex-to-cluster assignments directly, while larger values of K introduce intermediate grouping levels. We use a learning rate of 5×10^{-4} for all datasets except Citeseer and DBLP, for which we use 1×10^{-4}. We use the δ-Linear splitting function as default, defined as $\Phi_{\delta\text{-Lin}}(x, \Gamma) = \min\{\delta, x, \Gamma - x\}$.

In the joint training objective, the relaxed HSE loss has unit weight, and $\lambda = 0.05$ weights the orthogonality regularizer $\mathcal{L}_{\text{orth}}$, which encourages hard, mutually exclusive assignments and discourages assignment collapse. The coefficients $\lambda_1 = 60$ and $\lambda_2 = 120$ weight the masked feature reconstruction loss $\mathcal{L}_{\text{MFR}}$ and the hyperedge filling loss $\mathcal{L}_{\text{HF}}$, respectively (Eq. 15). The former encourages the shared encoder to preserve vertex attributes by reconstructing masked features, whereas the latter captures higher-order membership patterns by predicting a held-out vertex from its hyperedge context. These coefficients control the contributions of the two generative tasks to joint optimization.

For topology enhancement, we assign weight $\beta = 3$ to each representation-induced hyperedge while retaining the original hyperedge weights. Thus, β controls the influence of the learned relations relative to the observed topology in both hierarchical assignment and the HSE objective; the incidence weights within each induced hyperedge remain determined by representation affinities.

Hyperedge Retention during Coarsening. When testing tree heights $K > 2$, we coarsen the hypergraph between successive assignment layers. To prevent the coarsened incidence matrices from becoming dense, each parent node retains connections to at most b hyperedges, selected by the largest incidence values in its row (Eq. 93). For each dataset, we choose b based on the average vertex degree in the original hypergraph:

$$b = \max\left\{1, \text{round}_{\text{even}}\left(\frac{\text{nnz}(\mathbf{H}^{\text{orig}})}{n}\right)\right\}, \tag{95}$$

Here, $\text{nnz}(\mathbf{H}^{\text{orig}})$ counts the original vertex–hyperedge incidences, and n is the number of original vertices; their ratio is the average number of hyperedges incident to a vertex. We round this average to the nearest integer, resolving half-integer ties toward the nearest even integer, and impose a minimum of one. The value of b is computed before adding representation-induced hyperedges and kept fixed across all coarsening levels and tested tree heights for that dataset.

E Additional Experimental Results

E.1 Additional Ablation Results

Table 4 complements the NMI and ACC results in Table 2 with ARI and F1 for the same 6 ablation variants.

Ablation configurations. The **w/o Enhan.** variant removes the entire EDVW topology enhancement module and uses the observed hypergraph for assignment learning and HSE evaluation. For **w/o Lovász**, we directly evaluate the splitting function on soft membership mass instead of using its Lovász extension. Specifically, in Eq. 10, we replace each $\widehat{F}_{j,\Phi}(\mathbf{s}^h_{j,\alpha})$ with

$$\Phi\left(\sum_{i=1}^{n} \mathbf{S}^h_{i\alpha}\mathbf{H}^{\mathrm{enh}}_{ij}, \Gamma^{\mathrm{enh}}_j\right), \qquad \Gamma^{\mathrm{enh}}_j = \sum_{i=1}^{n} \mathbf{H}^{\mathrm{enh}}_{ij}, \tag{96}$$

while retaining the other terms of the HSE objective. For the k**-means** variant, we retrain the representation-learning model without the hierarchical assigner, the HSE loss, or the orthogonality loss, optimizing only $\mathcal{L}_{\mathrm{rep}} = \lambda_1\mathcal{L}_{\mathrm{MFR}} + \lambda_2\mathcal{L}_{\mathrm{HF}}$. We then apply k-means to the vertex embeddings $\mathbf{Z}_v$ obtained from the uncorrupted input to produce cluster assignments.

Results. Removing hyperedge filling reduces both metrics across all evaluated datasets, while removing feature reconstruction or EDVW enhancement generally lowers performance, supporting the complementary contributions of semantic and structural supervision. Replacing the HSE-guided assigner with k-means consistently reduces ARI but yields mixed F1 results, suggesting that improvements in overall partition agreement do not always translate into higher macro-F1. The variants without the Lovász extension or orthogonality regularization both underperform the full model overall, despite matching or slightly exceeding it on a few dataset-metric combinations. Taken together, these ablation results support the complementary contributions of the components to the overall clustering performance of HSENet.

Table 4: Additional ablation results in ARI and F1 (%). Results are mean±std of 5 independent runs. The best and second-best results are highlighted in **bold** and underline, respectively.

Dataset	Metric	w/o Recon.	w/o Filling	w/o Enhan.	k-means	w/o Lovász	w/o Orth.	HSENet
Cora	ARI	49.7±1.4	36.2±5.2	47.3±2.3	52.5±0.8	51.0±2.1	52.6±1.0	**52.9±0.9**
	F1	64.5±4.4	42.8±7.0	60.5±6.3	**71.4±0.7**	66.4±6.9	67.2±4.2	69.5±4.0
Citeseer	ARI	45.5±1.9	38.6±5.8	46.6±0.9	45.4±0.6	47.3±1.0	**47.9±0.7**	**47.9±0.7**
	F1	60.1±1.5	49.6±8.9	60.9±0.7	**62.9±2.5**	61.4±0.4	61.7±0.4	61.7±0.4
Pubmed	ARI	37.6±2.4	20.4±0.8	**40.5±1.5**	36.9±0.3	39.8±3.4	37.7±4.5	40.1±3.0
	F1	67.4±9.3	46.1±3.3	71.8±1.2	70.4±0.1	68.1±9.5	68.8±2.9	**72.1±1.8**
20News	ARI	**50.5±1.1**	42.4±3.9	47.4±2.9	49.3±2.2	49.8±0.6	41.9±1.2	50.3±1.2
	F1	**73.1±1.2**	64.2±7.8	70.9±1.3	71.3±2.2	72.9±1.1	57.2±2.2	73.0±1.7
Mushroom	ARI	40.5±17.1	48.0±7.1	54.0±0.4	51.4±2.9	53.9±0.4	**54.2±0.2**	**54.2±0.1**
	F1	80.4±8.9	84.4±2.6	86.6±0.1	85.7±1.0	86.6±0.1	**86.7±0.1**	**86.7±0.0**
Aminer	ARI	34.7±1.2	33.5±1.6	32.5±0.6	28.2±0.5	**35.7±0.5**	34.5±0.6	35.4±0.5
	F1	22.7±0.9	21.6±0.8	25.0±1.1	**28.0±0.4**	21.9±0.8	21.9±1.7	22.1±1.4

E.2 Additional Sensitivity Analysis

Figure 7 complements the NMI and ACC sensitivity analysis with ARI and F1 for the representation-hyperedge weight β, orthogonality regularization weight λ, splitting function Φ, and tree height K.

Coefficient sensitivity. Both metrics are generally stable for moderate β, although their responses to larger or smaller values vary. For λ, values near the default 0.05 provide a favorable balance, whereas strong regularization often reduces ARI and F1. These results support moderate coefficient choices without indicating a single optimum shared by all evaluation settings.

Structural choices. The δ-Linear splitting function is generally competitive in both metrics, while the relative performance of the other splitting functions varies. The clearest trend concerns tree height: $K = 2$ achieves the highest reported ARI and F1 throughout the evaluation, and deeper trees reduce performance and can encounter memory limitations. These observations further support the use of a shallow partitioning tree.

Reconstruction loss weights. Figure 8 evaluates the feature reconstruction weight λ_1 and hyperedge filling weight λ_2 across all four metrics. Most results vary modestly with λ_1, with some non-monotonic fluctuations. Increasing λ_2 from very small to intermediate values often improves performance, while further increases offer limited gains or cause declines. These trends favor balanced reconstruction weights.

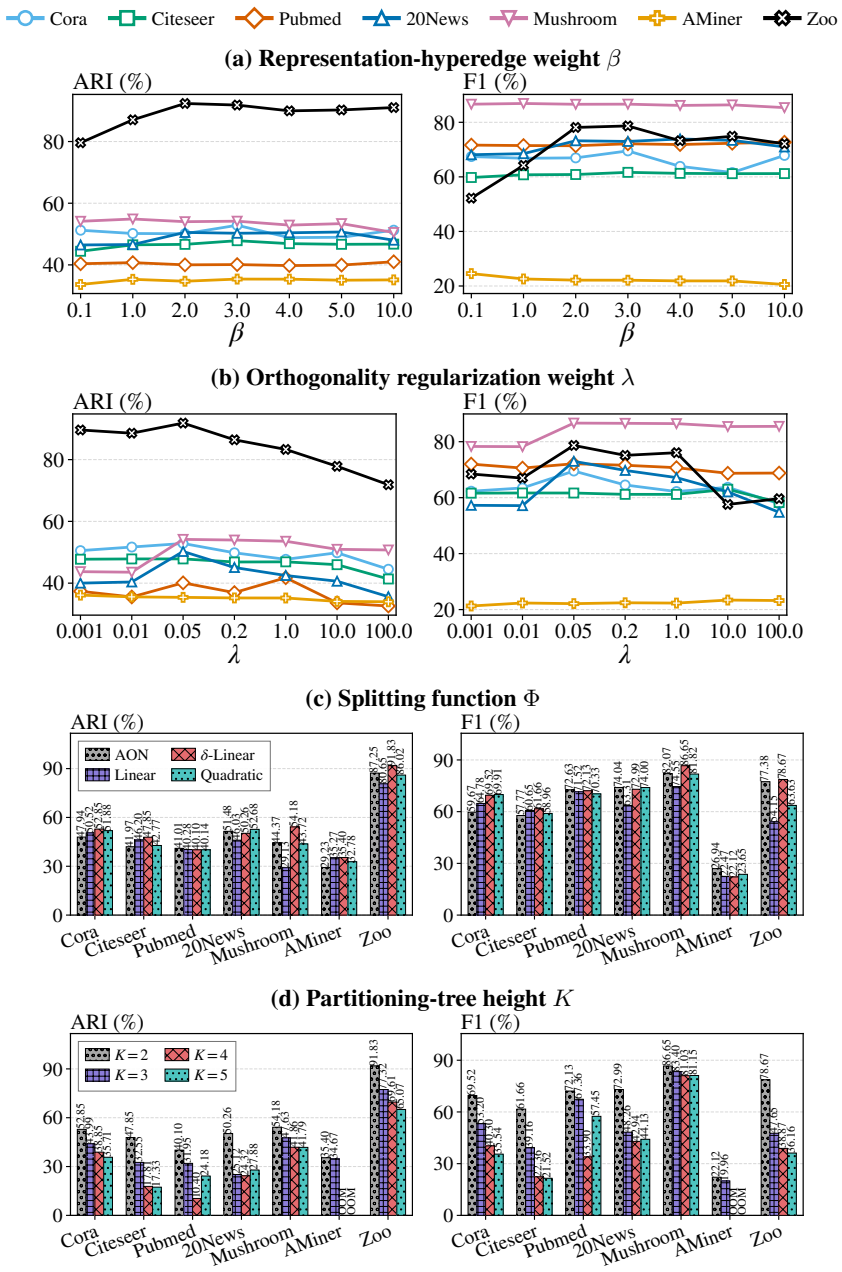


Figure 7: Additional sensitivity analysis in ARI (left) and F1 (right), reported as percentages. Rows examine β, λ, splitting function Φ, and tree height K, respectively. The shared dataset legend applies to the first two rows; the legends in the left panels apply to both panels in each of the last two rows. OOM denotes out of memory.

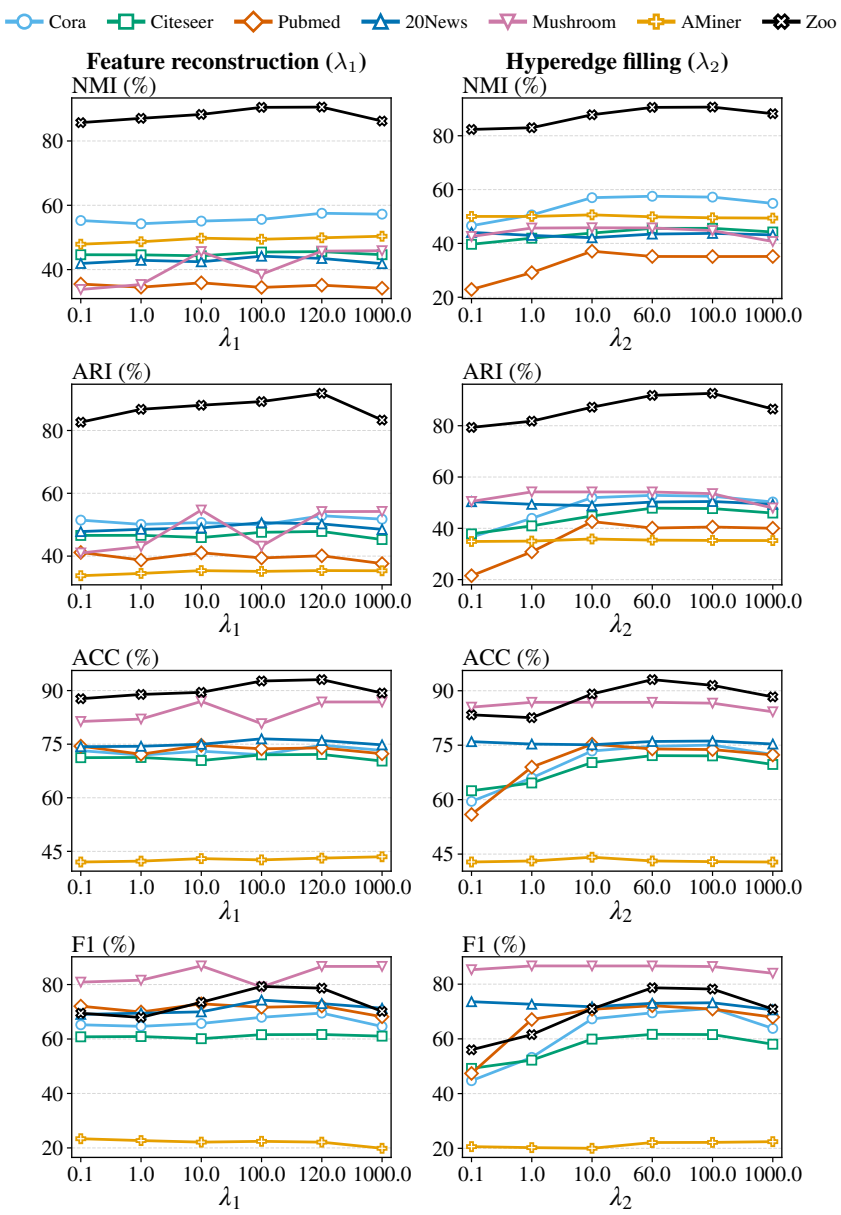


Figure 8: Sensitivity to the masked feature reconstruction weight λ_1 (left) and hyperedge filling weight λ_2 (right). Rows report NMI, ARI, ACC, and F1 as percentages. The shared legend applies to all panels.

E.3 Computational Efficiency

Table 5 compares per-epoch runtime and GPU memory usage across 8 datasets. HSENet achieves competitive per-epoch runtime and GPU memory usage among 6 baselines.

Table 5: Per-epoch runtime and GPU memory usage of 7 models on 8 datasets.

Dataset	Metric	TriCL	HyFi	SS-HT	HypeBoy	SE-HSSL	CAHC	HSENet
Cora	Time (s/epoch)	0.0963	0.1140	0.1019	0.0272	0.1015	0.1734	0.0460
	Memory (MB)	4860.4	353.6	413.7	232.8	5357.0	241.9	1001
Citeseer	Time (s/epoch)	0.0756	0.2781	0.0921	0.0326	0.0537	0.1419	0.0709
	Memory (MB)	3623.2	592.3	532.4	730.5	1452.2	274.2	963
Pubmed	Time (s/epoch)	0.5488	0.1424	0.3934	0.0867	0.2389	0.9553	0.0833
	Memory (MB)	5833.2	2778.7	3931.5	2171.4	5711.9	888.6	8159
DBLP	Time (s/epoch)	1.3025	OOM	OOM	OOM	OOT	1.9463	1.0663
	Memory (MB)	33498.0					29632.7	14681
20News	Time (s/epoch)	0.3437	0.3127	0.7587	0.0537	0.9386	0.3487	0.2859
	Memory (MB)	5169.8	8600.2	7911.8	4479.9	6680.5	4944.4	10709
Mushroom	Time (s/epoch)	0.6607	0.0820	0.3760	0.0439	0.4692	0.1232	0.0698
	Memory (MB)	9967.2	2440.9	2293.2	1608.5	5909.2	1515.6	2939
AMiner	Time (s/epoch)	0.2958	0.2731	OOM	0.0939	0.4234	0.7831	0.2386
	Memory (MB)	6709.7	5313.1		10157.1	1835.1	1711.6	11103
Zoo	Time (s/epoch)	0.0416	0.0374	0.0612	0.0292	0.0552	0.0541	0.0448
	Memory (MB)	123.3	30.2	34.7	163.4	4177.0	56.8	631

E.4 Robustness to Attribute Missing

HSENet uses vertex attributes to learn representations that also guide EDVW topology enhancement, so missing attributes may affect both semantic representations and the induced structural affinities used for clustering. Although masked feature reconstruction and hyperedge filling provide complementary training signals, their effectiveness on fully observed inputs does not establish robustness when the input attributes themselves are incomplete. We therefore progressively mask input attributes to assess how well HSENet preserves clustering quality as attribute information decreases and to examine which components continue to contribute under missing attributes.

Experimental setup. We evaluate HSENet under incomplete vertex attributes on 7 datasets: Cora, Citeseer, Pubmed, 20News, Mushroom, AMiner, and Zoo. We compare the full model with **w/o Recon.**, **w/o Filling**, and **w/o Enhan.**, which remove masked feature reconstruction, hyperedge filling, and EDVW topology enhancement, respectively, following the ablation study in Table 2. For a nominal masking ratio $r \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$, we set $\mathrm{round}(rd)$ attribute entries per vertex to zero, where d is the feature dimension. For each nonzero ratio and random seed, the attribute missingness mask is sampled once and kept fixed throughout training; all four variants share the same masked entries. The ratio r describes this imposed attribute missingness, distinct from the additional feature masking applied within the mask feature reconstruction task. Mask feature reconstruction uses the resulting incomplete feature vectors as reconstruction targets, retaining the imposed zeros in these targets rather than using the original fully observed features. We report the mean and standard deviation over 5 runs using NMI, ARI, ACC, and F1.

Overall robustness. Figure 9 and Table 6 demonstrate that HSENet is broadly robust to missing vertex attributes across most evaluated datasets. Even at a nominal masking ratio of $r = 0.6$, the model retains at least 82% of its unmasked reference NMI on 6 of the 7 datasets. Across the 4 metrics, performance generally degrades gradually as masking increases, with relatively small changes at low masking ratios and more pronounced losses under heavier masking on several datasets. The full model also generally compares favorably with the ablated variants under missing attributes; in particular, removing hyperedge filling leads to substantially poorer performance on the citation

datasets and Zoo. Table 6 further shows positive NMI gains from hyperedge filling on all 7 datasets at $r = 0.6$, supporting the value of structural supervision when attribute information is incomplete.

The degree of robustness nevertheless varies across datasets and metrics. Pubmed remains particularly stable, with NMI changing only from 0.3515 to 0.3511, while Zoo exhibits a relative NMI decrease of 5.1% at $r = 0.6$. The corresponding decreases on Cora, Citeseer, 20News, and AMiner range from 11.8% to 18.0%. Mushroom is the main exception: its NMI drops by 55.2%, from 0.4575 to 0.2051, with greater variability across runs under masking. Overall, these results indicate that HSENet preserves much of its clustering quality under substantial attribute missingness, while its tolerance to severe masking remains dataset dependent.

Table 6: NMI endpoints under attribute masking and gains from individual components. NMI is reported on the $[0, 1]$ scale as mean±std over 5 runs. Drop is the relative NMI decrease (%); component gains are in percentage points (pp).

Dataset	F	HSENet NMI		Drop (%)	Recon. (pp)		Filling (pp)		Enhan. (pp)	
		$r = 0$	$r = 0.6$		0	0.6	0	0.6	0	0.6
Cora	1433	0.5752 ± 0.0076	0.5075 ± 0.0160	11.8	$+3.4$	$+2.3$	$+10.9$	$+14.3$	$+4.7$	$+0.4$
Citeseer	3703	0.4559 ± 0.0056	0.4007 ± 0.0107	12.1	$+1.8$	$+0.5$	$+5.7$	$+18.4$	$+0.5$	$+0.9$
Pubmed	500	0.3515 ± 0.0202	0.3511 ± 0.0106	0.1	$+1.6$	$+1.4$	$+12.9$	$+15.2$	$+0.1$	-1.7
20News	100	0.4347 ± 0.0117	0.3613 ± 0.0202	16.9	-0.3	$+7.8$	$+4.8$	$+4.0$	$+2.3$	$+0.4$
Mushroom	22	0.4575 ± 0.0011	0.2051 ± 0.0678	55.2	$+13.0$	-6.2	$+6.1$	$+2.7$	$+0.2$	-0.5
AMiner	500	0.4989 ± 0.0091	0.4093 ± 0.0124	18.0	$+0.9$	-0.2	$+1.4$	$+0.5$	$+3.6$	$+1.6$
Zoo	16	0.9054 ± 0.0190	0.8595 ± 0.0256	5.1	$+2.4$	$+2.4$	$+11.5$	$+9.0$	$+10.7$	$+5.6$

Component contributions under missing attributes. To examine which components remain useful as attributes become incomplete, Figure 10 plots the NMI difference between HSENet and each ablated variant. Table 6 gives the corresponding gains at $r = 0$ and $r = 0.6$.

Hyperedge filling provides the largest NMI gains among the three components on Cora, Citeseer, and Pubmed at $r = 0.6$: 14.3, 18.4, and 15.2 percentage points, respectively. On Citeseer, the gain increases steadily across the nonzero masking ratios, from 5.7 points at $r = 0.1$ to 18.4 points at $r = 0.6$. The corresponding w/o Filling curves also deteriorate substantially in ARI, ACC, and F1. These observations support the value of hyperedge membership supervision when vertex attributes are incomplete. The benefit is dataset dependent: it remains substantial on Zoo (9.0 points at $r = 0.6$), but is small on AMiner (0.5 points).

Masked feature reconstruction is particularly beneficial on 20News under nonzero masking, with NMI gains of 4.3–7.8 percentage points across the sweep. It also yields positive mean NMI gains at every tested nonzero ratio on Cora, Citeseer, and Pubmed, while its contribution on AMiner is small and changes sign. On Mushroom, the mean gain becomes negative from $r = 0.2$ onward. However, the large run-to-run variability in Figure 9 limits conclusions about individual reversals; the gain curves alone do not establish statistical significance.

EDVW topology enhancement retains positive NMI gains throughout the sweep on Zoo and AMiner, reaching 5.6 and 1.6 percentage points at $r = 0.6$, respectively. Its effect is less uniform elsewhere: w/o Enhan. attains higher mean NMI on Pubmed for $r \geq 0.2$. Moreover, AMiner illustrates a metric tradeoff, as topology enhancement improves NMI, ARI, and ACC while w/o Enhan. has higher F1. Taken together, these results show that the components provide complementary, dataset-dependent benefits under missing attributes, with hyperedge filling contributing most strongly on the three citation datasets.

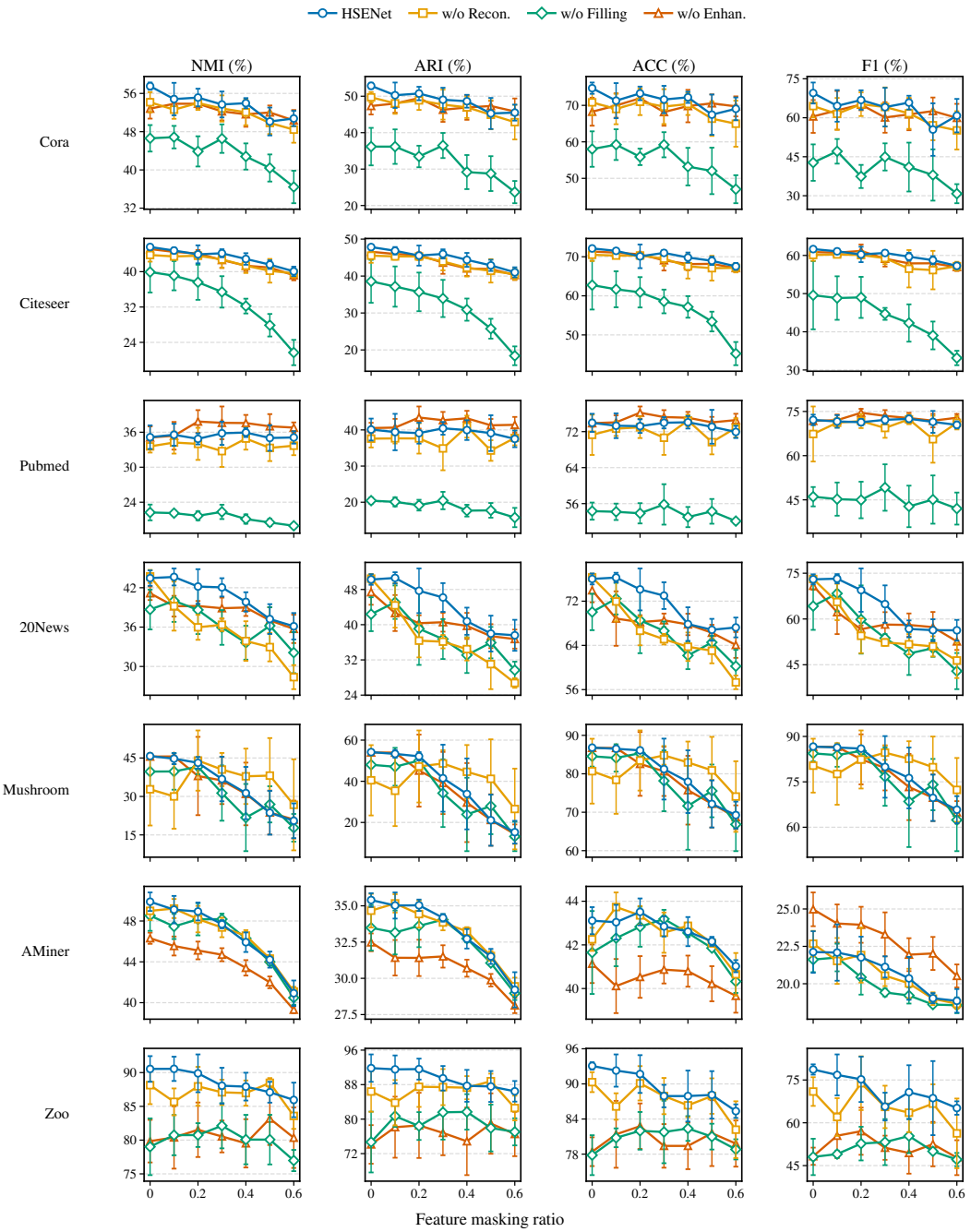


Figure 9: Robustness to attribute masking. Rows correspond to datasets and columns to NMI, ARI, ACC, and F1, displayed as percentages; error bars show standard deviations over 5 runs.

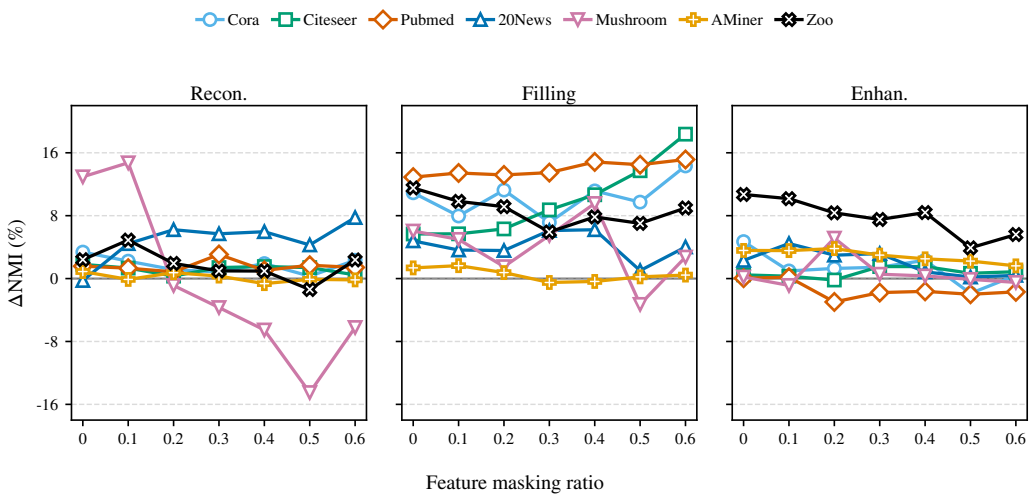


Figure 10: Component-wise NMI gains under attribute masking: full HSENet minus the variant without Recon., Filling, or Enhan. Values are in percentage points, so positive values favor the full model.