GEOMETRY-AWARE GENERATIVE MODELING FOR GRAPH CLUSTERING VIA HYPERSPHERICAL DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised graph clustering is fundamental for uncovering latent structures in graph-structured data, particularly in scenarios where labeled data is limited or unavailable. However, existing approaches often struggle to simultaneously achieve cluster-discriminative representations and geometric consistency. Conventional variational graph autoencoders rely on unimodal Gaussian priors in Euclidean space, often leading to overlapping latent clusters, while contrastive approaches depend on heuristic augmentations that may disrupt essential structural information. To overcome these limitations, we propose Hyperspherical Contrastive Diffusion (HCD), a novel unsupervised graph clustering framework that jointly leverages hyperspherical geometry and diffusion-based generative modeling. HCD constrains node embeddings to lie on a unit hypersphere and refines them via a multi-step temporal denoising diffusion process. It integrates a Product-of-Experts aggregation strategy, a von Mises–Fisher KL divergence to regularize angular latent distributions, a spherical contrastive loss to enforce discriminative alignment, and a cluster compactness-separation regularizer based on Student-t assignments and entropy minimization. These objectives collectively shape a latent space that preserves graph structure while promoting tight intra-cluster cohesion and clear inter-cluster separation. Comprehensive experiments across diverse benchmarks and multiple clinically and biologically significant real-world tissue clustering scenarios (ranging from complex neuroanatomical region identification to cancer tissue segmentation under varied conditions) demonstrate that HCD consistently achieves state-of-the-art performance in clustering accuracy, robustness, and stability.

Source Code — https://anonymous.4open.science/r/HCD-F475

1 INTRODUCTION

Unsupervised graph clustering has become a fundamental technique for uncovering latent structures in graph-structured data, such as citation networks, social graphs, and biological systems Kipf & Welling (2016); Long et al. (2023). Unlike supervised learning methods, which require expensive and often unavailable annotations, unsupervised clustering enables automatic grouping of nodes based solely on topological and attribute information, making it especially valuable in large-scale or label-scarce domains Wu et al. (2021). Furthermore, accurate graph clustering serves as a crucial pretext task that enhances performance on downstream applications such as semi-supervised classification, recommendation systems, and link prediction You et al. (2020).

Despite recent advances, current unsupervised graph clustering methods still face key challenges in learning representations that are both *cluster-friendly* and *geometrically consistent*. Classical (variational) graph autoencoders often assume a unimodal isotropic Gaussian prior in a Euclidean latent space, which can blur the separation between semantically distinct communities by collapsing them into overlapping latent regions Kipf & Welling (2016); Zhang et al. (2022a); Chen et al. (2024); Mrabah et al. (2024). On the other hand, contrastive approaches attempt to improve discriminability through random graph augmentations, such as edge dropping or node masking, but these often yield unstable signals by unintentionally destroying informative structural cues Veličković et al. (2019);

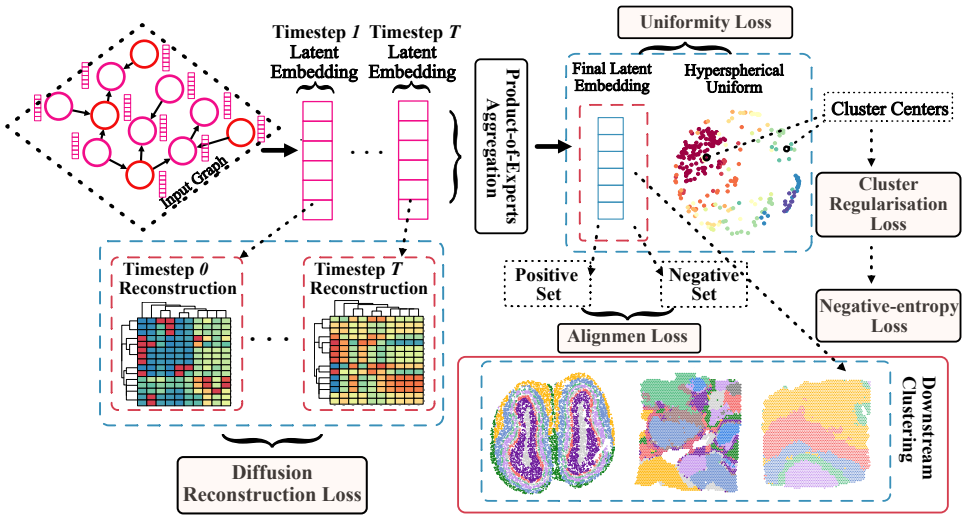

Figure 1: Overview of the HCD framework. The input graph representation is encoded via a T-step denoising diffusion process, producing timestep-specific embeddings. These embeddings are fused into a hyperspherical latent representation using a learnable PoE mechanism. Training is guided by complementary objectives: (1) *Diffusion reconstruction loss* (bottom left), which reconstructs the input adjacency matrix from latent embeddings; (2) *Uniformity loss* (top right), a vMF KL divergence that aligns the vMF posteriors with a uniform hyperspherical prior by penalizing excessive concentration and preventing representation collapse; (3) *Spherical contrastive alignment loss* (center), which pulls positive pairs closer while pushing negatives beyond a margin in angular space; (4) *Cluster regularization and negative entropy loss* (right), which tighten Student-t assignment distributions around learnable cluster centroids and prevent degeneracy.

Hassani & Khasahmadi (2020); Zhu et al. (2021); Deng et al. (2025). Moreover, deep graph encoders frequently suffer from the *over-smoothing problem*, where repeated message passing leads to homogenized node embeddings and indistinct inter-cluster boundaries Zhu & Koniusz (2020). Although prior works have incorporated adversarial learning or adaptive convolutions to boost representation quality Zhang et al. (2019); Pan et al. (2019); Wang et al. (2019), relatively few explicitly enforce the desirable clustering properties of high intra-cluster compactness and low inter-cluster overlap. Crucially, most fail to uncover a latent space that aligns with the intrinsic data geometry and reveals structure without relying on heuristics.

These limitations point to two critical gaps: (i) the lack of a latent geometry that naturally encourages separation and remains consistent in high dimensions, and (ii) the absence of principled objectives that sculpt a clustering-friendly latent space without relying on heuristic perturbations. To close these gaps, we propose a novel framework that integrates hyperspherical embeddings and diffusion processes to construct semantically meaningful, geometry-aware representations tailored for clustering.

In this work, we propose *Hyperspherical Contrastive Diffusion (HCD)*, a novel unsupervised graph clustering framework that addresses the above challenges by integrating geometry-aware embedding constraints with temporally-aware generative modeling. Specifically, HCD constrains graph representation to lie on the unit hypersphere and refines them through a multi-step denoising diffusion process. This design synergistically leverages the strengths of both generative modeling and hyperspherical geometry to produce robust and well-separated latent clusters. As illustrated in Figure 1, the key components of our approach include:

- A **temporal variational diffusion encoder**, which performs T iterative denoising steps and aggregates latent representations using a Product-of-Experts (PoE) mechanism, ensuring stability and robustness across varying noise levels;
- A **von Mises–Fisher (vMF) KL regularizer**, which aligns the posterior distribution with a uniform prior on the hypersphere, mitigating representation collapse (uniformity loss);

- A **spherical contrastive alignment loss**, which explicitly optimizes pairwise angular distances by pulling positive pairs closer while pushing negative pairs beyond a predefined angular margin;
- A **cluster compactness and separation regularizer**, built upon using Student-t assignments and normalized cluster centers, combined with an entropy penalty to prevent degeneracy and promote clear partitioning.

These objectives, together with the diffusion reconstruction loss, cooperatively shape a latent space that is both structure-preserving and cluster-discriminative. Compared to existing methods, HCD offers several key advantages: 1) The use of angular metrics and vMF regularization facilitates more interpretable and well-separated embeddings in high-dimensional spaces; 2) Diffusion-based encoding and spherical contrastive losses enable structure-aware training without relying on heuristic graph augmentations; 3) Temporal aggregation and entropy regularization improve stability across noisy graphs and enhance performance in low-label or zero-label regimes; 4) The joint optimization of geometric, contrastive, and clustering objectives within a single diffusion framework leads to coherent representation learning and avoids the limitations of conventional two-stage training pipelines. In summary, our main contributions are as follows:

- We propose the first unsupervised diffusion-driven graph clustering framework on the hypersphere, seamlessly combining denoising diffusion with contrastive learning and geometric regularization.
- We introduce a novel integration of spherical KL divergence, contrastive angular alignment, and cluster compactness/separation objectives to directly shape a cluster-friendly latent space.
- We conduct extensive evaluations across both canonical graph benchmarks and clinically realistic spatial-omics datasets (spanning neuroanatomical tissue mapping and heterogeneous cancer microenvironments) demonstrating that HCD consistently achieves state-of-the-art clustering performance under real-world conditions.

2 Related Work

Graph clustering has been extensively studied through both classical spectral methods and modern graph neural networks (GNNs). Spectral approaches typically formulate community detection as a graph cut minimization problem, such as RatioCut or Normalized Cut (Ncut), but suffer from poor scalability on large graphs Von Luxburg (2007); Zhao et al. (2023). To enhance efficiency, representation learning-based methods decouple node embedding and clustering. Early examples include DeepWalk Perozzi et al. (2014), GraphEncoder Tian et al. (2014), DNGR Cao et al. (2016), and adversarial variants such as GraphGAN Wang et al. (2018a) and GraphSGAN Ding et al. (2018). However, these methods often neglect node attribute information, limiting their effectiveness on attributed graphs.

Graph autoencoders (GAEs), especially those incorporating node features, have emerged to jointly capture graph structure and node attributes. Representative models include GAE and its variational counterpart VGAE Kipf & Welling (2016). Subsequent advancements introduce adversarial regularization Pan et al. (2018; 2019), kernel-based refinements Zhang et al. (2019; 2022a), mixture priors Hui et al. (2020), and attention-guided clustering objectives Wang et al. (2019). More recent pipelines—such as DAEGC Wang et al. (2019), SDCN Bo et al. (2020), DFCN Tu et al. (2021), and DCRN Liu et al. (2022)—achieve strong empirical performance but remain sensitive to the quality of the input adjacency matrix. To address this sensitivity, adaptive GAEs have been proposed to learn graph structures directly from data Li et al. (2021b); Zhang et al. (2022b). In parallel, contrastive learning has emerged as a powerful paradigm for unsupervised clustering, enhancing representation separability without relying on label supervision Chen et al. (2024); Mrabah et al. (2024); Deng et al. (2025).

In summary, state-of-the-art methods aim to integrate both topological and attribute information, while employing explicit learning objectives (such as reconstruction, attention, or contrastive loss) that promote cluster-discriminative and geometrically consistent representations.

3 PROPOSED METHOD

We consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges, respectively. The graph structure is represented by a normalized adjacency matrix $\mathcal{A} \in \mathbb{R}^{N\times N}$, and node attributes are encoded in a feature matrix $\mathcal{X} \in \mathbb{R}^{N\times F}$, where N is the number of nodes and F is the dimensionality of each node's feature vector.

To learn temporally-aware latent representations, we develop a diffusion-based encoder that operates over T discrete time steps. At each step t, the encoder produces a latent Gaussian distribution parameterized by a mean and log-variance:

$$\mathcal{H}^{(t)} = \phi^{(t)}(\mathcal{X}, \mathcal{A}), \quad \boldsymbol{\mu}^{(t)} = \phi_\mu(\mathcal{H}^{(t)}, \mathcal{A}), \quad \log \boldsymbol{\sigma}^{2\,(t)} = \phi_\sigma(\mathcal{H}^{(t)}, \mathcal{A}), \tag{1}$$

where $\phi^{(t)}, \phi_\mu, \phi_\sigma$ are graph convolution layers Kipf & Welling (2017), and $\mathcal{H}^{(t)}$ denotes the intermediate node features at time t. A latent sample $\boldsymbol{z}^{(t)}$ is drawn using the reparameterization trick:

$$\boldsymbol{z}^{(t)} = \boldsymbol{\mu}^{(t)} + \exp\big(\tfrac{1}{2}\log \boldsymbol{\sigma}^{2\,(t)}\big) \odot \boldsymbol{\epsilon}^{(t)}, \boldsymbol{\epsilon}^{(t)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d), \tag{2}$$

where d is the latent dimensionality and $\odot$ denotes the element-wise (Hadamard) product.

To integrate information across time steps, we employ a Product-of-Experts (PoE) aggregation mechanism with learnable temporal attention weights. Specifically, we define:

$$w_t = \frac{\exp(\gamma_t)}{\sum_{s=1}^{T}\exp(\gamma_s)}, \quad \boldsymbol{P}^{(t)} = \exp\big(-\log \boldsymbol{\sigma}^{2\,(t)}\big), \quad \bar{\boldsymbol{\mu}} = \frac{\sum_{t=1}^{T} w_t\, \boldsymbol{P}^{(t)} \odot \boldsymbol{\mu}^{(t)}}{\sum_{t=1}^{T} w_t\, \boldsymbol{P}^{(t)}} \in \mathbb{R}^{N\times d}, \tag{3}$$

where $\gamma_t \in \mathbb{R}$ are learnable scalar parameters for temporal weighting, and $\boldsymbol{P}^{(t)}$ represents the element-wise precision (i.e., inverse variance) of the t-th Gaussian. The PoE formulation enables closed-form inverse-variance fusion under Gaussian assumptions. A detailed derivation is provided in Appendix Section E: "Why Product-of-Experts Helps: Inverse-Variance Fusion."

Finally, to impose hyperspherical geometry on the latent space, the aggregated embeddings are projected onto the unit hypersphere via row-wise ℓ_2-normalization: $\hat{\boldsymbol{Z}} = \mathcal{N}(\bar{\boldsymbol{\mu}})$, where $\mathcal{N}(\cdot)$ denotes row-wise normalization such that each row of $\hat{\boldsymbol{Z}}$ has unit norm.

3.1 LOSS FUNCTIONS AND TRAINING OBJECTIVE

(1) Reconstruction via Temporal Variational Diffusion Decoder. Let $\{\boldsymbol{z}^{(t)}\}_{t=1}^{T}$ denote the sampled latent trajectories across T diffusion steps. A linear β-schedule defines the noise levels $\{\beta_t\}_{t=1}^{T}$, and the cumulative noise scaling terms $\sqrt{1-\bar{\alpha}_t}$ are computed with $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$. Let $c_t = \sqrt{1-\bar{\alpha}_t}$. At each timestep t, we compute cosine similarity matrices:

$$\boldsymbol{S}^{(t)} = \mathcal{N}(\boldsymbol{z}^{(t)})\,\mathcal{N}(\boldsymbol{z}^{(t)})^\top, \qquad \hat{\boldsymbol{A}} = \frac{\sum_{t=1}^{T}(c_t\, \boldsymbol{S}^{(t)})}{\sum_{t=1}^{T} c_t} \in [0,1]^{N\times N}, \tag{4}$$

where $\mathcal{N}(\cdot)$ denotes row-wise ℓ_2-normalization. The reconstruction loss is computed as a weighted binary cross-entropy against the normalized adjacency matrix $\mathcal{A}$:

$$\mathcal{L}_{\text{rec}} = \eta\ \mathrm{BCE}\big(\mathrm{vec}(\hat{\boldsymbol{A}}), \mathrm{vec}(\mathcal{A})\big), \tag{5}$$

where $\mathcal{L}_{\text{rec}}$ is the diffusion reconstruction loss, η is a global normalization scalar, and $\mathrm{vec}(\cdot)$ vectorizes the input matrix, $\mathrm{BCE}(\cdot)$ denotes the binary cross-entropy loss. As detailed in Appendix F, the decoder in Eq. (4) is not an ad-hoc aggregation, but the closed-form solution of a noise-free diffusion-style refinement chain $\hat{\boldsymbol{A}}_T \to \hat{\boldsymbol{A}}_{T-1} \to \cdots \to \hat{\boldsymbol{A}}_0$ on the reconstructed adjacency matrix.

(2) KL Divergence on the Hypersphere. Each row of $\hat{\boldsymbol{Z}} \in \mathbb{R}^{N\times d}$ defines the mean direction of a vMF distribution:

$$q(x;\, \mu, \kappa) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{d/2} I_{\frac{d}{2}-1}(\kappa)}\ \exp\big(\kappa\, \mu^\top x\big), \tag{6}$$

where $\mu \in \mathbb{R}^d$ is a unit-norm mean direction, $\kappa = \mathrm{softplus}(\lambda)$ is the concentration parameter with learnable λ, $\mathrm{softplus}(x) = \log\big(1+e^x\big)$, and $I_\nu(\cdot)$ is the modified Bessel function of the first kind.

The prior is the uniform distribution over the hypersphere, $\mathcal{U}(\mathbb{S}^{d-1})$. The corresponding uniformity loss is:

$$\mathcal{L}_{\text{UNI}} = \frac{1}{N}\sum_{i=1}^{N} \text{KL}\big(q_i \,\|\, p\big). \tag{7}$$

Under a uniform prior, it discourages overly peaky vMF posteriors and prevents representation collapse on $\mathbb{S}^{d-1}$.

(3) Spherical Contrastive Alignment Loss. Let $\hat{\boldsymbol{Z}} = [\hat{\boldsymbol{z}}_1, \ldots, \hat{\boldsymbol{z}}_N]^\top \in \mathbb{R}^{N\times d}$. We define the positive edge set as: $\mathcal{E}^+ = \big\{(i,j) \mid \mathcal{A}_{ij} = 1\big\}$, and construct a random negative edge set: $\mathcal{E}^- \subset \big\{(i,j) \mid \mathcal{A}_{ij} = 0\big\}, |\mathcal{E}^-| = |\mathcal{E}^+|\, n_{\text{neg}}$, where $n_{\text{neg}} \in \mathbb{N}$ is a hyperparameter specifying the number of negatives per positive. Using distance power $\alpha > 0$ and margin $m > 0$, we define the alignment loss:

$$\mathcal{L}_{\text{aln}} = \frac{1}{|\mathcal{E}^+|}\sum_{(i,j)\in\mathcal{E}^+} \|\hat{\boldsymbol{z}}_i - \hat{\boldsymbol{z}}_j\|_2^{\alpha} + \frac{1}{|\mathcal{E}^-|}\sum_{(i,j)\in\mathcal{E}^-} \Big[\max\big(0,\, m - \|\hat{\boldsymbol{z}}_i - \hat{\boldsymbol{z}}_j\|_2\big)\Big]^{\alpha}. \tag{8}$$

(4) Cluster Compactness and Separation Regularization. Let $\boldsymbol{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_K]^\top \in \mathbb{R}^{K\times d}$ be the matrix of learnable cluster centroids. We compute soft cluster assignments using a Student-t kernel:

$$p_{ik} = \frac{\big(1 + \|\hat{\boldsymbol{z}}_i - \boldsymbol{c}_k\|_2^2/\nu\big)^{-\frac{\nu+1}{2}}}{\sum_{r=1}^{K}\big(1 + \|\hat{\boldsymbol{z}}_i - \boldsymbol{c}_r\|_2^2/\nu\big)^{-\frac{\nu+1}{2}}}, k = 1, \ldots, K, \tag{9}$$

where $\nu > 0$ controls the shape of the distribution. Each vector $p_i = [p_{i1}, \ldots, p_{iK}]$ is a probability distribution over the K clusters. The compactness and separation terms are defined as:

$$\text{Intra} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} p_{ik}\, \|\hat{\boldsymbol{z}}_i - \boldsymbol{c}_k\|_2^2, \qquad \text{Inter} = \frac{2}{K(K-1)}\sum_{1\le k<\ell\le K} \|\boldsymbol{c}_k - \boldsymbol{c}_\ell\|_2, \tag{10}$$

with the overall cluster regularization loss defined as: $\mathcal{L}_{\text{clu}} = \frac{\text{Intra}}{\text{Inter}+10^{-9}}$.

(5) Entropy Regularization on Assignments. To avoid degenerate assignments (e.g., all nodes collapsing to a single cluster), we include a negative entropy term on the soft assignment matrix $p \in \mathbb{R}^{N\times K}$:

$$\mathcal{L}_{\text{ent}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} p_{ik}\, \log\big(p_{ik} + 10^{-9}\big). \tag{11}$$

(6)Final Training Objective. The overall training objective is a weighted sum of all loss terms:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{UNI}}\mathcal{L}_{\text{UNI}} + \lambda_{\text{aln}}\mathcal{L}_{\text{aln}} + \lambda_{\text{clu}}\mathcal{L}_{\text{clu}} + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}}, \tag{12}$$

where $\lambda_{\text{UNI}}, \lambda_{\text{aln}}, \lambda_{\text{clu}}, \lambda_{\text{ent}}$ are hyperparameters balancing the contributions of each term. Complete hyperparameter settings are listed in Appendix Section B. In addition, we provide a geometric guarantee showing that the spherical contrastive margin enforces a non-trivial minimum inter-cluster angular separation. Detailed derivations can be found in Appendix Section D: Geometric Guarantee: Angular Margin Implies Separation and Appendix Section F: Spherical Similarity Calibration.

The training procedure of our model is summarized in Algorithm 1. Model parameters are optimized using either Adam or SGD, with separate learning rates: lr for all network parameters and lr_κ for the vMF concentration parameters. After each training epoch, we fit a vMF mixture model with K components via Expectation–Maximization on the final embeddings $\hat{\boldsymbol{Z}}$, yielding the predicted cluster labels $\hat{\boldsymbol{y}}$, following the procedure of Taghia et al. (2014); Luo et al. (2025); Li et al. (2025).

Computational Complexity. A detailed analysis of the computational and memory complexity, along with empirical wall-clock runtime benchmarks and accuracy–efficiency trade-offs, is provided in Appendix Section H

Algorithm 1 HYPERSPHERICAL CONTRASTIVE DIFFUSION (HCD)

INPUT: Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, feature matrix $\mathcal{X} \in \mathbb{R}^{N \times F}$, normalized adjacency $\mathcal{A} \in \mathbb{R}^{N \times N}$, number of clusters K, diffusion depth T, loss weights $\lambda_{\text{UNI}}, \lambda_{\text{aln}}, \lambda_{\text{clu}}, \lambda_{\text{ent}}$
INITIALISE: Encoder parameters $\{\phi^{(t)}, \phi_\mu, \phi_\sigma\}_{t=1}^{T}$, Product-of-Experts weights $\{\gamma_t\}_1^T$, cluster centres $\boldsymbol{C} \in \mathbb{R}^{K \times d}$ (*row-wise ℓ_2-normalised*), vMF concentration parameter λ, optimiser ADAM
for epoch $= 1$ **to** $E_{\max}$ **do**
  for $t = 1$ **to** T **do**
    $\mathcal{H}^{(t)} \leftarrow \phi^{(t)}(\mathcal{X}, \mathcal{A})$
    $\boldsymbol{\mu}^{(t)} \leftarrow \phi_\mu(\mathcal{H}^{(t)}, \mathcal{A}), \quad \log \boldsymbol{\sigma}^{2\,(t)} \leftarrow \phi_\sigma(\mathcal{H}^{(t)}, \mathcal{A})$
    Sample $\boldsymbol{z}^{(t)} \sim \mathcal{N}\big(\boldsymbol{\mu}^{(t)}, \operatorname{diag}(\boldsymbol{\sigma}^{2\,(t)})\big)$
  end for
  PoE aggregation: compute element-wise precisions $\boldsymbol{P}^{(t)} = \exp(-\log \boldsymbol{\sigma}^{2\,(t)})$; soft weights $w_t = \exp(\gamma_t) / \sum_s \exp(\gamma_s)$
  $\bar{\boldsymbol{\mu}} = \dfrac{\sum_{t=1}^{T} w_t \, \boldsymbol{P}^{(t)} \odot \boldsymbol{\mu}^{(t)}}{\sum_{t=1}^{T} w_t \, \boldsymbol{P}^{(t)}}$
  Normalize $\hat{\boldsymbol{Z}} = \mathcal{N}\big(\bar{\boldsymbol{\mu}}\big)$ // node embeddings on $\mathbb{S}^{d-1}$
  Diffusion decoder: reconstruct $\hat{\mathcal{A}}$ using cosine-sim. trajectory and compute $\mathcal{L}_{\text{rec}}$
  Uniformity loss: $\mathcal{L}_{\text{UNI}} = \frac{1}{N} \sum_i \mathrm{KL}\big(q_i \| \mathcal{U}(\mathbb{S}^{d-1})\big)$ where q_i is vMF$(\hat{\boldsymbol{z}}_i, \kappa)$, $\kappa = \operatorname{softplus}(\lambda)$
  Contrastive alignment: sample negatives $\mathcal{E}^-$, compute $\mathcal{L}_{\text{aln}}$
  Cluster assignments: $p_{ik} \leftarrow \text{Student-t}(\hat{\boldsymbol{z}}_i, \boldsymbol{c}_k)$
  Compactness/separation: compute $\mathcal{L}_{\text{clu}}$ using intra/inter distances; **Entropy:** $\mathcal{L}_{\text{ent}} = \frac{1}{N} \sum_{i,k} p_{ik} \log(p_{ik} + 10^{-9})$
  Total loss: $\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{UNI}} \mathcal{L}_{\text{UNI}} + \lambda_{\text{aln}} \mathcal{L}_{\text{aln}} + \lambda_{\text{clu}} \mathcal{L}_{\text{clu}} + \lambda_{\text{ent}} \mathcal{L}_{\text{ent}}$
  Update all parameters via back-propagation on $\mathcal{L}$
end for
Clustering: fit a K-component vMF mixture on $\hat{\boldsymbol{Z}}$ and assign labels $\hat{\boldsymbol{y}}$
OUTPUT: cluster labels $\hat{\boldsymbol{y}}$, hyperspherical embeddings $\hat{\boldsymbol{Z}}$

4 EXPERIMENTS

We evaluate our method in two domains: (i) standard graph benchmarks and (ii) clinically relevant spatial transcriptomics datasets, enabling rigorous benchmarking and and real-world validation.

4.1 DATASETS

Large-scale graphs. We first consider two large-scale node classification datasets from the Open Graph Benchmark (OGB) Hu et al. (2020): **ogbn-arxiv** (a citation network of 169K nodes and 1.1M edges) and **ogbn-products** (an Amazon co-purchase network with 2.4M nodes and 123M edges).

Medium-scale graphs. We additionally evaluate on three widely used medium-sized benchmarks: **ACM** and **DBLP**, both representing co-authorship networks Bo et al. (2020); and **Wiki**, a web page linkage graph Yang et al. (2015).

Clinically relevant spatial transcriptomics datasets. To assess applicability in biomedical domains, we test on three real-world tissue datasets: (i) **STARmap (mouse visual cortex)**: 1,207 cells profiled by in situ sequencing across 1,020 genes Wang et al. (2018b); (ii) **DLPFC (human prefrontal cortex)**: section 151672 from 10x Visium, with 3,888 spatial capture spots spanning six cortical layers Maynard et al. (2021); (iii) **BRCA (human breast carcinoma)** includes 3,798 capture spots from tumour and stromal regions Polyak et al. (2011): 3,798 capture spots from tumor and stromal regions, measured with 10x Visium Polyak et al. (2011). Each spot in the spatial transcriptomics is annotated with Cartesian spatial coordinates and raw gene expression profiles.

4.2 BASELINE METHODS

Graph Domain Baselines. We compare against a comprehensive set of baselines, including: *Traditional Methods*: K-means and METIS Karypis & Kumar (1998). *Embedding & Self-Supervised Methods*: Node2Vec Grover & Leskovec (2016), DGI Veličković et al. (2019), S^3GC Devvrit et al. (2022), BGRL Thakoor et al. (2022), DMoN Tsitsulin et al. (2023), CVGAE Mrabah et al. (2024),

Table 1: Clustering performance (mean ± standard error) on ogbn-arxiv and ogbn-products datasets.

Method	ogbn-arxiv ($\lvert\mathcal{V}\rvert$ = 169,343, $\lvert\mathcal{E}\rvert$ = 1,166,243)			ogbn-products ($\lvert\mathcal{V}\rvert$ = 1,939,743, $\lvert\mathcal{E}\rvert$ = 21,111,007)		
	ACC(↑)	NMI(↑)	ARI(↑)	ACC(↑)	NMI(↑)	ARI(↑)
K-means	17.6±0.24	21.6±0.38	7.4±0.18	20.0±0.18	27.3±0.10	8.2±0.40
MinCutPool	24.2±0.10	38.0±0.19	13.9±0.42	25.7±0.18	43.0±0.13	13.0±0.57
METIS	20.9±0.50	34.5±0.16	12.6±0.11	29.4±0.29	46.8±0.29	14.5±0.31
Node2vec	29.0±0.11	40.6±0.28	19.0±0.24	35.7±0.53	48.9±0.19	17.0±0.16
DGI	31.4±0.19	41.2±0.56	22.3±0.30	32.0±0.53	46.7±0.36	17.4±0.09
DMoN	25.0±0.20	35.6±0.48	12.7±0.31	30.4±0.29	42.8±0.58	13.9±0.60
S^3GC	35.0±0.29	46.3±0.27	27.0±0.33	40.2±0.07	53.6±0.10	23.0±0.54
BGRL	22.7±0.60	32.1±0.10	13.0±0.18	–	–	–
CVGAE	35.4±0.47	47.0±0.58	27.6±0.60	39.7±0.45	52.9±0.44	22.6±0.57
THESAURUS	34.2±0.85	45.8±0.67	26.4±0.71	39.3±0.63	53.1±0.54	22.4±0.48
Ours	**41.3**±0.62	**50.4**±0.45	**31.5**±0.43	**45.0**±0.60	**57.6**±0.36	**28.2**±0.49

Table 2: Clustering results of different VGAE-based models on three benchmarks. Best scores are in bold (mean ± standard error, expressed as percentages).

Method	DBLP			ACM			Wiki		
	ACC(↑)	NMI(↑)	ARI(↑)	ACC(↑)	NMI(↑)	ARI(↑)	ACC(↑)	NMI(↑)	ARI(↑)
Graphite	65.5±0.48	32.7±0.28	30.9±0.42	85.7±0.46	56.7±0.19	62.2±0.15	42.6±0.44	41.5±0.27	24.3±0.08
GDN-VAE	57.7±0.31	27.3±0.17	16.0±0.56	79.4±0.25	49.4±0.50	46.0±0.53	45.0±0.18	42.3±0.42	24.1±0.28
$\mathcal{N}$-VGAE	59.0±0.21	21.3±0.10	21.7±0.48	84.7±0.25	54.2±0.34	59.8±0.43	44.9±0.52	40.0±0.24	25.3±0.07
$\mathcal{D}$-VGAE	59.4±0.53	27.9±0.20	18.0±0.37	79.6±0.59	49.0±0.08	47.4±0.38	46.3±0.25	40.3±0.48	24.4±0.30
GMM-VGAE	54.6±0.59	23.5±0.12	23.9±0.55	83.1±0.16	54.0±0.08	56.3±0.30	42.7±0.34	39.5±0.50	23.6±0.43
SI-VGAE	62.0±0.57	26.3±0.46	24.1±0.17	62.0±0.29	32.6±0.57	29.1±0.56	42.3±0.40	40.1±0.42	22.3±0.13
DCGL	64.2±0.55	33.5±0.33	31.3±0.42	84.8±0.24	55.3±0.44	61.9±0.36	47.8±0.16	41.3±0.42	25.1±0.26
Ours	**76.2±0.46**	**46.0±0.15**	**51.8±0.37**	**86.7±0.28**	**60.7±0.51**	**67.4±0.22**	**54.6±0.17**	**44.7±0.48**	**31.6±0.39**

and THESAURUS Deng et al. (2025). *VGAE-Based Methods (for Table 2)*: Graphite Grover et al. (2019), GDN-VAE Li et al. (2021a), $\mathcal{N}$-VGAE and $\mathcal{D}$-VGAE Li et al. (2020), GMM-VGAE Hui et al. (2020), SI-VGAE Hasanzadeh et al. (2019), and DCGL Chen et al. (2024).

Spatial Transcriptomics Baselines. We benchmark against both traditional and GNN-based approaches designed for spatial tissue modeling: **Traditional Spatial Methods**: Scanpy Wolf et al. (2018), SpatialPCA Shang & Zhou (2022). **GNN-Based Models**: SpaGCN Hu et al. (2021), STAGATE Dong & Zhang (2022), SpaceFlow Ren et al. (2022), CCST Li et al. (2022), and GraphST Long et al. (2023).

4.3 Implementation Details

The encoder architecture consists of two graph convolutional layers followed by a T-step temporal diffusion module. All latent embeddings are ℓ_2-normalized to lie on the unit hypersphere. Training is conducted for 1,000 epochs using the Adam optimizer, with a learning rate of 5×10^{-3} and weight decay of 5×10^{-4}. The learnable vMF concentration parameters are updated separately using a learning rate of 10^{-3}. For fairness, all baselines are reproduced using the authors' official implementations and default settings. Clustering performance is evaluated using three standard metrics: Accuracy (ACC), which measures label assignment correctness; Normalized Mutual Information (NMI), which quantifies information overlap between predicted and ground-truth labels; and Adjusted Rand Index (ARI), which accounts for chance-adjusted clustering agreement. Complete hyperparameter configurations for each dataset are detailed in Appendix Section B.

4.4 Results

Open Graph Benchmarks. Table 1 reports clustering performance (ACC, NMI, and ARI) on OGBN-ARXIV and OGBN-PRODUCTS. Our method, HCD, achieves the best performance across all metrics on both datasets. On OGBN-ARXIV, HCD surpasses the strongest baseline (CVGAE) by +5.9 ACC points (41.3 vs. 35.4), +3.4 NMI points (50.4 vs. 47.0), and +3.9 ARI points (31.5 vs. 27.6). On OGBN-PRODUCTS, HCD improves over the best baseline (S^3GC) by +4.8 ACC points (45.0 vs. 40.2), +4.0 NMI points (57.6 vs. 53.6), and +5.2 ARI points (28.2 vs. 23.0).

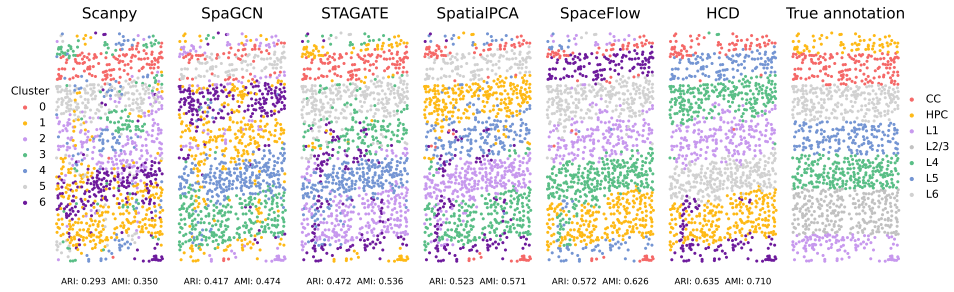

Figure 2: Murine visual cortex slice profiled by STARMAP, annotated with expert cell type labels.

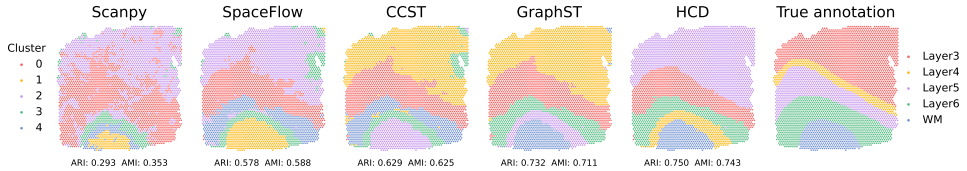

Figure 3: Human dorsolateral prefrontal cortex (DLPFC) section from 10x Visium (specimen #151672), visualized with clustering assignments from each competing method.

These improvements are attributed to three key factors: (i) the hyperspherical latent space, which enhances angular separability at scale; (ii) the temporal diffusion encoder, which effectively mitigates over-smoothing; and (iii) the integration of uniformity, contrastive, and cluster regularization losses, which stabilize training without relying on heuristic graph augmentations.

Canonical Graph Benchmarks. Table 2 presents results on widely-used canonical graphs: ACM, DBLP, and WIKI. HCD outperforms all VGAE-based baselines. On the high-degree DBLP graph, HCD achieves an ACC of 0.762, improving significantly over Graphite (0.655), with corresponding gains of approximately 13% in NMI and 21% in ARI. On the structurally saturated ACM graph, HCD yields consistent improvements across all metrics. For the highly imbalanced WIKI graph, it obtains the highest ARI (0.316) and NMI (0.447), demonstrating robustness to skewed community sizes and hub-dominated centrality. These results highlight the effectiveness of the hyperspherical prior and the cluster compactness-separation regularizer in handling heterogeneous topologies.

Real-World Spatial Transcriptomics Benchmarks. Figures 2, 3, and 4 present evaluations on spatial transcriptomics datasets: STARmap, DLPFC, and BRCA. Across all three datasets, HCD achieves the highest ARI and AMI scores, with particularly notable margins on complex tissue structures. For instance, HCD surpasses the best competing method by **+6.3** ARI points on the STARMAP mouse cortex, and by **+1.8** ARI points on the heterogeneous DLPFC sample. These results underscore the model's ability to preserve spatial continuity and capture biological layering, facilitated by diffusion-based refinement combined with hyperspherical embeddings.

Qualitative analyses further confirm the biological plausibility of the inferred clusters. In the mouse visual cortex (Fig. 2), HCD accurately resolves cortical layers, clearly distinguishing layer 2/3 from layer 4 despite subtle transcriptional gradients. In the human DLPFC (Fig. 3), it delineates white matter boundaries and successfully recovers the thin layer 4 band, which is often missed by competing methods. In the BRCA carcinoma sample (Fig. 4), HCD maintains coherent tumor–stroma boundaries across multiple clustering resolutions ($k \in \{10, 15, 20\}$), illustrating the stability induced by entropy regularization and Student-t–based soft assignments. See Appendix G for additional discussion.

4.5 ABLATION OVERVIEW

Scope. We conduct a comprehensive ablation study to evaluate the individual contributions and design choices within our framework. A brief summary is provided below, with complete experimental details available in the Appendix Section A: (i) Component removal: we sequentially drop vMF uniformity, spherical contrastive alignment, cluster compactness/separation, and entropy regu-

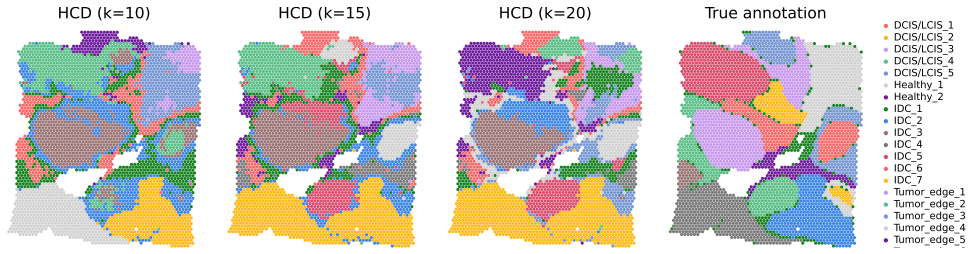

Figure 4: Breast carcinoma (BRCA) tissue measured with 10x Visium, demonstrating the impact of varying the number of target clusters on spatial segmentation.

Table 3: Ablation results on OGBN-ARXIV. Each row reports performance when one component is removed from the full model.

Variant	ACC↑	NMI↑	ARI↑
Full HCD	**41.3**±0.62	**50.4**±0.45	**31.5**±0.43
w/o vMF uniformity	39.2±0.58	48.1±0.44	29.1±0.39
w/o spherical contrast	38.6±0.55	47.6±0.47	27.8±0.41
w/o cluster regularizer	37.9±0.63	47.1±0.50	27.1±0.45
w/o entropy penalty	38.1±0.39	47.3±0.32	27.3±0.49

larization to assess their necessity. (ii) Geometry choice: we compare hyperspherical and Euclidean latent spaces while fixing the decoder to cosine similarity. (iii) Diffusion depth and aggregation: we vary the number of diffusion steps T, noise schedules, and aggregation strategies, including the PoE mechanism. (iv) Hyperparameter sensitivity: we analyze the effects of varying n_{neg}, angular margin m, distance power α, and loss weights. (v) Robustness analysis: we evaluate model stability under edge sparsification, noisy node features, as well as structural and heterogeneous noise.

Table 3 shows that each loss component contributes meaningfully on OGBN-ARXIV; removing any single term reduces both accuracy and stability. Table 4 isolates the effect of latent geometry under a fixed cosine decoder, demonstrating that hyperspherical embeddings yield cleaner partitions on both large-scale and medium-scale graphs. For the main model, we set $T = 10$ with PoE aggregation, as this configuration provides the best trade-off between accuracy and efficiency.

Table 4: Clustering performance with hyperspherical versus Euclidean embeddings.

Geometry	ogbn-arxiv			ACM		
	ACC↑	NMI↑	ARI↑	ACC↑	NMI↑	ARI↑
Hyperspherical	**41.3**±0.62	**50.4**±0.45	**31.5**±0.43	**86.7**±0.28	**60.7**±0.51	**67.4**±0.22
Euclidean	37.5±0.64	46.7±0.52	27.0±0.46	85.0±0.37	58.1±0.50	64.1±0.33

5 CONCLUSION

We introduced HCD, a novel unsupervised graph clustering framework that unifies hyperspherical embeddings, multi-step denoising diffusion, and contrastive learning within an end-to-end architecture. By jointly optimizing a vMF-based KL divergence, a spherical contrastive alignment loss, and a Student-t compactness–separation regularizer, HCD constructs a geometry-aware latent space tailored for cluster-discriminative and geometrically consistent representations. Empirical results across both standard graph benchmarks and real-world spatial transcriptomics datasets demonstrate the effectiveness and robustness of the proposed approach. Future research directions include: (i) extending HCD to dynamic or temporal graphs with evolving topologies, and (ii) generalizing the framework to multi-view scenarios, enabling joint clustering over graphs enriched with auxiliary modalities such as text or images. (iii) exploring fully hyperspherical variants of HCD where the diffusion process itself is defined on $\mathbb{S}^{d-1}$ with Riemannian (geodesic) noise. (iv) exploring alternative latent geometries such as hyperbolic or mixed-curvature manifolds for graphs with stronger hierarchical or tree-like structures; and (v) investigating how the proposed hyperspherical diffusion-based latent fusion mechanism can be adapted to non-graph data (e.g., images or tabular features).

REFERENCES

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

Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Mulin Chen, Bocheng Wang, and Xuelong Li. Deep contrastive graph learning with clustering-oriented guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11364–11372, 2024.

Bowen Deng, Tong Wang, Lele Fu, Sheng Huang, Chuan Chen, and Tao Zhang. Thesaurus: contrastive graph clustering by swapping fused gromov-wasserstein couplings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16199–16207, 2025.

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

Ming Ding, Jie Tang, and Jie Zhang. Semi-supervised learning on graphs with generative adversarial nets. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 913–922, 2018.

Kangning Dong and Shihua Zhang. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nature communications*, 13(1):1739, 2022.

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

Aditya Grover, Aaron Zweig, and Stefano Ermon. Graphite: Iterative generative modeling of graphs. In *International conference on machine learning*, pp. 2434–2444. PMLR, 2019.

Arman Hasanzadeh, Ehsan Hajiramezanali, Krishna Narayanan, Nick Duffield, Mingyuan Zhou, and Xiaoning Qian. Semi-implicit graph variational auto-encoders. *Advances in neural information processing systems*, 32, 2019.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pp. 4116–4126. PMLR, 2020.

Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J Irwin, Edward B Lee, Russell T Shinohara, and Mingyao Li. Spagcn: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature methods*, 18(11):1342–1351, 2021.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Binyuan Hui, Pengfei Zhu, and Qinghua Hu. Collaborative graph convolutional networks: Unsupervised learning meets semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4215–4222, 2020.

George Karypis and Vipin Kumar. A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. *University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN*, 38:7–1, 1998.

T. N. Kipf and M. Welling. Variational graph auto-encoders. In *NIPS Workshop on Bayesian Deep Learning*, 2016.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*, 2017.

Jia Li, Jianwei Yu, Jiajin Li, Honglei Zhang, Kangfei Zhao, Yu Rong, Hong Cheng, and Junzhou Huang. Dirichlet graph variational autoencoder. *Advances in Neural Information Processing Systems*, 33:5274–5283, 2020.

Jia Li, Jiajin Li, Yang Liu, Jianwei Yu, Yueting Li, and Hong Cheng. Deconvolutional networks on graph data. *Advances in Neural Information Processing Systems*, 34:21019–21030, 2021a.

Jiachen Li, Siheng Chen, Xiaoyong Pan, Ye Yuan, and Hong-Bin Shen. Cell clustering for spatial transcriptomics data with graph neural networks. *Nature Computational Science*, 2(6):399–408, 2022.

Xuelong Li, Hongyuan Zhang, and Rui Zhang. Adaptive graph auto-encoder for general data clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9725–9732, 2021b.

Zhixiang Li, Zhiwen Luo, Nizar Bouguila, Weifeng Su, and Wentao Fan. Disentangled representation learning for multi-view clustering via von mises-fisher hyperspherical embedding. *Neural Networks*, pp. 107802, 2025.

Yue Liu, Wenxuan Tu, Sihang Zhou, Xinwang Liu, Linxuan Song, Xihong Yang, and En Zhu. Deep graph clustering via dual correlation reduction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 7603–7611, 2022.

Yahui Long, Kok Siong Ang, Mengwei Li, Kian Long Kelvin Chong, Raman Sethi, Chengwei Zhong, Hang Xu, Zhiwei Ong, Karishma Sachaphibulkij, Ao Chen, et al. Spatially informed clustering, integration, and deconvolution of spatial transcriptomics with graphst. *Nature Communications*, 14(1):1155, 2023.

Zhiwen Luo, Wentao Fan, Manar Amayri, and Nizar Bouguila. Dynamic deep clustering of high-dimensional directional data via hyperspherical embeddings with bayesian nonparametric mixtures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025*, pp. 938–949. ACM, 2025.

Kristen R Maynard, Leonardo Collado-Torres, Lukas M Weber, Cedric Uytingco, Brianna K Barry, Stephen R Williams, Joseph L Catallini, Matthew N Tran, Zachary Besich, Madhavi Tippani, et al. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature neuroscience*, 24(3):425–436, 2021.

Nairouz Mrabah, Mohamed Bouguessa, and Riadh Ksantini. A contrastive variational graph auto-encoder for node clustering. *Pattern recognition*, 149:110209, 2024.

Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 2609–2615, 2018.

Shirui Pan, Ruiqi Hu, Sai-fu Fung, Guodong Long, Jing Jiang, and Chengqi Zhang. Learning graph embedding with adversarial training methods. *IEEE transactions on cybernetics*, 50(6):2475–2487, 2019.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Kornelia Polyak et al. Heterogeneity in breast cancer. *The Journal of clinical investigation*, 121(10): 3786–3788, 2011.

Honglei Ren, Benjamin L Walker, Zixuan Cang, and Qing Nie. Identifying multicellular spatiotemporal organization of cells with spaceflow. *Nature communications*, 13(1):4076, 2022.

Lulu Shang and Xiang Zhou. Spatially aware dimension reduction for spatial transcriptomics. *Nature communications*, 13(1):7203, 2022.

Jalil Taghia, Zhanyu Ma, and Arne Leijon. Bayesian estimation of the von-mises fisher mixture model with variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 36(9):1701–1715, 2014.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L. Dyer, Rémi Munos, Petar Velickovic, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. In *The Tenth International Conference on Learning Representations, ICLR*, 2022.

Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 24(127):1–21, 2023.

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

Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. Deep graph infomax. In *ICLR*, 2019.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

C Wang, S Pan, R Hu, G Long, J Jiang, and C Zhang. Attributed graph clustering: A deep attentional embedding approach. In *International Joint Conference on Artificial Intelligence*, 2019.

Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018a.

Xiao Wang, William E Allen, Matthew A Wright, Emily L Sylwestrak, Nikolay Samusik, Sam Vesuna, Kathryn Evans, Cindy Liu, Charu Ramakrishnan, Jia Liu, et al. Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science*, 361(6400):eaat5691, 2018b.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, 2021. doi: 10.1109/TNNLS.2020.2978386.

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *IJCAI*, volume 2015, pp. 2111–2117, 2015.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823, 2020.

Hongyuan Zhang, Pei Li, Rui Zhang, and Xuelong Li. Embedding graph auto-encoder for graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):9352–9362, 2022a.

Hongyuan Zhang, Jiankun Shi, Rui Zhang, and Xuelong Li. Non-graph data clustering via $o(n)$ bipartite graph convolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (7):8729–8742, 2022b.

Xiaotong Zhang, Han Liu, Qimai Li, and Xiao-Ming Wu. Attributed graph clustering via adaptive graph convolution. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 4327–4333, 2019.

Mingyu Zhao, Weidong Yang, and Feiping Nie. Deep multi-view spectral clustering via ensemble. *Pattern Recognition*, 144:109836, 2023.

Hao Zhu and Piotr Koniusz. Simple spectral graph convolution. In *International conference on learning representations*, 2020.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the web conference 2021*, pp. 2069–2080, 2021.

APPENDIX AND SUPPLEMENTARY MATERIAL

LLM USAGE

We disclose the use of large language models (LLMs) in accordance with the ICLR 2026 Author Guide and policy.

Models and tools. We used ChatGPT (GPT-5), 2025-08 build.

Purpose and scope. LLMs were used for text polishing of grammar and wording.

Human oversight. All LLM outputs were checked and, when necessary, rewritten by the authors. The authors take full responsibility for all content.

A ABLATION STUDIES

We conduct a comprehensive ablation analysis along five axes to evaluate the effectiveness and necessity of HCD's design components:

- *Loss Components*: Evaluate the individual contributions of vMF uniformity, spherical contrastive alignment, Student-*t* cluster regularization, and entropy regularization;
- *Latent Geometry*: Compare hyperspherical versus Euclidean latent spaces while keeping the decoder fixed;
- *Diffusion & Aggregation*: Vary the number of diffusion steps T, the noise schedule, and aggregation mechanisms including Product-of-Experts (PoE);
- *Sensitivity*: Assess the impact of contrastive hyperparameters ($n_{\text{neg}}, m, \alpha$) and loss weights;
- *Robustness*: Test model behavior under edge sparsification, feature corruption, and mismatched cluster counts.

Unless stated otherwise, all results are reported as mean $\pm$ standard error over five random seeds, using Hungarian-matched ACC, NMI, and ARI. Experiments are conducted on one large-scale graph (OGBN-ARXIV) and one medium-scale graph (ACM) to capture performance across domains and scales.

A.1 DO THE PROPOSED OBJECTIVES MATTER?

Table A1 presents results from ablations in which each loss term is removed individually. We observe the following:

- Removing vMF uniformity increases hub attraction, degrading cluster separation on OGBN-ARXIV.
- Disabling spherical contrastive alignment reduces boundary sharpness, leading to consistent drops in ARI.
- Omitting the cluster compactness/separation regularizer destabilizes assignments and reduces performance.
- Eliminating the entropy penalty results in mode collapse, especially under class imbalance.

These findings support the complementary roles of geometry-aware priors, contrastive alignment, and distributional regularization in stabilizing training and promoting discriminative clustering.

Table A1: Component-wise ablations on two representative datasets. Each row removes exactly one component.

Variant	ogbn-arxiv			ACM		
	ACC↑	NMI↑	ARI↑	ACC↑	NMI↑	ARI↑
Full HCD	**41.3**±0.62	**50.4**±0.45	**31.5**±0.43	**86.7**±0.28	**60.7**±0.51	**67.4**±0.22
w/o vMF uniformity	39.2±0.58	48.1±0.44	29.1±0.39	85.6±0.33	58.8±0.46	65.1±0.28
w/o spherical contrast	38.6±0.55	47.6±0.47	27.8±0.41	85.1±0.35	58.2±0.49	64.0±0.30
w/o cluster regularizer	37.9±0.63	47.1±0.50	27.1±0.45	84.9±0.37	57.9±0.52	63.6±0.32
w/o entropy penalty	38.1±0.39	47.3±0.32	27.3±0.49	85.2±0.60	58.4±0.58	63.9±0.4 9

A.2 Is Hyperspherical Geometry Necessary?

To assess the contribution of hyperspherical embeddings, we compare them against Euclidean latent spaces while keeping the decoder fixed to cosine similarity. As shown in Table 4, the hyperspherical geometry consistently improves cluster boundary sharpness and inter-cluster separation, especially on large and heterophilous graphs, where angular distance metrics scale more stably than Euclidean norms.

Table A2: Effects of diffusion depth T and aggregation strategy.

Setting	ogbn-arxiv			ACM		
	ACC↑	NMI↑	ARI↑	ACC↑	NMI↑	ARI↑
T=5, PoE	39.9±0.37	49.2±0.35	30.1±0.29	86.0±0.32	59.8±0.49	66.3±0.27
T=10, PoE	**41.3**±0.62	**50.4**±0.45	**31.5**±0.43	**86.7**±0.28	**60.7**±0.51	**67.4**±0.22
T=20, PoE	41.2±0.50	50.3±0.47	31.4±0.46	86.7±0.29	60.6±0.57	67.3±0.29
T=15, mean-agg	39.1±0.68	48.5±0.47	29.2±0.41	85.5±0.42	59.1±0.44	65.3±0.39
T=15, attn-agg	39.6±0.41	48.9±0.49	29.8±0.46	85.8±0.33	59.4±0.48	65.9±0.26
T=15, PoE	40.9±0.61	50.0±0.46	31.0±0.42	86.4±0.54	60.3±0.46	67.1±0.37

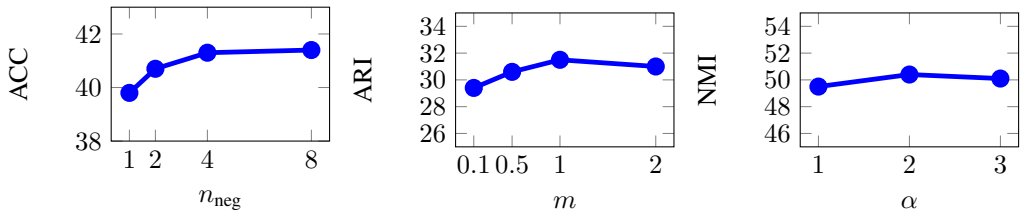

Figure A1: Sensitivity analysis over key contrastive and regularization parameters.

A.3 How Much Diffusion and How to Aggregate?

We study the effects of diffusion depth T and aggregation strategy. Table A2 shows that a moderate depth ($T = 10$) yields strong performance, with diminishing returns beyond that point.

Among aggregation schemes, PoE outperforms naive averaging and attention-based alternatives by adaptively down-weighting uncertain timesteps via inverse variance.

To disentangle the effect of PoE itself from the temporal reweighting induced by γ_t, we further compare PoE with fixed uniform weights ($w_t = 1/T$) against PoE with learnable γ_t (Table A3). Both variants substantially outperform mean / attention-based aggregation, confirming that inverse-variance fusion is the main driver of the gains, while learnable γ_t provides small but consistent improvements across datasets (about 0.5–1.0 points in ACC/ARI).

Table A3: Ablation on temporal weights w_t in PoE aggregation. We compare fixed uniform weights and learnable γ_t on two representative datasets.

Variant	ogbn-arxiv			ACM		
	ACC↑	NMI↑	ARI↑	ACC↑	NMI↑	ARI↑
PoE (uniform $w_t = 1/T$)	40.7±0.59	49.8±0.48	30.7±0.40	86.3±0.31	60.1±0.46	66.8±0.26
PoE (learnable γ_t)	**41.3±0.62**	**50.4±0.45**	**31.5±0.43**	**86.7±0.28**	**60.7±0.51**	**67.4±0.22**

Table A4: Robustness under corruption. We report ogbn-arxiv results when applying edge drop and feature noise at test time.

Setting	Edge drop (%)			Feature noise (stdev)		
	0	20	40	0.0	0.5	1.0
ACC↑	41.3	39.6	37.9	41.3	40.5	39.1
NMI↑	50.4	48.9	47.2	50.4	49.6	48.1
ARI↑	31.5	29.8	27.9	31.5	30.6	29.1

A.4 Sensitivity to Contrastive Parameters and Loss Weights

We perform sensitivity analysis over key contrastive and regularization parameters. Figure A1 shows that small angular margins m yield weak separation, while overly large m over-penalizes mid-range pairs. Increasing n_{neg} improves performance until saturation. We sweep $\lambda_{\text{UNI}}, \lambda_{\text{aln}}, \lambda_{\text{clu}}, \lambda_{\text{ent}}$ on a log scale and find stable optima across datasets. A practical range for loss weights (from log-scale search) that works across datasets is: $\lambda_{\text{UNI}} \in [0.01, 1]$, $\lambda_{\text{aln}} \in [0.1, 1]$, $\lambda_{\text{clu}} \in [0.1, 2]$, $\lambda_{\text{ent}} \in [0.001, 0.1]$.

A.5 Robustness Under Graph and Feature Corruption

To simulate noisy real-world conditions, we randomly drop edges and inject Gaussian noise into node features (z-scored per dimension) during testing. Table A4 shows that HCD degrades gracefully, maintaining stronger NMI and ARI than baselines. We attribute this robustness to: (i) angular normalization, which reduces noise amplification; and (ii) diffusion-based encoding, which aggregates across steps and mitigates over-smoothing.

A.6 Cluster Count Mismatch and Assignment Stability

In realistic deployments, the true number of clusters is often unknown. We evaluate HCD's stability under mismatched K values by varying the number of clusters around the ground truth. Table A5 shows that HCD remains stable within a range of $K \pm 2$, with smooth degradation beyond that. The entropy regularization term plays a key role in preventing fragmentation when K is overestimated.

A.7 Robustness to Structural and Heterogeneous Noise

To further assess robustness beyond simple edge sparsification and global Gaussian noise, we evaluate HCD under (i) degree-preserving structural perturbations and (ii) heterogeneous feature corruption. For structural noise, we randomly rewire a fraction $\rho \in \{0.2, 0.4\}$ of edges while approximately preserving the degree distribution. For heterogeneous feature noise, we select 30% of the nodes and, only for this subset, mask 50% of the feature dimensions and add high-variance Gaussian noise, while the remaining nodes receive mild Gaussian perturbations as in the main robustness experiment.

Table A6 reports the results on OGBN-ARXIV. Overall, clustering performance decreases smoothly as the noise level increases, and HCD maintains reasonable ACC/NMI/ARI under all tested structural and heterogeneous noise settings.

Table A5: Cluster-count K sensitivity on ACM.

K	$K-2$	$K-1$	K (nominal)	$K+1$	$K+2$	$K+4$
ACC↑	84.1	85.7	**86.7**	86.1	85.4	83.9
NMI↑	58.6	60.0	**60.7**	60.2	59.4	57.8
ARI↑	64.8	66.5	**67.4**	66.8	66.0	64.1

Table A6: Robustness of HCD to structural and heterogeneous feature noise on OGBN-ARXIV.

Noise setting	ACC↑	NMI↑	ARI↑
Clean graph	41.3	50.4	31.5
20% edge rewiring	38.7	47.8	28.7
40% edge rewiring	35.6	44.8	25.3
Heterogeneous feature noise	38.4	47.7	28.5
Rewiring + heterogeneous noise	35.8	45.1	25.1

B Specific Hyperparameters

We summarize the default hyperparameter settings in Table A7, and provide per-dataset configurations in Table A8, covering all experiments in this work.

C Geometric Guarantee: Angular Margin Implies Separation

Let $\hat{\boldsymbol{Z}} \in \mathbb{S}^{N\times(d-1)}$ be the normalized latent embeddings, and let $\mathcal{E}^+$ and $\mathcal{E}^-$ denote the sets of positive and negative pairs. The alignment loss is:

$$\mathcal{L}_{\text{aln}} = \frac{1}{|\mathcal{E}^+|}\sum_{(i,j)\in\mathcal{E}^+}\|\hat{\boldsymbol{z}}_i-\hat{\boldsymbol{z}}_j\|_2^{\alpha} + \frac{1}{|\mathcal{E}^-|}\sum_{(i,j)\in\mathcal{E}^-}\big[m-\|\hat{\boldsymbol{z}}_i-\hat{\boldsymbol{z}}_j\|_2\big]_+^{\alpha}, \tag{13}$$

where $m\in(0,2)$ is the margin, and $\alpha\geq 1$. Assume the embeddings follow a K-component vMF posterior with concentration κ and means $\boldsymbol{\mu}_k{}_{k=1}^{K}$.

Proposition 1 (Angular Separation Lower Bound). *If $\mathcal{L}_{\text{aln}}=0$ and each vMF component satisfies $\kappa\geq\kappa_{\min}>0$, then for any two clusters $k\neq\ell$, the angle $\theta_{k\ell}=\arccos(\boldsymbol{\mu}_k^\top\boldsymbol{\mu}_\ell)$ obeys*

$$\theta_{k\ell} \;\geq\; 2\arcsin\Big(\frac{m}{2}\Big) \;-\; \underbrace{\mathcal{O}\big(\kappa_{\min}^{-1/2}\big)}_{\textit{intra-cluster spread}}. \tag{14}$$

Proof: On the unit sphere, $\|\hat{\boldsymbol{z}}_i-\hat{\boldsymbol{z}}_j\|_2 = 2\sin(\frac{\theta_{ij}}{2})$ with θ_{ij} the angle between embeddings. A zero hinge loss on negatives forces $\|\hat{\boldsymbol{z}}_i-\hat{\boldsymbol{z}}_j\|_2\geq m$ hence $\theta_{ij}\geq 2\arcsin(m/2)$ across negatives.

Table A7: Global hyperparameters used across all datasets.

Category	Symbol	Value
Latent dimension	d	64 / 128
Diffusion steps	T	30 / 10 / 20
KL weight	λ_{UNI}	0.1
Contrastive weight	λ_{aln}	0.1
Cluster reg. weight	λ_{clu}	0.1
Entropy weight	λ_{ent}	2e-3
Student-t d.o.f.	ν	1
negative sampling rate	n_{neg}	8
Distance power	α	2
Margin	m	1
Learning rate	lr	1×10^{-4}
vMF LR	lr_κ	1×10^{-3}
Epochs		1000

Table A8: Dataset-specific configurations.

Dataset	d	T	$\lvert\mathcal{V}\rvert$	Notes
ogbn-arxiv	128	10	169,343	citation (OGB)
ogbn-products	128	10	2,449,029	co-purchase (OGB)
Wiki	64	10	2,405	web graph
ACM	128	10	3,025	co-author
DBLP	128	10	26,128	co-author
STARmap	64	20	1,207	tissue RNA-seq
DLPFC	64	20	3,888	Visium #151672
BRCA	64	20	3,798	Visium tumour slice

Within a vMF distribution, samples concentrate in a cone of half-angle $\mathcal{O}(\kappa^{-1/2})$ around $\boldsymbol{\mu}_k$. Thus, the angle between any two cluster means must exceed the negative-pair threshold minus twice the intra-cluster spread, establishing the bound. □

Implication. The combination of vMF uniformity (bounded κ) and angular margin m guarantees a nontrivial minimum inter-cluster angle, strengthening cluster separability.

D Why Product-of-Experts Helps: Inverse-Variance Fusion

At timestep t, each node has a Gaussian posterior $\mathcal{N}(\boldsymbol{\mu}^{(t)}, \mathrm{diag}(\boldsymbol{\sigma}^{2\,(t)}))$. Assuming conditional independence across t, the product posterior is

$$p(\boldsymbol{z} \mid \{t\}) \propto \prod_{t=1}^{T} \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}^{(t)}, \mathrm{diag}(\boldsymbol{\sigma}^{2\,(t)})) = \mathcal{N}\Big(\boldsymbol{z};\, \bar{\boldsymbol{\mu}},\, \mathrm{diag}\big(\bar{\boldsymbol{\sigma}}^{2}\big)\Big), \tag{15}$$

with precisions $\boldsymbol{P}^{(t)} = \mathrm{diag}\big(1/\boldsymbol{\sigma}^{2\,(t)}\big)$ and

$$\bar{\boldsymbol{\mu}} = \Big(\sum_t \boldsymbol{P}^{(t)}\Big)^{-1}\Big(\sum_t \boldsymbol{P}^{(t)}\boldsymbol{\mu}^{(t)}\Big), \quad \bar{\boldsymbol{\sigma}}^{-2} = \sum_t \boldsymbol{P}^{(t)}. \tag{16}$$

The learnable weights w_t modulate per-step trust. When the diffusion noise is high, resulting in large $\boldsymbol{\sigma}^{2,(t)}$, the PoE mechanism naturally down-weights these steps.

E Spherical Similarity Calibration

Let $\hat{\boldsymbol{z}} \sim \mathrm{vMF}(\boldsymbol{\mu}, \kappa)$ be a unit vector on $\mathbb{S}^{d-1}$. For any unit reference $\boldsymbol{v} \in \mathbb{S}^{d-1}$, the inner product $\hat{\boldsymbol{z}}^\top \boldsymbol{v}$ concentrates around its mean with a sub-Gaussian tail: there exists an absolute constant $c > 0$ such that, for any $\epsilon > 0$,

$$\Pr\big(\,|\hat{\boldsymbol{z}}^\top \boldsymbol{v} - \mathbb{E}[\hat{\boldsymbol{z}}^\top \boldsymbol{v}]| \geq \epsilon\,\big) \;\leq\; 2\exp\big(-c\,\kappa\,\epsilon^2\big). \tag{17}$$

Here, $\Pr$ denotes probability and $\mathbb{E}[\cdot]$ denotes expectation. Larger κ implies stronger concentration and hence more stable similarities.

Implication for hinge thresholds: Because inner products are tightly concentrated, a fixed hinge threshold in the alignment loss induces a predictable error rate for negatives. This threshold is therefore approximately calibrated when κ is moderate. In practice, we use $\kappa \in [5, 20]$, which yields close agreement between expected hinge activation and the observed false negative rate.

F Detailed Diffusion Process

In the main text we define the diffusion reconstruction by

$$\boldsymbol{S}^{(t)} = \mathcal{N}(\boldsymbol{z}^{(t)})\,\mathcal{N}(\boldsymbol{z}^{(t)})^\top, \tag{18}$$

$$\hat{\boldsymbol{A}} = \frac{\sum_{t=1}^{T} c_t\, \boldsymbol{S}^{(t)}}{\sum_{t=1}^{T} c_t} \in [0,1]^{N\times N}, \tag{19}$$

Algorithm 2 Noise-free forward process $q(\mathcal{A}_{t-1} \mid \mathcal{A}_t, \mathcal{A}_0)$

1: **Input:** observed adjacency $\mathcal{A}$, number of steps T
2: Initialize terminal state: $\mathcal{A}_T \leftarrow \mathcal{A}$
3: **for** $t = T, T-1, \ldots, 1$ **do**
4: Set $\mathcal{A}_{t-1} \leftarrow \mathcal{A}_t$
5: $\{q(\mathcal{A}_{t-1} \mid \mathcal{A}_t, \mathcal{A}_0)$ is the identity (noise-free) transition$\}$
6: **end for**
7: **Return** $\{\mathcal{A}_t\}_{t=0}^{T}$

Algorithm 3 Reverse process $p_\theta(\hat{\boldsymbol{A}}_{t-1} \mid \hat{\boldsymbol{A}}_t)$

1: **Input:** latent embeddings $\{\boldsymbol{z}^{(t)}\}_{t=1}^{T}$, schedule $\{c_t\}_{t=1}^{T}$
2: **Output:** reconstructed adjacency $\hat{\boldsymbol{A}}$
3: Initialize reconstruction at terminal step: $\hat{\boldsymbol{A}}_T \leftarrow \mathbf{0} \in \mathbb{R}^{N \times N}$
4: **for** $t = T, T-1, \ldots, 1$ **do**
5: Compute normalized similarities: $\boldsymbol{S}^{(t)} \leftarrow \mathcal{N}(\boldsymbol{z}^{(t)})\,\mathcal{N}(\boldsymbol{z}^{(t)})^\top$
6: Update reconstruction: $\hat{\boldsymbol{A}}_{t-1} \leftarrow \hat{\boldsymbol{A}}_t + c_t\,\boldsymbol{S}^{(t)}$
7: **end for**
8: Normalize the accumulated matrix using Eq. equation 23: $\hat{\boldsymbol{A}} \leftarrow \hat{\boldsymbol{A}}_0 / \big(\sum_{s=1}^{T} c_s\big)$
9: **Return** $\hat{\boldsymbol{A}}$

where $\mathcal{N}(\cdot)$ denotes row-wise ℓ_2-normalization, $\{\boldsymbol{z}^{(t)}\}_{t=1}^{T}$ are the step-wise latent embeddings, and $c_t = \sqrt{1-\bar{\alpha}_t}$ follows a standard linear diffusion schedule. This appendix provides the algorithmic view of the reconstruction rule in Eq. equation 19 and shows that it is *exactly the closed-form solution* of a noise-free diffusion-style refinement process acting on the reconstructed adjacency matrix.

F.1 Noise-free Forward Process

We consider a degenerate, noise-free forward process on the adjacency matrix. Let $\mathcal{A} \in \mathbb{R}^{N \times N}$ denote the observed (normalized) adjacency, and let $\{\mathcal{A}_t\}_{t=0}^{T}$ denote its forward trajectory. We set the terminal state to the observed graph:

$$\mathcal{A}_T = \mathcal{A}, \tag{20}$$

and define the forward transition as the identity kernel:

$$q(\mathcal{A}_{t-1} \mid \mathcal{A}_t, \mathcal{A}_0) = \delta\big(\mathcal{A}_{t-1} - \mathcal{A}_t\big). \tag{21}$$

In our setting, the prediction target is a discrete adjacency matrix that encodes graph connectivity. Applying continuous Gaussian noise to $\mathcal{A}$ in the forward process would introduce two major issues: (i) it may disrupt the underlying graph topology by creating off-manifold or invalid edge patterns, and (ii) it introduces a mismatch with the decoder, which ultimately operates in a discrete (edge / non-edge) regime unless additional projection or thresholding mechanisms are introduced. To avoid these problems, we deliberately design a *noise-free, structure-preserving* forward process. So that all intermediate states satisfy $\mathcal{A}_T = \mathcal{A}_{T-1} = \cdots = \mathcal{A}_0 = \mathcal{A}$, as shown in Algorithm 2.

While the forward path remains deterministic and topologically faithful to the input graph, the reverse model is kept probabilistic and uncertainty-aware through the latent posteriors $q_t(\boldsymbol{z}^{(t)} \mid \mathcal{X}, \mathcal{A})$, which allows the model to capture diverse generative reconstructions in latent space.

F.2 Reverse Refinement and Closed-form Solution

On the decoder side, as described in Algorithm 3, we maintain a sequence of reconstructed adjacencies $\{\hat{\boldsymbol{A}}_t\}_{t=0}^{T}$ and refine them in a multi-step fashion. Starting from an initial reconstruction $\hat{\boldsymbol{A}}_T$ (e.g., the zero matrix), we update:

$$\hat{\boldsymbol{A}}_{t-1} = \hat{\boldsymbol{A}}_t + c_t\,\boldsymbol{S}^{(t)}, \qquad t = T, T-1, \ldots, 1, \tag{22}$$

where $\boldsymbol{S}^{(t)}$ is the similarity matrix computed from the step-specific embeddings $\boldsymbol{z}^{(t)}$, and $c_t = \sqrt{1-\bar{\alpha}_t}$ is the diffusion-style weight at timestep t. The final reconstruction is obtained by normalizing the accumulated matrix:

$$\hat{\boldsymbol{A}} = \frac{1}{\sum_{s=1}^{T} c_s} \hat{\boldsymbol{A}}_0. \tag{23}$$

Closed-form equivalence. Unrolling the recursion in Eq. equation 22 yields:

$$\hat{\boldsymbol{A}}_0 = \hat{\boldsymbol{A}}_T + \sum_{t=1}^{T-1} c_t \, \boldsymbol{S}^{(t)} = \sum_{t=1}^{T} c_t \, \boldsymbol{S}^{(t)}, \tag{24}$$

since $\hat{\boldsymbol{A}}_T$ is initialized to zero. Substituting this expression into Eq. equation 23 directly recovers the decoder in Eq. equation 19:

$$\hat{\boldsymbol{A}} = \frac{\sum_{t=1}^{T} (c_t \, \boldsymbol{S}^{(t)})}{\sum_{t=1}^{T} c_t}. \tag{25}$$

Therefore, the reconstruction rule used in the main text is *precisely* the closed-form solution of a noise-free diffusion-style refinement process on the adjacency matrix, where temporal structure is encoded in the sequence $\{\boldsymbol{S}^{(t)}\}_{t=1}^{T}$ and the diffusion schedule $\{c_t\}_{t=1}^{T}$.

G Additional Theoretical Discussion

This section provides further theoretical justification for two central design choices in HCD: (i) the use of multi-step diffusion with Product-of-Experts (PoE) aggregation, and (ii) the adoption of hyperspherical latent geometry. The goal is to complement the empirical evidence in Section 4 and to clarify under which assumptions these choices are expected to be beneficial. The detailed derivations are already contained in Appendix D, Appendix C, and Appendix E; here we summarize the main insights and their implications for clustering.

G.1 Multi-step Diffusion with PoE vs. Single-step Encoders

As discussed in Appendix D, the temporal encoder in HCD produces, for each node i and diffusion step $t \in \{1, \ldots, T\}$, a Gaussian latent posterior of the form:

$$\mathbf{z}_i^{(t)} \sim \mathcal{N}\big(\boldsymbol{\mu}_i^{(t)}, \mathrm{diag}(\boldsymbol{\sigma}_i^{2\,(t)})\big), \tag{26}$$

and the PoE aggregation combines these step-wise posteriors into a single Gaussian with mean $\bar{\boldsymbol{\mu}}_i$ and precision equal to the sum of per-step precisions (modulated by learnable temporal weights). To make the role of PoE explicit, we adopt the following simple generative view.

Gaussian approximation. Assume that for each node i there exists a latent "clean" representation $\mathbf{z}_i^\star$, and that each diffusion step provides an independent noisy estimate:

$$\mathbf{z}_i^{(t)} \sim \mathcal{N}\big(\mathbf{z}_i^\star, \Sigma_t\big), \quad t = 1, \ldots, T, \tag{27}$$

with Σ_t positive definite and conditionally independent across t given $\mathbf{z}_i^\star$. Under this model, the PoE posterior over $\mathbf{z}_i$ is Gaussian with

$$\bar{\mathbf{z}}_i \sim \mathcal{N}\left(\mathbf{z}_i^\star, \left(\sum_{t=1}^{T} \Sigma_t^{-1}\right)^{-1}\right), \tag{28}$$

i.e., the covariance of the aggregated estimate is the inverse of the sum of per-step precisions. This corresponds exactly to the inverse-variance fusion derived in Appendix D.

Proposition 2 (Variance reduction of temporal PoE)**.** *Under the Gaussian approximation above, the PoE estimate $\bar{\mathbf{z}}_i$ is the minimum-variance linear unbiased estimator of $\mathbf{z}_i^\star$. Moreover,*

$$\left(\sum_{t=1}^{T} \Sigma_t^{-1}\right)^{-1} \preceq \Sigma_t \quad \textit{for all } t \in \{1, \ldots, T\}, \tag{29}$$

that is, the covariance of $\bar{\mathbf{z}}_i$ is no larger (in the Loewner ordering) than the covariance at any single diffusion step. The single-step encoder is recovered as the special case $T{=}1$.

Proof. The result follows from standard Gaussian conditioning and the properties of generalized least squares: the PoE posterior coincides with the solution of a linear minimum-variance estimation problem where each $\mathbf{z}_i^{(t)}$ is treated as a noisy observation of the same ground-truth latent $\mathbf{z}_i^\star$. We refer to Appendix D for the full derivation.

Implications for clustering. Our clustering objective ultimately depends on pairwise distances between node embeddings. Proposition 2 shows that, under mild assumptions, multi-step diffusion with PoE yields embeddings that are more concentrated around the underlying latent $\mathbf{z}_i^\star$ than any single-step encoder. This variance reduction directly translates into: (i) tighter intra-cluster scatter, and (ii) more stable decision boundaries between clusters. Intuitively, different diffusion steps provide complementary views of each node at different noise levels; PoE performs principled inverse-variance fusion of these views instead of committing to an arbitrary single step. This theoretical perspective helps explain why, in our experiments, multi-step diffusion with PoE consistently improves clustering performance over single-step encoders.

G.2 When Does Hyperspherical Geometry Help?

HCD constrains the aggregated mean directions $\hat{\mathbf{z}}_i$ to lie on the unit hypersphere $\mathbb{S}^{d-1}$ via row-wise ℓ_2-normalization, and uses a vMF prior as described in the main text. This choice interacts with the contrastive and clustering losses in two important ways.

Angular margins and cluster separation. On the hypersphere, the Euclidean distance between two unit vectors $\hat{\mathbf{z}}_i$ and $\hat{\mathbf{z}}_j$ is a monotone function of the geodesic angle θ_{ij}:

$$\|\hat{\mathbf{z}}_i - \hat{\mathbf{z}}_j\|_2 \;=\; 2\sin(\theta_{ij}/2). \tag{30}$$

Therefore, the hinge margin used in our spherical contrastive loss can be interpreted as an *angular* margin. Appendix C formalizes this intuition: Proposition 1 shows that, when the vMF components have non-trivial concentration, driving the alignment loss to zero enforces a strictly positive lower bound on the angle between any two cluster means. Equivalently, the combination of hyperspherical normalization, vMF regularity, and angular margin yields a guaranteed minimum inter-cluster separation in angle space. This kind of margin-to-angle guarantee does *not* hold in unconstrained Euclidean space, where the same Euclidean distance may arise from different combinations of norms and angles.

Norm control and avoidance of hub dominance. In an unconstrained Euclidean latent space, the encoder is free to increase vector norms without bound. As a consequence, distances can be dominated by a few "hub" nodes with very large norms, leading to skewed cluster centers and blurred cluster boundaries. By constraining all embeddings to $\mathbb{S}^{d-1}$ and using a uniform hyperspherical prior, HCD removes this degree of freedom and encourages representations that are primarily encoded in the direction rather than in the magnitude. Appendix E further shows that the vMF distribution yields sub-Gaussian concentration of inner products around their mean, which stabilizes similarity scores and makes the hinge thresholds used in our loss approximately calibrated.

H Computational Complexity

H.1 Theoretical Analysis

For a graph with $|\mathcal{V}|$ nodes, $|\mathcal{E}|$ edges, diffusion steps T, and latent dimension d, a single forward pass of HCD costs:

$$\mathcal{O}\Big(T\big(\underbrace{|\mathcal{E}|}_{\text{GCN message passing}} + \underbrace{|\mathcal{V}|d}_{\text{PoE + vMF ops}}\big)\Big), \tag{31}$$

which is *linear* in both edges and nodes. The spherical contrastive and Student-t losses add: $\mathcal{O}\big(|\mathcal{E}| + |\mathcal{V}|K\big)$, which is dominated by the GCN term when $K \ll |\mathcal{V}|$.

H.2 Scalability on Large-Scale Graphs

We briefly clarify how HCD is implemented to remain practical on million-node graphs.

Table A9: Training time (seconds ± std) per epoch on OGB graphs (single RTX 4090, 24 GB).

Method	ogbn-arxiv	ogbn-products
S^3GC	29 ± 3.1	89 ± 3.9
CVGAE	36 ± 4.0	117 ± 5.7
HCD (ours)	47 ± 5.7	145 ± 5.3

Sparse encoder. The adjacency matrix is stored and used in sparse form throughout training. Each graph convolution layer applies a sparse–dense matrix multiplication of the form $\mathcal{A}\mathcal{H}$, where $\mathcal{A}$ is the normalized adjacency and $\mathcal{H}$ is a dense node feature matrix. The cost of the encoder is therefore $\mathcal{O}(|\mathcal{E}|d)$ per layer, and no dense $N \times N$ matrices are materialised.

Edge-wise diffusion decoder and contrastive loss. In practice we never construct $\hat{\mathcal{A}}$ or any $N \times N$ similarity matrix explicitly. The diffusion reconstruction loss and the spherical contrastive loss are both evaluated only on node pairs: observed edges and randomly sampled non-edges. Let B be the number of such pairs used in one optimisation epoch. For each pair and each diffusion step, we sample latent representations, project them to the hypersphere, and accumulate cosine similarities. This yields a decoder and contrastive cost of $\mathcal{O}(TBd)$ in time and $\mathcal{O}(d)$ per micro-batch in memory, independent of N.

Clustering regularisation. The Student-t cluster compactness/separation term and the entropy regulariser operate on node–centroid distances, with complexity $\mathcal{O}(|\mathcal{V}|Kd)$, where K is the number of clusters (typically small). The vMF mixture used at the end of training has the same order. These terms do not introduce any quadratic dependence on N.

Overall cost on large graphs. A training epoch therefore consists of one full-batch sparse GNN encoder pass, a fixed number of edge-wise diffusion/contrastive updates, and node-wise clustering regularisation, with overall cost $\mathcal{O}(|\mathcal{E}|d + TBd + |\mathcal{V}|Kd)$. In our experiments this setup scales to graphs with up to 1.9×10^6 nodes and 2.1×10^7 edges using a single modern GPU, without additional system-level optimisations.

Memory footprint. In terms of memory, HCD does not materialise any dense $N \times N$ adjacency or similarity matrices. The encoder stores node features and latent activations for all nodes, which requires $\mathcal{O}(|\mathcal{V}|(F + Td))$ memory, where F is the input feature dimension and T is the diffusion depth (typically $T \leq 20$ in our experiments). The normalized adjacency is held in a sparse format, which adds $\mathcal{O}(|\mathcal{E}|)$ memory. The edge-wise diffusion decoder and spherical contrastive loss operate on sampled node pairs only, and therefore use $\mathcal{O}(Bd)$ memory per mini-batch for pair embeddings. The clustering and vMF mixture components depend on K cluster centres and mixture components, contributing $\mathcal{O}(Kd)$ parameters. Ignoring constant factors due to back-propagation, the peak memory usage is thus

$$\mathcal{O}\big(|\mathcal{V}|(F + d) + |\mathcal{E}| + Bd + Kd\big), \tag{32}$$

up to constant factors in T, which is linear in graph size.

Practical accuracy–efficiency trade-offs. In resource-constrained settings, HCD exposes several simple knobs that allow practitioners to trade clustering accuracy for runtime and memory: (i) reducing the diffusion depth T, which decreases both encoder and decoder cost almost linearly; as shown in Table A2, T=5 or T=10 already yields strong performance; (ii) decreasing the number of sampled node pairs B used in the diffusion decoder and spherical contrastive loss, which lowers the $\mathcal{O}(TBd)$ component while preserving linear scaling in $|\mathcal{V}|$ and $|\mathcal{E}|$; (iii) using a smaller latent dimension d on extremely large graphs; and (iv) applying standard early-stopping criteria based on clustering metrics or proxy objectives on a small validation split.

H.3 Empirical Wall-Clock Benchmarks

As shown in Table A9, HCD is approximately $1.2\times$ to $1.4\times$ slower per epoch than a standard VGAE. This overhead arises from: (i) $T = 10$ diffusion steps, (ii) additional objective terms, and (iii) the

Table A10: End-to-end time-to-target comparison on OGB datasets. For each method, we report the best ARI achieved during training, the number of epochs and wall-clock time required to reach 95% of that ARI (*time-to-0.95×ARI$_{best}$*), and the total time to reach ARI$_{\text{best}}$. All runs use a single RTX 4090 GPU (24 GB).

Dataset	Method	ARI$_{\text{best}}$ (%)	Epochs to 0.95×ARI$_{\text{best}}$	Time to 0.95×ARI$_{\text{best}}$ (min)	Epochs to ARI$_{\text{best}}$	Time to ARI$_{\text{best}}$ (min)
ogbn-arxiv	CVGAE	27.6	284	170.4	392	235.2
	HCD (ours)	**31.5**	231	181.0	315	246.8
ogbn-products	CVGAE	22.6	178	347.1	246	479.7
	HCD (ours)	**28.2**	149	360.1	210	507.5

absence of sigmoid activations, which reduces memory traffic and computation per step, keeping the runtime overhead moderate.

H.4 End-to-end time-to-target analysis

To make the accuracy–compute trade-off more explicit, we further provide an end-to-end *time-to-target* analysis in Table A10.

For each method and dataset, we track the clustering ARI over training and define ARI$_{\text{best}}$ as the maximum value attained. We then compute the smallest epoch index $e^\star$ such that $\text{ARI}_{e^\star} \geq 0.95\times\text{ARI}_{\text{best}}$ and report both $e^\star$ and the corresponding wall-clock time (*time-to-0.95×ARI$_{best}$*). We also report the number of epochs and total time required to reach ARI$_{\text{best}}$ itself.

On both OGBN-ARXIV and OGBN-PRODUCTS, HCD requires a modest per-epoch overhead compared to CVGAE (Table A9), but attains substantially higher ARI with a comparable end-to-end wall-clock budget, thereby quantifying the claimed accuracy–efficiency trade-off.

I Extension to Heterogeneous and Dynamic Graphs

Our empirical evaluation in the main paper focuses on static, homogeneous graphs and spatial transcriptomics graphs. Although we do not provide experiments on heterogeneous or dynamic graph families, the structure of HCD makes it possible, to combine the proposed hyperspherical diffusion framework with encoders that are designed for heterogeneous or dynamic graphs. Below we briefly discuss how such extensions could be formulated.

Heterogeneous graphs. In heterogeneous graphs, nodes and edges can belong to multiple types and relation categories. The core components of HCD (PoE-based temporal aggregation, projection onto the unit hypersphere, the vMF KL regularizer, the spherical contrastive alignment loss, and the Student-t cluster compactness–separation regularizer) are all defined on node embeddings and on sampled node pairs. They do not depend on a particular choice of encoder as long as the encoder produces node representations.

In principle, one can replace the homogeneous GCN encoder $\phi^{(t)}, \phi_\mu, \phi_\sigma$ in Eq. (1) with a heterogeneous GNN that explicitly models different node and edge types, while keeping the remainder of the HCD framework unchanged. In that case, the adjacency $\mathcal{A}$ in our notation can be interpreted as an aggregation or block representation over relation-specific adjacency matrices. This modification, necessitates the design of relation-aware positive and negative sampling strategies for the spherical contrastive loss. A systematic empirical investigation of these design choices lies beyond the scope of this work.

Dynamic graphs. Dynamic graphs exhibit time-varying node features and/or edge sets. In our current experiments, we work with a single static snapshot $(\mathcal{X}, \mathcal{A})$ for each dataset, and the diffusion depth index $t = 1, \ldots, T$ is an internal denoising time for the latent variables rather than a graph time index.

To handle dynamic graphs with evolving topology, one possible extension is a snapshot-based formulation. Given a sequence of graphs $\{\mathcal{G}^{(\tau)} = (\mathcal{V}^{(\tau)}, \mathcal{E}^{(\tau)})\}_{\tau=1}^{T_{\text{dyn}}}$, HCD can be applied to each snapshot with shared parameters, which produces embeddings $\{\hat{\boldsymbol{Z}}^{(\tau)}\}$ that are then aggregated across

graph time, for example by temporal averaging, attention, or a PoE-style fusion, before clustering. An alternative formulation is to treat the graph time index τ explicitly in the encoder, replacing $\phi^{(t)}(\mathcal{X}, \mathcal{A})$ by $\phi^{(t,\tau)}(\mathcal{X}^{(\tau)}, \mathcal{A}^{(\tau)})$ while reusing the same hyperspherical and clustering objectives over the joint set of embeddings. Both variants preserve the geometric structure of HCD but add an extra temporal dimension. A comprehensive empirical evaluation in these settings is an important direction for future work.

J Additional biological interpretation of spatial transcriptomics results

Here we provide additional qualitative comments on the spatial transcriptomics experiments in Figures 2–4, with an emphasis on how the inferred domains relate to known tissue biology.

STARmap mouse visual cortex. The STARmap annotations we use correspond to expert-defined neuronal and glial cell types in the mouse visual cortex reported by Wang et al. (2018b) and subsequent studies, which are based on transcriptomic profiles and previously established cortical markers and cell-type taxonomies. The "True annotation" panel in Figure 2 shows a laminar organization from superficial to deep layers. Compared to other methods, HCD produces clusters whose spatial arrangement and boundaries more closely track these laminar transitions: superficial excitatory layers, intermediate layers, and deeper layers are separated by relatively sharp, horizontally aligned borders, with only limited mixing across layer boundaries. This qualitatively matches the expected columnar–laminar organization of the visual cortex.

Human DLPFC (10x Visium). For the DLPFC section, the ground-truth labels we use follow Maynard et al. (2021) and later re-analyses, where cortical layers and white matter are delineated based on histology (H&E-stained sections) and known layer-specific markers. In Figure 3, HCD recovers six cortical layers and white matter as spatially continuous bands with smooth boundaries and a narrow layer-4 band between upper and deeper layers. Several competing methods either fragment the white-matter region or blur the thin layer-4 domain into neighboring layers. The close alignment between HCD's clusters and these curated layer labels suggests that the hyperspherical diffusion embeddings preserve both the laminar architecture and the underlying molecular distinctions used by experts to annotate this dataset.

Breast carcinoma (BRCA) tissue. The BRCA "True annotation" in Figure 4 follows the manual domain labels provided with the Visium sample Polyak et al. (2011), which distinguish different in situ and invasive carcinoma regions, tumor-edge zones, and histologically healthy tissue. Across different choices of k (10, 15, 20), HCD yields spatially coherent tumor cores, peri-tumoral edge regions, and stromal/normal compartments. In particular, the main tumor mass remains a contiguous cluster, while peri-tumoral clusters form shells around it instead of being scattered throughout the slide. This behavior is consistent with the expected spatial organization of carcinoma versus stroma in breast cancer tissue and indicates that the learned clusters respect both spatial continuity and the expert-defined pathological compartments.

Overall, these qualitative observations complement the ARI/AMI scores in the main text: because the expert annotations themselves are derived from histology and marker-based domain definitions, the strong agreement between HCD and the "True annotation" panels provides indirect biological validation without relying on additional supervision. A more detailed, gene-level marker enrichment analysis would be a valuable extension, but we leave such systematic biological studies to future work.

K Latent-space diffusion vs. graph-space operations

We analyze the rationale for applying the diffusion process in the latent space instead of directly operating on the raw adjacency matrix or node features.

Avoiding repeated mixing in graph space. Over-smoothing phenomena in deep GNNs have been widely attributed to repeatedly propagating information along graph edges using the same or similar

propagation rules. As more layers are stacked, node representations within a connected component can become nearly indistinguishable, and community boundaries are progressively blurred. In HCD we explicitly decouple these two roles: information propagation over the graph is handled by a relatively shallow GNN encoder, and the subsequent multi-step refinement is carried out purely in the latent space without further message passing along edges. This design allows us to benefit from a deep reverse refinement chain while avoiding additional rounds of graph-space mixing that would exacerbate over-smoothing.

Why diffuse in the latent space instead of on adjacency/features. At each diffusion step t, the encoder produces a Gaussian latent distribution with mean $\boldsymbol{\mu}^{(t)}$ and diagonal covariance. Under the Gaussian approximation used in Appendix D and Appendix G.1, the Product-of-Experts aggregation can be interpreted as an inverse-variance fusion of these step-wise posteriors: uncertain steps contribute less, and the aggregated latent embedding has smaller variance than any single step. Intuitively, the multi-step diffusion in HCD acts as a sequence of learned denoisers that progressively refine a shared "clean" latent representation, rather than repeatedly smoothing raw graph signals.

Applying such a denoising chain directly to the adjacency matrix would require operating in a very high-dimensional and combinatorial space, where small perturbations may easily destroy graph connectivity patterns that are important for clustering. Applying it directly to raw node features would ignore the structural information encoded by the graph and would treat all nodes as independent inputs. By first using a GNN encoder to map the graph into a hyperspherical latent space that already reflects the topology, and then running diffusion in that latent space, HCD lets the denoising process focus on refining cluster structure instead of reconstructing raw edges or features.

Relation to community structure. Graph-structured data used for clustering are often approximately piecewise-smooth: node features tend to be relatively homogeneous within communities, but can change more abruptly across community boundaries. The hyperspherical latent space in HCD, together with the vMF regularizer and the contrastive and clustering objectives, encourages different communities to occupy well-separated angular regions, which helps tighten intra-cluster cohesion without introducing additional mixing across communities in the original graph.