HHGD: HYPERBOLIC HYPERGRAPH DIFFUSION FOR VALID HYPERGRAPH GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Hypergraphs provide a natural representation for higher-order relations in social interactions, biological systems, recommender systems, and scientific networks, where pairwise graphs cannot faithfully capture group structure. Generating such data requires learning native node–hyperedge incidence patterns while producing discrete hyperedges with realistic cardinalities and non-degenerate overlaps. We propose **HHGD**, a hyperbolic latent diffusion framework that jointly represents nodes and hyperedges as tokens in hyperbolic space. HHGD combines an incidence-aware hyperbolic encoder, numerically stable diffusion in the origin tangent space, a curvature-aware hyperbolic score network, and a cardinality-aware Poincaré-distance decoder that maps generated tokens to binary incidence matrices. Experiments on synthetic and real-world hypergraph datasets show that HHGD can produce samples with exact uniqueness and novelty, while strict validity varies substantially across datasets and is strongest on several synthetic regimes, attaining the highest displayed mean strict validity on ER, Hypergraph SBM, Tree, and Tags-Math while remaining competitive in matching local and global structural distributions. Comprehensive architecture ablations further show that the hyperbolic latent components provide a coherent inductive bias for higher-order incidence generation, with their strongest benefits appearing in validity and hyperedge-size fidelity, while improvements in global topology remain dataset dependent. These results demonstrate the promise of hyperbolic latent diffusion for valid native hypergraph generation and clarify where its structural benefits are strongest.

1 INTRODUCTION

Many real-world systems are organized around group interactions rather than isolated pairs, ranging from co-authorship teams to biochemical networks. Hypergraphs directly capture these higher-order relations by allowing hyperedges to connect arbitrary node subsets, serving as a standard abstraction in complex systems (Feng et al., 2019; Do et al., 2020; Lee et al., 2025). Generating realistic hypergraphs supports simulation and reveals structural regularities (Do et al., 2020; Lee et al., 2025); however, advances in graph generators do not directly transfer (You et al., 2018; Jo et al., 2022; Vignac et al., 2023). Because hyperedges are combinatorial sets rather than simple node pairs, realistic generation must jointly capture variable cardinalities, heterogeneous participation, higher-order overlaps, and global topology (Do et al., 2020; Lee et al., 2025).

Existing hypergraph generators tackle scalability through varied approximations, including direct hyperedge parameterization, projection–reconstruction, bipartite reformulations, and local expansion strategies (Zuo et al., 2023; Wen & Yu, 2025; Gailhard et al., 2025). For instance, HYGENE expands hypergraphs over bipartite representations, while NoAH captures structure–attribute interplay via stochastic core–fringe mechanisms (Gailhard et al., 2025; Chun et al., 2025). Despite their progress, these methods often rely on Euclidean parameterizations, graph surrogates, or localized construction rules, thereby capturing native node–hyperedge incidence and hierarchical topology only indirectly.

We take a geometric view of this limitation. Many real hypergraphs exhibit strong heterogeneity: a small number of nodes or hyperedges may participate in many interactions, while most remain sparse or peripheral. Such patterns naturally suggest latent hierarchy and scale variation; Figure 1 provides a qualitative geometric illustration. Hyperbolic spaces are well suited for representing hierarchical and scale-free structures because their exponential volume growth allows tree-like organization to be

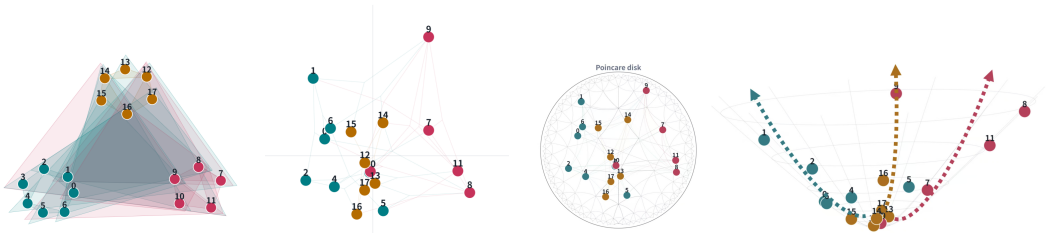
Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping 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.

embedded with low distortion (Ratcliffe, 2006; Nickel & Kiela, 2017; Ganea et al., 2018; Chami et al., 2019). However, directly combining hyperbolic geometry with generative modeling is nontrivial: diffusion models are most stable in Euclidean coordinates, whereas incidence reconstruction should respect node–hyperedge latent geometry.

To address these challenges, we propose **HHGD**, a hyperbolic latent diffusion framework that generates hypergraphs directly in their native node–hyperedge incidence form. HHGD jointly represents nodes and hyperedges as latent tokens in a hyperbolic latent space with a fixed shared interface curvature, avoiding reduction to a pairwise graph or dependence on a spectral intermediate. It maps the tokens to the tangent space at the origin, where a DDPM equipped with a curvature-aware hyperbolic score network learns their joint distribution, and maps the denoised tokens back to the Poincaré ball. A cardinality-aware Poincaré-distance decoder then converts node–hyperedge geometry and predicted hyperedge sizes into a discrete incidence matrix. By separating Euclidean diffusion coordinates from hyperbolic representation and decoding, HHGD combines stable generative learning with a geometry suited to heterogeneous and hierarchical higher-order structure (Ho et al., 2020; Song et al., 2021; Rombach et al., 2022; De Bortoli et al., 2022).

Our main contributions are summarized as follows:

- We propose **HHGD**, an end-to-end hyperbolic latent diffusion framework that generates native node–hyperedge incidence structures without reducing hypergraphs to pairwise graphs or relying on spectral intermediates.
- We develop an **incidence-aware hyperbolic encoder** and a **curvature-aware hyperbolic score network** that jointly model node and hyperedge tokens through Lorentz re-embedding in the encoder and Poincaré-space transformations in the denoiser, linked by a shared hyperbolic latent interface across both stages.
- We combine numerically stable **tangent-space diffusion** with a **cardinality-aware Poincaré-distance decoder**, translating continuous hyperbolic latents into discrete hyperedges with explicitly modeled cardinalities.



(a) Original hypergraph (b) Euclidean embedding (c) Hyperbolic embedding

Figure 1: **Visualization of node embeddings by singular value decomposition (SVD).** (a) Original structure visualization of the hypergraph, where different colors indicate different labels. (b) Visualization of node embeddings in 2D Euclidean space with planar projection. (c) Visualization of node embeddings in 2D hyperbolic space with Poincaré disk projection.The hyperbolic view provides a geometric representation in which nodes can occupy different radial scales, offering a natural inductive bias for heterogeneous or approximately hierarchical relations.

2 Related Work

Hypergraph Generative Models. Hypergraph generation must preserve variable hyperedge cardinalities, higher-order overlaps, and global topology beyond pairwise connectivity. While early work analyzed structural properties and recent surveys highlight its importance in hypergraph mining (Do et al., 2020; Lee et al., 2025), existing learning-based approaches tackle scalability through varied paradigms: direct hyperedge parameterization (Zuo et al., 2023), projection–reconstruction via surrogate graphs (Wen & Yu, 2025), bipartite-based diffusion expansion (Gailhard et al., 2025), and attribute-guided stochastic modeling (Chun et al., 2025). Standard graph generators also establish neural modeling for relational data (You et al., 2018; Jo et al., 2022; Vignac et al., 2023), yet fail

to capture native hypergraph incidence. Modern methods like HHGD address these limitations by preserving incidence throughout diffusion and learning hyperbolic latent representations directly without relying on graph surrogates.

Diffusion and Latent Generative Models. Diffusion models provide a powerful framework for generative modeling through iterative denoising, with foundational work establishing standard noise objectives (Ho et al., 2020), deterministic sampling (Song et al., 2021), and refined noise schedules (Nichol & Dhariwal, 2021). Latent diffusion models show that denoising in a learned latent space reduces generative complexity (Rombach et al., 2022), while diffusion Transformers demonstrate that self-attention backbones scale effectively for token generation (Vaswani et al., 2017; Peebles & Xie, 2023). These advances motivate operating over latent space rather than sparse incidence matrices directly. While graph diffusion models confirm the efficacy of diffusion for relational data (Jo et al., 2022; Huang et al., 2022; Vignac et al., 2023), they target pairwise structures. HHGD adapts latent diffusion to hypergraphs by fitting a DDPM prior over tangent-space coordinates of hyperbolic node and hyperedge tokens.

Hypergraph Representation Learning. Hypergraph neural networks directly model higher-order incidence structures, starting from spectral hypergraph convolutions (Feng et al., 2019) to graph approximations, unified message passing, and set-function formulations (Yadati et al., 2019; Huang & Yang, 2021; Chien et al., 2022). Recent hypergraph Transformers further emphasize node–hyperedge interactions and global token mixing (Li et al., 2023). Although primarily designed for Euclidean discriminative tasks, these incidence-aware designs naturally align with structural preservation in hypergraph generation. HHGD builds on this foundation by encoding nodes and hyperedges as latent tokens, while substituting Euclidean spaces with hyperbolic geometry to better capture hierarchical and heterogeneous topologies.

Hyperbolic and Riemannian Modeling. Hyperbolic representations effectively embed hierarchical and scale-free structures with low distortion (Ratcliffe, 2006; Nickel & Kiela, 2017). Hyperbolic neural networks extend standard operations to non-Euclidean manifolds (Ganea et al., 2018), with HGCN showing that trainable-curvature message passing captures graph hierarchies better than Euclidean GNNs (Chami et al., 2019). Subsequent work expands fully hyperbolic operations and diverse graph architectures (Liu et al., 2019; Dai et al., 2021; Chen et al., 2022). For generative modeling, Riemannian score-based models establish diffusion on manifolds (De Bortoli et al., 2022), while recent hyperbolic graph diffusion methods explore latent graph generation (Wen et al., 2024; Fu et al., 2024). HHGD shares this hyperbolic latent-diffusion perspective, but distinctly targets hypergraphs and preserves native node–hyperedge incidence throughout the pipeline, as categorized in Table 1.

Table 1: Comparison of representative graph and hypergraph generative paradigms.

Method	**Output**	**Representation**	**Generation Mechanism**
HYGENE (Gailhard et al., 2025)	Hypergraph	Bipartite	Euclidean local expansion
HypDiff (Fu et al., 2024)	Graph	Pairwise adjacency	Hyperbolic latent diffusion
HHGD (Ours)	Hypergraph	Native incidence	Tangent-space diffusion

3 METHODOLOGY

3.1 PROBLEM SETUP AND MODEL OVERVIEW

Let $G = (V, E)$ have n nodes and m hyperedges. Fixed dataset budgets $N \geq n$ and $M \geq m$ define a padded incidence matrix $H \in \{0, 1\}^{N \times M}$, where $H_{i,e} = 1$ iff node i belongs to hyperedge e (Feng et al., 2019). Zero columns denote inactive slots, so HHGD outputs a fixed-shape $\widehat{H}$ while allowing its active-edge count and cardinalities to vary. For reconstruction, sorted node-index tuples are assigned to columns lexicographically. This ordering is deterministic under a fixed node labeling, but the complete pipeline is not claimed to be permutation invariant.

All modules share the Poincaré interface $\mathbb{B}^d_{c_\star} = \{z \in \mathbb{R}^d : c_\star \|z\|_2^2 < 1\}$ with $c_\star = 1$:

$$\begin{aligned} Z_0 &= \mathrm{Enc}_\phi(H) = [Z_V; Z_E], \\ V_0 &= \log_0^{c_\star}(Z_0), \\ V &\sim p_\theta(\cdot|V_0), \\ (P, \widehat{s}) &= \mathrm{Dec}_\psi(\exp_0^{c_\star}(V)), \\ \widehat{H} &= \mathcal{Q}_\gamma(P, \widehat{s}). \end{aligned} \tag{1}$$

Here $Z_0 \in (\mathbb{B}^d_{c_\star})^{N+M}$ retains typed slot order, $V_0 \in \mathbb{R}^{(N+M)\times d}$ provides Gaussian coordinates, and $\mathcal{Q}_\gamma$ converts soft membership scores P and predicted sizes $\widehat{s}$ into a binary matrix by validation-selected top-k decoding; P is not an independent-Bernoulli law. The encoder's internal Lorentz scales are numerical and are not learned through backpropagation in the evaluated implementation.

Phase I jointly trains the encoder and decoder with incidence, topology, boundary, and cardinality losses. Phase II freezes them and learns a DDPM prior over V_0 (Ho et al., 2020; Nichol & Dhariwal, 2021; Rombach et al., 2022); its denoiser uses the same fixed uniform context during training and generation. Figure 2 summarizes the pipeline.

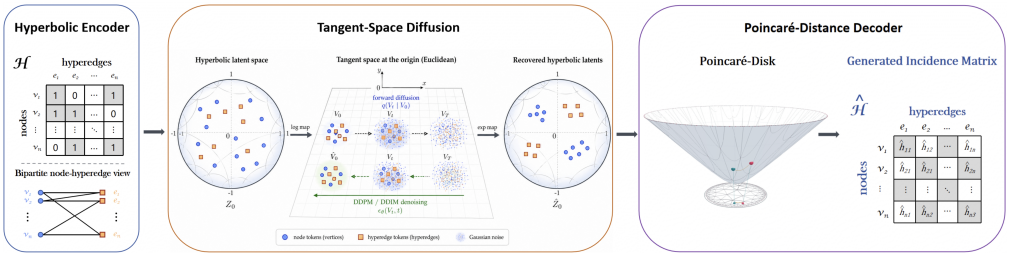


Figure 2: **Overall pipeline of HHGD.** An incidence matrix is encoded as hyperbolic node and hyperedge tokens, mapped to origin-tangent coordinates for diffusion, returned to the Poincaré ball, and decoded by validation-calibrated cardinality-aware per-edge top-k selection.

3.2 Hyperbolic Latent Space

Hyperbolic geometry provides an inductive bias for heterogeneous incidence structure, motivated by its compact representation of latent hierarchies (Nickel & Kiela, 2017). HHGD is a coordinate-based latent diffusion model rather than intrinsic manifold Brownian motion: ordered tokens are mapped into Lorentz coordinates, projected to a fixed Poincaré interface ($c_* = 1.0$), diffused via origin-tangent Gaussian noise, denoised with Poincaré mappings, and decoded via Poincaré distances. These conversions preserve slot order but are not isometric across curvatures. Appendix A gives full maps and numerical safeguards.

3.3 Incidence-Aware Tangent-Space Encoder with Lorentz Re-embedding

The encoder operates directly on biadjacency $H \in \{0,1\}^{N\times M}$, avoiding clique expansion (Feng et al., 2019; Huang & Yang, 2021; Chien et al., 2022; Li et al., 2023). Node and hyperedge features $U_V^{(0)} = f_V(H)$ and $U_E^{(0)} = f_E(H^\top)$ are lifted to Lorentz coordinates. At layer ℓ, tokens map to origin-tangent coordinates $V_\star^{(\ell)} = \mathrm{Log}_o^{K_\ell}(X_\star^{(\ell)})$ with fixed radius K_ℓ. Clamping degrees at $\varepsilon = 10^{-8}$, bidirectional mean aggregation is:

$$A_E^{(\ell)} = \mathrm{diag}(d_E)^{-1} H^\top \phi_V^{(\ell)}(V_V^{(\ell)}), \qquad A_V^{(\ell)} = \mathrm{diag}(d_V)^{-1} H \phi_E^{(\ell)}(V_E^{(\ell)}). \tag{2}$$

Residuals are LayerNormalized and re-embedded via $\mathrm{Exp}_o^{K_\ell}(\cdot)$, realizing tangent-space message passing rather than intrinsic Fréchet aggregation. A Euclidean Transformer globally mixes all tangent tokens (without key-padding masks) before type-specific heads map them to Z_0 (Vaswani et al., 2017; Peebles & Xie, 2023; Li et al., 2023). Crucially, scales K_ℓ are converted to Python scalars and receive no gradients. Appendix B details updates, tensor shapes, and complexity.

3.4 Tangent-Space Diffusion over Hyperbolic Latents

For $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, HHGD applies a standard DDPM to V_0 (Ho et al., 2020; Nichol & Dhariwal, 2021):

$$q(V_t \mid V_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t} V_0, (1 - \bar{\alpha}_t) I\big), \qquad V_t = \sqrt{\bar{\alpha}_t} V_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \tag{3}$$

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{H,t,\epsilon} \|\epsilon - \epsilon_\theta(V_t, t)\|_2^2, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{4}$$

During Phase II, the encoder and decoder are frozen, and only the denoiser parameters θ are optimized using $\mathcal{L}_{\text{diff}}$. Before denoising, V_t passes through the safeguarded round trip $\widetilde{V}_t = \log_0^{c_\star}(\text{Proj}_{c_\star}(\exp_0^{c_\star}(V_t)))$. Thus $\epsilon_\theta(V_t, t)$ abbreviates $\epsilon_\theta(\widetilde{V}_t, t; C)$, where the same fixed uniform context $C = M^{-1}\mathbf{1}_N\mathbf{1}_M^\top$ is used during training and generation. Sampling starts from $V_T \sim \mathcal{N}(0, I)$ and uses 50 deterministic DDIM steps ($\eta = 0$) (Song et al., 2021), clipping predicted clean-token norms to 3.0 and projecting each update back to the ball. Reverse equations and safeguards appear in Appendix C.

3.5 Poincaré-Space Denoising Network

The HHGD denoiser predicts $\widehat{\epsilon} = \epsilon_\theta(\widetilde{V}_t, t; C) \in \mathbb{R}^{(N+M)\times d}$. After an affine lift to the Poincaré ball, it alternates incidence-aware node–hyperedge aggregation with time-conditioned global attention and feed-forward blocks; tangent residuals and a final log/output map return the noise estimate. It uses fixed interface curvature $c_\star$ and is distinct from the Lorentz HGCN score ablation.

In the `uniform` configuration used in our experiments, the denoiser receives the exact same fixed structural context

$$C = M^{-1}\mathbf{1}_N\mathbf{1}_M^\top$$

during both training and unconditional sampling. Crucially, the target incidence matrix H is used exclusively as input to the encoder and to form the supervision loss; it is never passed to the denoiser. This design prevents structural leakage during training and guarantees complete train–inference consistency for unconditional generation.

3.6 Cardinality-Aware Poincaré Distance Decoder

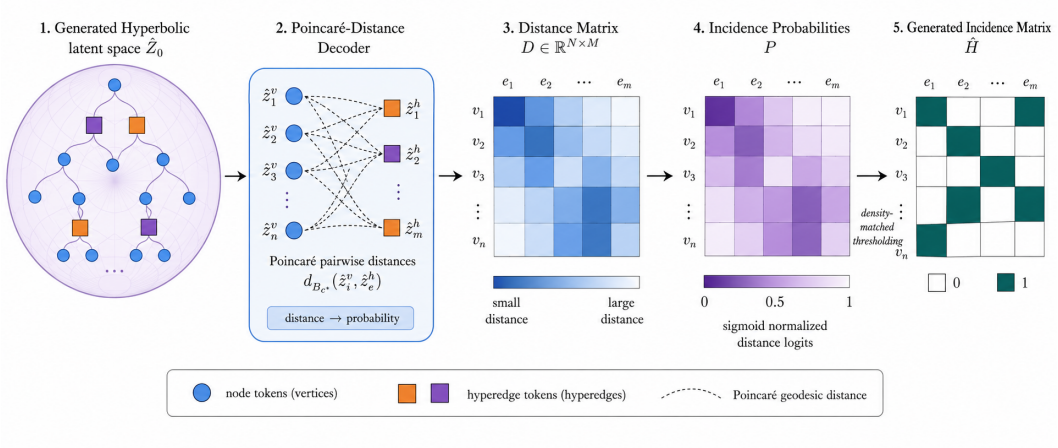


Figure 3: Overview of the HHGD decoder. Given generated Poincaré node and hyperedge tokens, the decoder computes normalized pairwise geodesic-distance scores to rank candidate memberships. A cardinality head predicts the size of each hyperedge slot, and validation-calibrated per-edge top-k selection converts the soft scores into a binary incidence matrix; repeated hyperedges are retained only once.

Figure 3 illustrates the decoding module. For generated tokens $\widehat{Z}_0 = [\widehat{Z}_V; \widehat{Z}_E]$, the decoder ranks memberships by normalized geodesic distance and predicts slot cardinalities:

$$r_{i,e} = -\frac{d_{\mathbb{B}_{c_\star}}(\hat{z}_i^v, \hat{z}_e^h) - \mu_d}{\tau \sigma_d}, \qquad \tau = \mathrm{softplus}(\omega) + 0.1, \tag{5}$$

$$a_e = g_\psi\left([\mathrm{Log}_0^{c_\star}(\hat{z}_e^h); \|\hat{z}_e^h\|_2]\right), \qquad \widehat{s}_e = N\sigma(a_e), \qquad P_{i,e} = \sigma(r_{i,e} + a_e). \tag{6}$$

The sample-wise statistics use $\sigma_d = \max(\mathrm{Std}[d], 10^{-6})$. Weighted BCE supervises P and cardinality loss supervises $\widehat{s}$, making P a continuous ranking score rather than an independent Bernoulli law. Validation-only calibration fixes γ and sets $k_e = \mathrm{clip}_{[0,N]}(\mathrm{round}(\gamma \widehat{s}_e))$. Slots with $k_e < 2$ stay inactive; otherwise the k_e largest $P_{:,e}$ entries are selected and duplicate hyperedges suppressed. Calibration is frozen before held-out evaluation. Appendix E gives the calibration criterion and complete discretization rule.

During Phase I, the encoder and decoder are jointly optimized using

$$\mathcal{L}_{\mathrm{AE}} = \mathcal{L}_{\mathrm{BCE}} + \lambda_{\mathrm{topo}}\mathcal{L}_{\mathrm{topo}} + \lambda_{\mathrm{boundary}}\mathcal{L}_{\mathrm{boundary}} + \lambda_{\mathrm{size}}\mathcal{L}_{\mathrm{size}}. \tag{7}$$

The positive-weighted BCE reconstructs incidence entries; the topology term matches normalized node degrees, hyperedge sizes, and incidence density; the boundary term penalizes token radii outside a prescribed interior band; and the size term supervises normalized cardinalities using Smooth L1. The nonnegative weights λ_{topo}, $\lambda_{\mathrm{boundary}}$, and λ_{size} balance these auxiliary losses against BCE. Their values and the full loss definitions are given in Appendix E.

Algorithm 1 summarizes the two-stage learning procedure, validation-only decoder calibration, and generation.

Algorithm 1 Two-Stage Training, Calibration, and Sampling of HHGD

Require: Training set $\mathcal{D}_{\mathrm{tr}}$, validation set $\mathcal{D}_{\mathrm{val}}$, fixed interface curvature $c_\star$, noise schedule $\{\bar{\alpha}_t\}_{t=1}^T$, calibration grid Γ

Phase I: incidence autoencoder training

for each minibatch $H \sim \mathcal{D}_{\mathrm{tr}}$ **do**
  $Z_0 \leftarrow \mathrm{Enc}_\phi(H)$
  $(P, \widehat{s}) \leftarrow \mathrm{Dec}_\psi(Z_0)$
  Update ϕ, ψ using $\mathcal{L}_{\mathrm{AE}}$
end for

Phase II: tangent-coordinate diffusion training

Freeze ϕ and ψ
Set $C \leftarrow M^{-1}\mathbf{1}_N\mathbf{1}_M^\top$
for each minibatch $H \sim \mathcal{D}_{\mathrm{tr}}$ **do**
  $Z_0 \leftarrow \mathrm{Enc}_\phi(H)$; $V_0 \leftarrow \log_0^{c_\star}(Z_0)$
  Sample t and $\epsilon \sim \mathcal{N}(0, I)$
  $V_t \leftarrow \sqrt{\bar{\alpha}_t}V_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$
  $\widetilde{V}_t \leftarrow \log_0^{c_\star}\left(\mathrm{Proj}_{c_\star}(\exp_0^{c_\star}(V_t))\right)$
  $\widehat{\epsilon} \leftarrow \epsilon_\theta(\widetilde{V}_t, t; C)$
  Update θ using $\mathcal{L}_{\mathrm{diff}}$
end for

Validation-only decoder calibration

Generate an independent soft calibration batch $(P^{\mathrm{cal}}, \widehat{s}^{\mathrm{cal}})$
$\gamma \leftarrow \arg\min_{g\in\Gamma} \mathrm{ValScore}(\mathcal{Q}_g(P^{\mathrm{cal}}, \widehat{s}^{\mathrm{cal}}), \mathcal{D}_{\mathrm{val}})$

Sampling

Set $C \leftarrow M^{-1}\mathbf{1}_N\mathbf{1}_M^\top$
Sample $V_T \sim \mathcal{N}(0, I)$; $Z_T \leftarrow \mathrm{Proj}_{c_\star}(\exp_0^{c_\star}(V_T))$
$\widehat{Z}_0 \leftarrow \mathrm{DDIMSample}(Z_T, \epsilon_\theta(\cdot,\cdot;C), \eta = 0)$
$(P, \widehat{s}) \leftarrow \mathrm{Dec}_\psi(\widehat{Z}_0)$
$\widehat{H} \leftarrow \mathcal{Q}_\gamma(P, \widehat{s})$
return $\widehat{H}$

4 EXPERIMENTS

4.1 DATASETS AND PREPROCESSING

We evaluate HHGD on eight datasets spanning synthetic and real-world regimes: Erdős–Rényi (ER), hypergraph SBM, Tree, Ego, Contact Primary School, NDC Substances, Email-Eu, and Tags-Math (Erdős & Rényi, 1960; Kim et al., 2018; Nieminen & Peltola, 1999; Comrie & Kleinberg, 2021; Benson et al., 2018). Each dataset contains 350 training, 50 validation, and 100 held-out test hypergraphs. For the temporal real-world sources, events are partitioned chronologically before local hypergraph construction; duplicate incidence matrices are rejected across splits, and test data are reserved for final evaluation. Appendix F gives the complete construction, padding, and leakage-control protocol, and Table 5 reports the fixed slot budgets.

4.2 TRAINING AND SAMPLING SETTINGS

We train the autoencoder for 50 epochs and the diffusion prior for 100 epochs. Checkpoints and decoding calibration are fixed using validation data only. Evaluation uses 50-step DDIM and 100 generated samples per run; we report mean $\pm$ sample standard deviation over three training–generation seed pairs. Architecture, optimizer settings, and calibration details are provided in Appendix F; reconstruction-loss weights are specified in Appendix E.

4.3 BASELINES AND ABLATIONS

We conduct a 2×2 architecture ablation that varies the encoder and score network while keeping the cardinality-aware Poincaré decoder, data splits, training budget, validation protocol, and three seed pairs fixed. The displayed encoders are HGCN and Euclidean; the displayed score networks are Hyperbolic and Lightweight. We use **HHGD** for HGCN+Hyperbolic. The four displayed configurations are summarized in Table 2; because the decoder is shared, this study isolates component choices rather than comparing fully Euclidean and hyperbolic pipelines.

We compare HHGD with six external baselines: HyperPA, HyperLap+, HYGENE, and the VAE, GAN, and incidence-image diffusion adaptations released with HYGENE. All executable methods use the same frozen splits, evaluator, and three fitting/training–generation seed pairs. HyperLap+ fits templates from training data only, and HYGENE on Email-Eu is reported as OOM after exceeding the 24 GB accelerator limit at batch size one; no incomplete output is substituted.

4.4 EVALUATION METRICS

We evaluate structural fidelity by comparing generated and held-out hypergraph distributions after unpadding (Do et al., 2020; Gailhard et al., 2025; Lee et al., 2025). We report 1-Wasserstein (W_1) distances for hyperedge counts, pooled node degrees, hyperedge sizes, pair co-degrees, and intersection sizes. Spectral fidelity is measured via biased squared MMD over normalized-Laplacian eigenvalue histograms with a Gaussian kernel on total-variation distance. Lower values indicate better alignment; additional W_1 diagnostics are reported in Appendix F.

We also report validity, uniqueness, and novelty ($\uparrow$). Validity requires a non-empty binary matrix with the prescribed active-node count and no empty, singleton, or duplicate hyperedges. Uniqueness and novelty rely on exact isomorphism testing on typed bipartite graphs, rendering them permutation-invariant. All metrics are evaluated across all 100 generated samples per run over three fixed seed pairs, reporting mean and sample standard deviation.

4.5 ABLATION RESULTS

Table 2 identifies the four configurations compared below. Comparing columns within a row isolates the score-network choice; comparing rows within a column isolates the encoder choice, with the decoder unchanged.

Tables 3 and 7 show complementary effects of the encoder–score architecture across the four real-world and four synthetic datasets. Within the displayed HGCN- and Euclidean-encoder configurations, the Hyperbolic score network achieves lower mean hyperedge-size W_1 than the corresponding

Table 2: Architectural configurations in the 2×2 ablation study. All variants utilize the identical cardinality-aware Poincaré decoder.

Encoder	Proposed Score Net	Lightweight Score Net
HGCN (Hyperbolic)	**HHGD (Ours)**	HGCN + Light
Euclidean	Euc + Hyperbolic	Euc + Light

Table 3: Encoder–score ablation on four real-world held-out datasets; synthetic results appear in Table 7. Entries are three-seed means with $\pm$ sample standard deviations; each run generates 100 samples. Bold marks the best displayed mean, including ties, without implying statistical significance.

Encoder+Score	Valid (%)↑	EdgeCnt↓	NodeDeg↓	EdgeSize↓	PairDeg↓	Intersect↓	Spectral↓
Contact Primary School							
HHGD	$1.7_{\pm 1.2}$	$\mathbf{8.78}_{\pm 0.63}$	$2.079_{\pm 0.007}$	$0.931_{\pm 0.044}$	$\mathbf{0.232}_{\pm 0.090}$	$\mathbf{0.047}_{\pm 0.018}$	$0.510_{\pm 0.007}$
Euc+Hyp	$\mathbf{2.7}_{\pm 2.1}$	$8.92_{\pm 0.80}$	$2.127_{\pm 0.099}$	$\mathbf{0.908}_{\pm 0.190}$	$0.514_{\pm 0.580}$	$0.069_{\pm 0.084}$	$0.518_{\pm 0.022}$
HGCN+Light	$0.7_{\pm 0.6}$	$9.15_{\pm 3.07}$	$\mathbf{1.680}_{\pm 0.100}$	$1.288_{\pm 0.011}$	$0.749_{\pm 0.000}$	$0.066_{\pm 0.000}$	$0.410_{\pm 0.015}$
Euc+Light	$1.0_{\pm 0.0}$	$13.09_{\pm 8.10}$	$1.729_{\pm 0.120}$	$1.284_{\pm 0.014}$	$0.749_{\pm 0.000}$	$0.066_{\pm 0.000}$	$\mathbf{0.399}_{\pm 0.024}$
NDC Substances							
HHGD	$\mathbf{31.7}_{\pm 1.5}$	$14.10_{\pm 1.20}$	$2.823_{\pm 0.197}$	$\mathbf{3.289}_{\pm 0.063}$	$0.497_{\pm 0.196}$	$\mathbf{0.826}_{\pm 0.058}$	$0.101_{\pm 0.005}$
Euc+Hyp	$31.3_{\pm 1.5}$	$13.99_{\pm 0.71}$	$\mathbf{2.805}_{\pm 0.122}$	$3.324_{\pm 0.089}$	$\mathbf{0.442}_{\pm 0.168}$	$0.850_{\pm 0.069}$	$0.099_{\pm 0.004}$
HGCN+Light	$31.3_{\pm 0.6}$	$8.75_{\pm 1.08}$	$2.870_{\pm 0.056}$	$3.508_{\pm 0.019}$	$0.659_{\pm 0.007}$	$0.973_{\pm 0.003}$	$\mathbf{0.080}_{\pm 0.002}$
Euc+Light	$31.0_{\pm 1.0}$	$\mathbf{8.68}_{\pm 1.02}$	$2.860_{\pm 0.086}$	$3.502_{\pm 0.027}$	$0.654_{\pm 0.005}$	$0.971_{\pm 0.002}$	$\mathbf{0.080}_{\pm 0.002}$
Email-Eu							
HHGD	$36.0_{\pm 1.0}$	$\mathbf{18.21}_{\pm 0.52}$	$2.745_{\pm 0.320}$	$\mathbf{2.031}_{\pm 0.200}$	$0.676_{\pm 0.331}$	$0.197_{\pm 0.004}$	$0.236_{\pm 0.007}$
Euc+Hyp	$\mathbf{36.3}_{\pm 2.1}$	$19.96_{\pm 0.50}$	$2.894_{\pm 0.119}$	$2.104_{\pm 0.071}$	$\mathbf{0.531}_{\pm 0.371}$	$\mathbf{0.171}_{\pm 0.020}$	$0.238_{\pm 0.014}$
HGCN+Light	$33.7_{\pm 1.2}$	$38.05_{\pm 20.40}$	$\mathbf{2.631}_{\pm 0.675}$	$2.641_{\pm 0.159}$	$1.705_{\pm 0.113}$	$0.436_{\pm 0.021}$	$\mathbf{0.124}_{\pm 0.000}$
Euc+Light	$33.7_{\pm 0.6}$	$38.05_{\pm 21.73}$	$2.691_{\pm 0.814}$	$2.659_{\pm 0.146}$	$1.721_{\pm 0.109}$	$0.440_{\pm 0.020}$	$0.129_{\pm 0.003}$
Tags-Math							
HHGD	$\mathbf{89.0}_{\pm 0.0}$	$22.51_{\pm 2.30}$	$\mathbf{3.414}_{\pm 0.081}$	$0.380_{\pm 0.061}$	$\mathbf{0.623}_{\pm 0.057}$	$0.063_{\pm 0.029}$	$0.037_{\pm 0.012}$
Euc+Hyp	$\mathbf{89.0}_{\pm 0.0}$	$20.76_{\pm 1.85}$	$3.475_{\pm 0.087}$	$\mathbf{0.327}_{\pm 0.036}$	$0.637_{\pm 0.047}$	$\mathbf{0.053}_{\pm 0.023}$	$0.034_{\pm 0.008}$
HGCN+Light	$\mathbf{89.0}_{\pm 0.0}$	$17.25_{\pm 0.40}$	$3.729_{\pm 0.150}$	$0.521_{\pm 0.093}$	$0.949_{\pm 0.086}$	$0.183_{\pm 0.036}$	$0.021_{\pm 0.001}$
Euc+Light	$\mathbf{89.0}_{\pm 0.0}$	$\mathbf{17.02}_{\pm 0.15}$	$3.702_{\pm 0.175}$	$0.473_{\pm 0.032}$	$0.900_{\pm 0.033}$	$0.168_{\pm 0.017}$	$\mathbf{0.020}_{\pm 0.002}$

Lightweight score on seven of eight datasets, with Tree as the exception. For the four real-world datasets shown here, novelty and uniqueness are both 1.000 ± 0.000 for all displayed configurations.Low strict validity on several real-world datasets also remains a limitation despite high uniqueness and novelty.

4.6 Comparison with External Baselines

Table 4 evaluates performance on ER, Tree, Contact and Tags-Math, where HHGD achieves superior strict validity. Table 6 presents the complementary results on the remaining four datasets under the same protocol. Overall, HHGD shows strong hyperedge structural fidelity and validity, notably achieving the lowest mean edge-count W_1 on Contact Primary School while maintaining robust incidence modeling across varied hypergraph topologies.

Table 4: External-baseline comparison on ER, Tree, Contact, and Tags-Math datasets (means ± SD over three seeds, 100 samples/run). Bold indicates the best mean; higher validity/uniqueness (↑) and lower structural distances (↓) are preferred. Novelty is 1.000 ± 0.000 across all completed runs and is omitted.

Method	Valid (%)↑	Unique (%)↑	EdgeCnt↓	NodeDeg↓	EdgeSize↓	PairDeg↓	Intersect↓	Spectral↓
Erdős–Rényi (ER)								
HyperPA	$0.0_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$4.95_{\pm 0.57}$	$4.506_{\pm 0.074}$	$0.021_{\pm 0.013}$	$0.936_{\pm 0.018}$	$0.134_{\pm 0.002}$	$0.053_{\pm 0.003}$
HyperLap+	$4.3_{\pm 1.5}$	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{1.95}_{\pm 0.24}$	$0.695_{\pm 0.041}$	$\mathbf{0.005}_{\pm 0.001}$	$\mathbf{0.040}_{\pm 0.002}$	$\mathbf{0.006}_{\pm 0.002}$	$\mathbf{0.003}_{\pm 0.001}$
HYGENE	$6.0_{\pm 5.3}$	$88.7_{\pm 10.0}$	$48.27_{\pm 5.63}$	$3.533_{\pm 1.224}$	$0.588_{\pm 0.071}$	$0.356_{\pm 0.243}$	$0.060_{\pm 0.030}$	$0.079_{\pm 0.051}$
VAE	$0.0_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$6.81_{\pm 0.26}$	$0.501_{\pm 0.038}$	$0.492_{\pm 0.008}$	$0.137_{\pm 0.005}$	$0.061_{\pm 0.003}$	$0.008_{\pm 0.000}$
GAN	$0.0_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$3.22_{\pm 1.31}$	$\mathbf{0.293}_{\pm 0.053}$	$0.545_{\pm 0.030}$	$0.133_{\pm 0.015}$	$0.056_{\pm 0.004}$	$0.006_{\pm 0.001}$
Inc. Diff.	$0.3_{\pm 0.6}$	$92.0_{\pm 13.9}$	$15.04_{\pm 9.04}$	$1.942_{\pm 0.623}$	$0.747_{\pm 0.126}$	$0.522_{\pm 0.118}$	$0.151_{\pm 0.026}$	$0.032_{\pm 0.031}$
HHGD	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$16.52_{\pm 6.01}$	$2.806_{\pm 0.247}$	$0.179_{\pm 0.067}$	$0.400_{\pm 0.103}$	$0.086_{\pm 0.026}$	$0.046_{\pm 0.012}$
Tree								
HyperPA	$28.3_{\pm 1.2}$	$\mathbf{100.0}_{\pm 0.0}$	$3.32_{\pm 0.65}$	$0.962_{\pm 0.008}$	$0.006_{\pm 0.004}$	$0.123_{\pm 0.008}$	$0.038_{\pm 0.002}$	$0.096_{\pm 0.006}$
HyperLap+	$62.0_{\pm 4.4}$	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{0.14}_{\pm 0.01}$	$0.487_{\pm 0.006}$	$\mathbf{0.005}_{\pm 0.001}$	$\mathbf{0.020}_{\pm 0.002}$	$\mathbf{0.011}_{\pm 0.001}$	$\mathbf{0.053}_{\pm 0.002}$
HYGENE	$0.0_{\pm 0.0}$	$85.0_{\pm 12.1}$	$72.28_{\pm 39.71}$	$2.574_{\pm 2.026}$	$0.802_{\pm 0.039}$	$0.145_{\pm 0.101}$	$0.020_{\pm 0.003}$	$0.238_{\pm 0.019}$
VAE	$1.0_{\pm 1.0}$	$\mathbf{100.0}_{\pm 0.0}$	$1.79_{\pm 0.36}$	$0.442_{\pm 0.101}$	$0.878_{\pm 0.053}$	$0.067_{\pm 0.008}$	$0.074_{\pm 0.008}$	$0.064_{\pm 0.002}$
GAN	$0.0_{\pm 0.0}$	$94.7_{\pm 8.4}$	$2.90_{\pm 1.26}$	$\mathbf{0.372}_{\pm 0.159}$	$0.886_{\pm 0.076}$	$0.062_{\pm 0.019}$	$0.065_{\pm 0.018}$	$0.067_{\pm 0.002}$
Inc. Diff.	$1.7_{\pm 2.1}$	$\mathbf{100.0}_{\pm 0.0}$	$2.79_{\pm 1.86}$	$0.846_{\pm 0.371}$	$1.270_{\pm 0.242}$	$0.177_{\pm 0.058}$	$0.189_{\pm 0.058}$	$0.085_{\pm 0.002}$
HHGD	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$1.80_{\pm 0.67}$	$0.601_{\pm 0.072}$	$0.181_{\pm 0.025}$	$0.126_{\pm 0.018}$	$0.075_{\pm 0.011}$	$0.083_{\pm 0.021}$
Contact Primary School								
HyperPA	$\mathbf{14.0}_{\pm 5.3}$	$\mathbf{100.0}_{\pm 0.0}$	$16.82_{\pm 3.51}$	$1.951_{\pm 0.156}$	$\mathbf{0.049}_{\pm 0.006}$	$0.711_{\pm 0.246}$	$0.148_{\pm 0.016}$	$0.365_{\pm 0.016}$
HyperLap+	$0.0_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$25.74_{\pm 2.05}$	$\mathbf{1.210}_{\pm 0.046}$	$0.571_{\pm 0.065}$	$\mathbf{0.091}_{\pm 0.026}$	$\mathbf{0.007}_{\pm 0.003}$	$0.259_{\pm 0.014}$
HYGENE	$10.7_{\pm 18.5}$	$91.0_{\pm 8.2}$	$180.70_{\pm 134.49}$	$24.584_{\pm 21.027}$	$1.870_{\pm 1.815}$	$3.011_{\pm 0.359}$	$2.833_{\pm 2.579}$	$0.284_{\pm 0.053}$
VAE	$0.0_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$16.41_{\pm 0.80}$	$6.585_{\pm 0.438}$	$0.799_{\pm 0.083}$	$2.093_{\pm 0.622}$	$0.362_{\pm 0.136}$	$\mathbf{0.200}_{\pm 0.004}$
GAN	$0.0_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$32.50_{\pm 7.60}$	$4.577_{\pm 0.442}$	$0.730_{\pm 0.097}$	$0.221_{\pm 0.061}$	$0.103_{\pm 0.024}$	$0.239_{\pm 0.003}$
Inc. Diff.	$0.3_{\pm 0.6}$	$\mathbf{100.0}_{\pm 0.0}$	$36.99_{\pm 5.68}$	$5.030_{\pm 1.119}$	$0.849_{\pm 0.154}$	$0.401_{\pm 0.075}$	$0.059_{\pm 0.030}$	$0.238_{\pm 0.003}$
HHGD	$1.7_{\pm 1.2}$	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{8.78}_{\pm 0.63}$	$2.079_{\pm 0.007}$	$0.931_{\pm 0.044}$	$0.232_{\pm 0.090}$	$0.047_{\pm 0.018}$	$0.510_{\pm 0.007}$
Tags-Math								
HyperPA	$5.7_{\pm 2.1}$	$\mathbf{100.0}_{\pm 0.0}$	$31.53_{\pm 2.82}$	$2.923_{\pm 0.195}$	$\mathbf{0.099}_{\pm 0.013}$	$1.773_{\pm 0.189}$	$0.167_{\pm 0.012}$	$0.190_{\pm 0.006}$
HyperLap+	$0.3_{\pm 0.6}$	$\mathbf{100.0}_{\pm 0.0}$	$13.21_{\pm 0.55}$	$1.698_{\pm 0.063}$	$0.165_{\pm 0.024}$	$0.321_{\pm 0.031}$	$\mathbf{0.024}_{\pm 0.013}$	$\mathbf{0.015}_{\pm 0.001}$
HYGENE	$0.0_{\pm 0.0}$	$96.0_{\pm 5.2}$	$58.56_{\pm 18.77}$	$3.314_{\pm 1.946}$	$0.894_{\pm 0.217}$	$0.577_{\pm 0.196}$	$0.107_{\pm 0.034}$	$0.135_{\pm 0.039}$
VAE	$0.7_{\pm 0.6}$	$\mathbf{100.0}_{\pm 0.0}$	$12.91_{\pm 0.30}$	$3.215_{\pm 0.223}$	$0.712_{\pm 0.048}$	$0.677_{\pm 0.028}$	$0.086_{\pm 0.010}$	$0.045_{\pm 0.003}$
GAN	$0.3_{\pm 0.6}$	$\mathbf{100.0}_{\pm 0.0}$	$14.04_{\pm 1.17}$	$\mathbf{1.631}_{\pm 0.196}$	$0.649_{\pm 0.074}$	$\mathbf{0.191}_{\pm 0.133}$	$0.042_{\pm 0.013}$	$0.045_{\pm 0.001}$
Inc. Diff.	$10.3_{\pm 15.3}$	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{12.64}_{\pm 5.54}$	$15.484_{\pm 19.271}$	$6.340_{\pm 9.441}$	$22.795_{\pm 37.643}$	$5.238_{\pm 8.800}$	$0.067_{\pm 0.034}$
HHGD	$\mathbf{89.0}_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$22.51_{\pm 2.30}$	$3.414_{\pm 0.081}$	$0.380_{\pm 0.061}$	$0.623_{\pm 0.057}$	$0.063_{\pm 0.029}$	$0.037_{\pm 0.012}$

4.7 Discussion

The evaluation indicates selective rather than uniform advantages: HHGD excels in strict validity and hyperedge-size fidelity on several datasets, whereas lightweight denoising often achieves better spectral alignment, with encoder impacts remaining dataset-dependent. Crucially, these distributional metrics do not directly confirm hierarchy recovery. Validity reflects the end-to-end generation and decoding pipeline. The frozen decoder maps validation-calibrated cardinalities into top-k memberships without using test labels or target incidence matrices, thereby serving as part of the architecture rather than post-hoc evaluation repair. A key limitation remains the use of the target-independent uniform context C, which guarantees generation consistency but may constrain the model's ability to capture fine-grained, sample-specific structural dependencies.

5 Conclusion

We propose **HHGD**, which combines native node–hyperedge tokens, hyperbolic representation and denoising, tangent-coordinate diffusion, and cardinality-aware Poincaré-distance decoding. Across synthetic and real-world datasets, HHGD attains the highest reported mean strict validity on Erdős–Rényi, Tree, and Tags-Math, while its validity varies substantially across the real-world datasets. Future work should develop a richer permutation-equivariant context mechanismfor modeling sample-specific incidence dependencies while preserving matched training and generation conditions.

AI Use Statement

During preparation of this manuscript, the authors used generative AI tools for LaTeX formatting, literature organization, language editing, and condensing the manuscript. The tools were also used to assist with reviewing and reviewing generated results; all such interpretations were independently checked against the stored result artifacts and evaluation scripts. The authors did not use generative

AI tools to generate synthetic datasets, design the methodology, formulate mathematical claims, implement the experimental code, alter reported data, or fabricate results. All AI-assisted material was reviewed by the authors, who take responsibility for the final text, claims, analyses, code, and artifacts.

REFERENCES

Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon Kleinberg. Simplicial closure and higher-order link prediction. *Proceedings of the National Academy of Sciences*, 115 (48):E11221–E11230, 2018.

Ines Chami, Rex Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Weize Chen, Xu Han, Yankai Lin, Hexu Zhao, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Fully hyperbolic neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 5672–5686, 2022.

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

Jaewan Chun, Seokbum Yoon, Minyoung Choe, Geon Lee, and Kijung Shin. Attributed hypergraph generation with realistic interplay between structure and attributes. In *2025 IEEE International Conference on Data Mining*, 2025.

Cazamere Comrie and Jon Kleinberg. Hypergraph ego-networks and their temporal evolution. In *2021 IEEE International Conference on Data Mining*, pp. 91–100, 2021.

Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia. A hyperbolic-to-hyperbolic graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 154–163, 2021.

Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. Riemannian score-based generative modelling. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Manh Tuan Do, Se-eun Yoon, Bryan Hooi, and Kijung Shin. Structural patterns and generative models of real-world hypergraphs. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 176–186, 2020.

Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

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.

Xingcheng Fu, Yisen Gao, Yuecen Wei, Qingyun Sun, Hao Peng, Jianxin Li, and Xianxian Li. Hyperbolic geometric latent diffusion model for graph generation. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Dorian Gailhard, Enzo Tartaglione, Lirida Naviner, and Jhony H. Giraldo. HYGENE: A diffusion-based hypergraph generation method. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(16):16682–16690, 2025.

Octavian-Eugen Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Han Huang, Leilei Sun, Bowen Du, Yanjie Fu, and Weifeng Lv. GraphGDP: Generative diffusion processes for permutation invariant graph generation. In *2022 IEEE International Conference on Data Mining*, pp. 201–210, 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*, pp. 2563–2569, 2021.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10362–10383. PMLR, 2022.

Chiheon Kim, Afonso S. Bandeira, and Michel X. Goemans. Stochastic block model for hypergraphs: Statistical limits and a semidefinite programming approach, 2018.

Geon Lee, Fanchen Bu, Tina Eliassi-Rad, and Kijung Shin. A survey on hypergraph mining: Patterns, tools, and generators. *ACM Computing Surveys*, 57(8):203:1–203:36, 2025.

Mengran Li, Yong Zhang, Xiaoyong Li, Yuchen Zhang, and Baocai Yin. Hypergraph transformer neural networks. *ACM Transactions on Knowledge Discovery from Data*, 17(5):63:1–63:22, 2023.

Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8230–8241, 2019.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 2021.

Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

J. Nieminen and M. Peltola. Hypertrees. *Applied Mathematics Letters*, 12(2):35–38, 1999.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

John G. Ratcliffe. *Foundations of Hyperbolic Manifolds*. Springer, 2 edition, 2006.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. DiGress: Discrete denoising diffusion for graph generation. In *International Conference on Learning Representations*, 2023.

Lingfeng Wen, Xuan Tang, Mingjie Ouyang, Xiangxiang Shen, Jian Yang, Daxin Zhu, Mingsong Chen, and Xian Wei. Hyperbolic graph diffusion model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(14):15823–15831, 2024.

Weihuang Wen and Tianshu Yu. HyperPLR: Hypergraph generation through projection, learning, and reconstruction. In *International Conference on Learning Representations*, 2025.

Naganand Yadati, Madhav Nimishakavi, Prateek Yadav, Vikram Nitin, Anand Louis, and Partha Talukdar. HyperGCN: A new method for training graph convolutional networks on hypergraphs. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. GraphRNN: Generating realistic graphs with deep auto-regressive models. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5708–5717. PMLR, 2018.

Xianglin Zuo, Wenqi Chen, Xianduo Song, Xin Wang, and Ying Wang. Generating real-world hypergraphs via deep generative models. *Information Sciences*, 647:119412, 2023.

A Hyperbolic Geometry Details

A.1 Curvature Convention

HHGD uses two curvature-related conventions. For the final Poincaré latent space, we use a curvature magnitude $c_\star > 0$ and write

$$\mathbb{B}^d_{c_\star} = \{z \in \mathbb{R}^d : c_\star \|z\|_2^2 < 1\}. \tag{A.1}$$

All final latent tokens, diffusion logarithmic and exponential maps, and decoder distances use the same $c_\star$.

For Lorentz or hyperboloid representations used inside the encoder and denoising network, we use a radius parameter $K > 0$:

$$\mathbb{L}^d_K = \{x \in \mathbb{R}^{d+1} : \langle x, x\rangle_{\mathcal{L}} = -K,\ x_0 > 0\}. \tag{A.2}$$

The sectional curvature of $\mathbb{L}^d_K$ is $-1/K$. When the Lorentz and Poincaré models represent the same hyperbolic geometry, their parameters satisfy $c = 1/K$. In HHGD, Lorentz coordinates are used for numerically stable internal re-embedding in stacked neural layers, while the Poincaré ball is used for the final latent interface, tangent-space diffusion, and Poincaré-distance decoding.

A.2 Poincaré Ball Operations

The d-dimensional Poincaré ball with curvature magnitude $c > 0$ is

$$\mathbb{B}^d_c = \{z \in \mathbb{R}^d : c\|z\|_2^2 < 1\}. \tag{A.3}$$

At a point $z \in \mathbb{B}^d_c$, the conformal factor is

$$\lambda^c_z = \frac{2}{1 - c\|z\|_2^2}. \tag{A.4}$$

The origin exponential map used by HHGD is

$$\exp_0^c(v) = \tanh\left(\frac{\sqrt{c}\|v\|_2}{2}\right)\frac{v}{\sqrt{c}\|v\|_2}, \tag{A.5}$$

with the continuous extension $\exp_0^c(0) = 0$. The corresponding origin logarithmic map is

$$\log_0^c(z) = \frac{2}{\sqrt{c}}\operatorname{arctanh}\left(\sqrt{c}\|z\|_2\right)\frac{z}{\|z\|_2}, \tag{A.6}$$

with $\log_0^c(0) = 0$.

For $x, y \in \mathbb{B}^d_c$, Möbius addition is

$$x \oplus_c y = \frac{(1 + 2c\langle x, y\rangle + c\|y\|_2^2)x + (1 - c\|x\|_2^2)y}{1 + 2c\langle x, y\rangle + c^2\|x\|_2^2\|y\|_2^2}. \tag{A.7}$$

The Poincaré geodesic distance is

$$d_{\mathbb{B}_c}(x, y) = \frac{2}{\sqrt{c}}\operatorname{arctanh}\left(\sqrt{c}\|(-x) \oplus_c y\|_2\right). \tag{A.8}$$

The decoder in HHGD uses this distance with $c = c_\star$ to score node–hyperedge membership probabilities.

A.3 LORENTZ MODEL OPERATIONS

The Lorentz model with radius parameter $K > 0$ is

$$\mathbb{L}_K^d = \{x \in \mathbb{R}^{d+1} : \langle x, x\rangle_{\mathcal{L}} = -K,\ x_0 > 0\}, \tag{A.9}$$

where the Minkowski inner product is

$$\langle x, y\rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{j=1}^{d} x_j y_j. \tag{A.10}$$

The origin is

$$o_K = (\sqrt{K}, 0, \ldots, 0), \tag{A.11}$$

and the tangent space at the origin can be identified with

$$T_{o_K}\mathbb{L}_K^d = \{(0, u) : u \in \mathbb{R}^d\} \cong \mathbb{R}^d. \tag{A.12}$$

For $u \in \mathbb{R}^d$, the origin exponential map is

$$\begin{aligned}\operatorname{Exp}_o^K(u) = \Big(&\sqrt{K}\cosh(\|u\|_2/\sqrt{K}),\\ &\sqrt{K}\sinh(\|u\|_2/\sqrt{K})\frac{u}{\|u\|_2}\Big),\end{aligned} \tag{A.13}$$

with the continuous extension at $u = 0$. For $x = (x_0, x_{1:d}) \in \mathbb{L}_K^d$, the origin logarithmic map returns the spatial tangent coordinates

$$\operatorname{Log}_o^K(x) = \frac{\operatorname{arcosh}(x_0/\sqrt{K})}{\sinh(\operatorname{arcosh}(x_0/\sqrt{K}))} x_{1:d}. \tag{A.14}$$

The Lorentz geodesic distance is

$$d_{\mathbb{L}_K}(x, y) = \sqrt{K}\operatorname{arcosh}\left(-\frac{\langle x, y\rangle_{\mathcal{L}}}{K}\right). \tag{A.15}$$

A.4 LORENTZ-TO-POINCARÉ FINAL PROJECTION

For a Lorentz point $x = (x_0, x_{1:d}) \in \mathbb{L}_K^d$, HHGD uses the final coordinate projection

$$\Pi_{K\to c_\star}(x) = \frac{1}{\sqrt{c_\star}} \frac{x_{1:d}}{x_0 + \sqrt{K}}. \tag{A.16}$$

When $c_\star = 1/K$, this is the standard coordinate conversion between the Lorentz and Poincaré models of the same hyperbolic space. When $c_\star$ is chosen independently, $\Pi_{K\to c_\star}$ is used only as a final bounded coordinate projection; we do not claim that it is an isometry between spaces of different curvatures.

In the implementation, `hyperboloid.to_poincare` corresponds to the special case

$$\Pi_{K\to 1}(x) = \frac{x_{1:d}}{x_0 + \sqrt{K}}. \tag{A.17}$$

The resulting point is then treated as a Poincaré coordinate used by the final latent interface, tangent-space diffusion, and decoder.

A.5 NUMERICAL LORENTZ SCALE PARAMETERIZATION

Each HGCN convolution layer stores a positive Lorentz scale through

$$K_\ell = \exp(\rho_\ell), \qquad K_\ell > 0, \tag{A.18}$$

which corresponds to sectional curvature $-1/K_\ell$. In the present implementation, K_ℓ is treated as a fixed numerical scale during optimization; thus, the reported results do not rely on adaptive curvature learning. A future extension could make these layer-wise scales learnable by preserving their differentiable dependence throughout the Lorentz operations. Evaluating whether such adaptive geometry improves generation quality is left for future work.

A.6 Numerical Projection and Stabilization

All geometric operations use standard numerical safeguards. For Poincaré points, we project to the interior of the ball:

$$\mathrm{Proj}_{\mathbb{B}_c}(z) = \begin{cases} z, & \sqrt{c}\|z\|_2 < 1-\varepsilon, \\ \dfrac{1-\varepsilon}{\sqrt{c}\|z\|_2} z, & \text{otherwise.} \end{cases} \tag{A.19}$$

For Lorentz points, we project by recomputing the time coordinate from the spatial coordinates:

$$\mathrm{Proj}_{\mathbb{L}_K}(x) = \left(\sqrt{K + \|x_{1:d}\|_2^2}, x_{1:d} \right). \tag{A.20}$$

We clamp the input of $\operatorname{arctanh}$ to $(-1+\epsilon, 1-\epsilon)$ and the input of arcosh to $[1+\epsilon, \infty)$. Norms in exp/log maps are clamped from below, and the maps use their continuous extensions at the origin. These stabilizations match the numerical role of `EPS`, `MIN_NORM`, and ball-boundary projection in the implementation.

B Encoder Details

B.1 Batched Tensor Shapes

For a mini-batch of hypergraphs, the incidence tensor is

$$H \in \{0,1\}^{B\times N\times M}, \tag{B.1}$$

where B is the batch size, N is the number of node slots, and M is the number of hyperedge slots. The initial Euclidean node and hyperedge features are

$$U_V^{(0)} \in \mathbb{R}^{B\times N\times d}, \qquad U_E^{(0)} \in \mathbb{R}^{B\times M\times d}. \tag{B.2}$$

After origin exponential mapping to the Lorentz model, the corresponding manifold tokens have one additional time coordinate:

$$X_V^{(\ell)} \in \mathbb{R}^{B\times N\times (d+1)}, \qquad X_E^{(\ell)} \in \mathbb{R}^{B\times M\times (d+1)}. \tag{B.3}$$

The origin logarithmic map returns spatial tangent coordinates:

$$V_V^{(\ell)} \in \mathbb{R}^{B\times N\times d}, \qquad V_E^{(\ell)} \in \mathbb{R}^{B\times M\times d}. \tag{B.4}$$

After L encoder layers and global token mixing, the final Poincaré latent token matrix is represented by ball-valued coordinates

$$Z_0 = [Z_V; Z_E] \in (\mathbb{B}_{c_\star}^d)^{B\times (N+M)} \subset \mathbb{R}^{B\times (N+M)\times d}. \tag{B.5}$$

B.2 Incidence Normalization and Masking

For each hypergraph in the batch, define the stabilized degree matrices

$$\bar{D}_V = \mathrm{diag}(\max(H\mathbf{1}_M, \varepsilon)), \qquad \bar{D}_E = \mathrm{diag}\big(\max(H^\top \mathbf{1}_N, \varepsilon)\big), \tag{B.6}$$

where the maximum is applied elementwise and $\varepsilon = 10^{-8}$ prevents division by zero for isolated nodes or empty hyperedge slots. This formulation corresponds to the elementwise lower-bound clamping used in the evaluated encoder. The normalized incidence aggregations are

$$\bar{D}_E^{-1} H^\top \quad \text{and} \quad \bar{D}_V^{-1} H. \tag{B.7}$$

For fixed-size datasets without padding, all node and hyperedge slots are valid, rendering masks redundant. For padded variable-size data, masks delineate active entities and can theoretically support mask-aware aggregation, losses, and evaluation. However, in the evaluated fixed-slot path, node and hyperedge masks are not passed to the encoder aggregation, Transformer mixing, reconstruction loss, or diffusion loss; instead, the full padded tensor is processed. Masks are utilized exclusively by the dataset loader, validator, and evaluator to identify active entities and enforce output validity. Thus, padding serves as part of the fixed-width representation rather than as a conditioning signal for generation.

B.3 Full Encoder Layer

At layer ℓ, the encoder first maps Lorentz tokens to tangent coordinates:

$$V_V^{(\ell)} = \mathrm{Log}_o^{K_\ell}(X_V^{(\ell)}), \qquad V_E^{(\ell)} = \mathrm{Log}_o^{K_\ell}(X_E^{(\ell)}). \tag{B.8}$$

Message functions are applied in tangent space:

$$M_V^{(\ell)} = \phi_V^{(\ell)}(V_V^{(\ell)}), \qquad M_E^{(\ell)} = \phi_E^{(\ell)}(V_E^{(\ell)}). \tag{B.9}$$

The node-to-hyperedge and hyperedge-to-node aggregations are

$$A_E^{(\ell)} = \bar{D}_E^{-1} H^\top M_V^{(\ell)}, \qquad A_V^{(\ell)} = \bar{D}_V^{-1} H M_E^{(\ell)}. \tag{B.10}$$

The residual tangent updates are

$$\widetilde{V}_V^{(\ell)} = \mathrm{LN}\left(V_V^{(\ell)} + g_V^{(\ell)}([V_V^{(\ell)}; A_V^{(\ell)}])\right), \tag{B.11}$$

$$\widetilde{V}_E^{(\ell)} = \mathrm{LN}\left(V_E^{(\ell)} + g_E^{(\ell)}([V_E^{(\ell)}; A_E^{(\ell)}])\right). \tag{B.12}$$

Finally, the updated tangent vectors are re-embedded:

$$X_V^{(\ell+1)} = \mathrm{Exp}_o^{K_{\ell+1}}(\widetilde{V}_V^{(\ell)}), \qquad X_E^{(\ell+1)} = \mathrm{Exp}_o^{K_{\ell+1}}(\widetilde{V}_E^{(\ell)}). \tag{B.13}$$

This layer is tangent-space incidence message passing followed by Lorentz re-embedding.

B.4 Global Token Mixing

After L incidence aggregation layers, node and hyperedge tokens are concatenated in tangent coordinates:

$$V^{(L)} = \left[\mathrm{Log}_o^{K_L}(X_V^{(L)}); \mathrm{Log}_o^{K_L}(X_E^{(L)})\right] \in \mathbb{R}^{B\times(N+M)\times d}. \tag{B.14}$$

A standard Euclidean Transformer is then applied over the $(N+M)$ token dimension:

$$\bar{V} = \mathrm{Transformer}(V^{(L)}). \tag{B.15}$$

This stage allows non-local dependencies among all node and hyperedge slots to be modeled after local incidence aggregation. The transformed tangent tokens are mapped back to Lorentz coordinates and then projected to the final Poincaré ball:

$$\bar{X} = \mathrm{Proj}_{K_L}\left(\mathrm{Exp}_o^{K_L}(F_{\mathrm{out}}(\bar{V}))\right), \qquad Z_0 = \Pi_{K_L \to c_\star}(\bar{X}). \tag{B.16}$$

B.5 Computational Complexity

Let $|H|$ denote the number of nonzero incidence entries in one batch element. Using sparse incidence multiplication, bidirectional incidence aggregation costs approximately

$$O(B|H|d). \tag{B.17}$$

The global Transformer over node and hyperedge tokens costs

$$O(B(N+M)^2 d) \tag{B.18}$$

per layer, ignoring constant factors from multi-head projections. The decoder computes all node–hyperedge pairwise distances and costs

$$O(BNMd). \tag{B.19}$$

In the current fixed-size datasets, N and M are modest, so the Transformer and decoder are tractable. For substantially larger hypergraphs, sparse attention, blockwise decoding, or hierarchical hyperedge candidate selection would be natural extensions.

C Diffusion and Sampling Details

C.1 Encoder-Induced Tangent Coordinates

Let the frozen encoder produce the ordered Poincaré token sequence

$$Z_0 = \mathrm{Enc}_\phi(H) \in (\mathbb{B}^d_{c_\star})^{N+M}, \tag{C.1}$$

and define its origin-tangent coordinates by

$$V_0 = \log_0^{c_\star}(Z_0) \in \mathbb{R}^{(N+M)\times d}. \tag{C.2}$$

The diffusion model is trained on the empirical distribution of these coordinates. This is a coordinate-based construction: it does not optimize a density with respect to the manifold volume measure, and the discrete decoder is applied only after reverse diffusion.

C.2 DDPM Forward Marginal

For the cosine variance schedule used in our experiments, let

$$\alpha_t = 1 - \beta_t, \qquad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s. \tag{C.3}$$

The Euclidean forward marginal is

$$V_t = \sqrt{\bar{\alpha}_t} V_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{C.4}$$

with independent Gaussian noise over token and feature dimensions. The implementation then forms a ball-valued state and the network input as

$$Z_t = \mathrm{Proj}_{c_\star}\left(\exp_0^{c_\star}(V_t)\right), \qquad \widetilde{V}_t = \log_0^{c_\star}(Z_t). \tag{C.5}$$

Thus, V_t denotes the ideal Gaussian coordinate, whereas $\widetilde{V}_t$ includes the numerical exp–project–log path used by the actual model call.

C.3 Noise-Prediction Objective and Context

The HHGD denoiser predicts the injected tangent noise under the fixed uniform context:

$$\widehat{\epsilon} = \epsilon_\theta(\widetilde{V}_t, t; C), \qquad C = M^{-1}\mathbf{1}_N\mathbf{1}_M^\top. \tag{C.6}$$

With the encoder and decoder frozen, Phase II minimizes

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{H,t,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(\widetilde{V}_t, t; C)\right\|_2^2\right]. \tag{C.7}$$

Here H is used to construct the encoded latent V_0 and the supervision signal, whereas the denoiser condition C is fixed and independent of the target hypergraph. The same context is used during training and unconditional generation.

C.4 DDPM Reverse Parameterization

For completeness, the DDPM reverse mean in tangent coordinates is

$$\mu_\theta(\widetilde{V}_t, t; C) = \frac{1}{\sqrt{\alpha_t}}\left(\widetilde{V}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\widetilde{V}_t, t; C)\right), \tag{C.8}$$

where $C = M^{-1}\mathbf{1}_N\mathbf{1}_M^\top$ is the fixed uniform context used during both training and sampling. Each implemented reverse step maps its updated tangent coordinate through the exponential map and projects it back inside the Poincaré ball.

C.5 Deterministic DDIM Sampling

Formal generation uses $K = 50$ selected reverse timesteps from the $T = 1000$ cosine schedule and DDIM with $\eta = 0$ (Song et al., 2021). At a selected time τ_k, the sampler stores a ball state Z_{τ_k}, sets $\widetilde{V}_{\tau_k} = \log_0^{c_\star}(Z_{\tau_k})$, and estimates

$$V_{0,\theta}^{(k)} = \frac{\widetilde{V}_{\tau_k} - \sqrt{1-\bar{\alpha}_{\tau_k}}\,\epsilon_\theta(\widetilde{V}_{\tau_k}, \tau_k; C_{\text{gen}})}{\sqrt{\bar{\alpha}_{\tau_k}}}. \tag{C.9}$$

For numerical stability, the implementation clips the norm of each predicted clean token to at most 3. Writing the clipped value as $\overline{V}_{0,\theta}^{(k)}$, the deterministic update is

$$V_{\tau_{k-1}} = \sqrt{\bar{\alpha}_{\tau_{k-1}}}\,\overline{V}_{0,\theta}^{(k)} + \sqrt{1-\bar{\alpha}_{\tau_{k-1}}}\,\epsilon_\theta(\widetilde{V}_{\tau_k}, \tau_k; C_{\text{gen}}), \tag{C.10}$$

followed by

$$Z_{\tau_{k-1}} = \text{Proj}_{c_\star}\left(\exp_0^{c_\star}(V_{\tau_{k-1}})\right). \tag{C.11}$$

The final ball-valued state is passed directly to the frozen decoder.

C.6 Relation to Manifold Densities

An explicit coordinate density over Poincaré latents would require the change-of-variables factor

$$p_Z^{\text{coord}}(z) = p_V(\log_0^{c_\star}(z))\,\left|\det D \log_0^{c_\star}(z)\right|. \tag{C.12}$$

HHGD does not optimize this expression. Its diffusion loss is Euclidean noise prediction on encoder-induced tangent coordinates, interleaved in the implementation with Poincaré mappings and projection. It should therefore be interpreted as a tangent-coordinate prior over hyperbolic latent tokens, not as intrinsic Riemannian score matching.

D Poincaré-Space Denoising Network Details

D.1 Network Interface and Input Lift

For a batch of noisy token coordinates, the HHGD denoiser implements

$$\epsilon_\theta : (\widetilde{V}_t, t, C) \longmapsto \widehat{\epsilon}, \qquad \widetilde{V}_t, \widehat{\epsilon} \in \mathbb{R}^{B\times(N+M)\times d}. \tag{D.1}$$

A sinusoidal timestep embedding is processed by an MLP to obtain e_t. An affine projection lifts the noisy coordinates to the fixed-curvature Poincaré ball:

$$Z^{(0)} = \text{Proj}_{c_\star}\left(\exp_0^{c_\star}(W_{\text{in}}\widetilde{V}_t + b_{\text{in}})\right). \tag{D.2}$$

The sequence is split after the first N positions into node tokens $Z_V^{(0)}$ and hyperedge tokens $Z_E^{(0)}$.

D.2 Incidence-Aware Hypergraph Block

Even-indexed layers use the context matrix $C \in \mathbb{R}^{B\times N\times M}$. The implemented hypergraph convolution first transforms node and hyperedge tokens in the Poincaré ball, aggregates transformed nodes to hyperedges with a conformal-factor-weighted mean, updates the hyperedge tokens, and then aggregates those updated hyperedges back to nodes. We denote this operation by

$$(\bar{Z}_V, \bar{Z}_E) = \text{HConv}_{c_\star}(Z_V, Z_E; C). \tag{D.3}$$

A timestep MLP produces feature-wise scale and shift vectors (a_t, b_t). For either token type $S \in \{V, E\}$, the convolutional message is modulated in origin-tangent coordinates,

$$M_S = \log_0^{c_\star}(\bar{Z}_S) \odot (1 + a_t) + b_t, \tag{D.4}$$

and combined with the incoming token through a learned gate:

$$U_S = \log_0^{c_\star}(Z_S) + \sigma(W_g M_S + b_g) \odot M_S, \qquad Z'_S = \text{Proj}_{c_\star}(\exp_0^{c_\star}(U_S)). \tag{D.5}$$

This block preserves the node–hyperedge partition while allowing the supplied incidence context to control bidirectional aggregation.

D.3 Time-Conditioned Global Block

Odd-indexed layers concatenate the node and hyperedge tokens and apply Poincaré multi-head self-attention followed by a Poincaré feed-forward network. Adaptive normalization injects e_t before each branch. Residuals are formed after mapping the branch outputs to the origin tangent space:

$$\begin{aligned} A^{(\ell)} &= \mathrm{MHA}_{c_\star}\Big(\mathrm{AdaNorm}_{c_\star}(Z^{(\ell)}, e_t)\Big), \\ U_A^{(\ell)} &= \log_0^{c_\star}(Z^{(\ell)}) + \sigma(s_\ell)\log_0^{c_\star}(A^{(\ell)}), \end{aligned} \tag{D.6}$$

followed by an exp-map, projection, and an analogous normalized feed-forward residual using the same learned scalar gate $\sigma(s_\ell)$.

D.4 Training and Generation Contexts

The context supplied to the incidence-aware layers is fixed to

$$C_{\mathrm{train}} = C_{\mathrm{gen}} = \frac{1}{M}\mathbf{1}_N\mathbf{1}_M^\top. \tag{D.7}$$

The target incidence matrix H is used only to construct the encoded latent V_0 and the diffusion supervision signal. It is never passed to the denoiser. Thus, training and unconditional generation use matched conditioning. The fixed context is distinct from a target incidence matrix and from a padding mask.

D.5 Output Noise Prediction

After the final block, node and hyperedge tokens are concatenated and Poincaré-normalized. The network returns Euclidean tangent noise through

$$\widehat{\epsilon} = W_{\mathrm{out}}\log_0^{c_\star}\Big(\mathrm{Norm}_{c_\star}(Z^{(L)})\Big) + b_{\mathrm{out}}. \tag{D.8}$$

The output projection uses a small nonzero Xavier initialization in the frozen implementation. The corresponding Euclidean score ablation changes the score-network architecture; it is not obtained merely by deleting a single Lorentz radial-scaling map from the HHGD denoising network.

E Decoder, Losses, and Discretization

E.1 Poincaré Distance and Cardinality Logits

For $Z = [Z_V; Z_E] \in (\mathbb{B}^d_{c_\star})^{N+M}$, the decoder computes

$$d_{i,e} = d_{\mathbb{B}_{c_\star}}(z_i^v, z_e^h), \qquad D \in \mathbb{R}^{B\times N\times M}. \tag{E.1}$$

Non-finite distances are sanitized before the mean and standard deviation are computed over all NM entries of each sample. With

$$\mu_d = \mathrm{Mean}_{i,e}[d_{i,e}], \qquad \sigma_d = \max(\mathrm{Std}_{i,e}[d_{i,e}], 10^{-6}), \tag{E.2}$$

the normalized distance contribution is

$$r_{i,e} = -\frac{d_{i,e} - \mu_d}{(\mathrm{softplus}(\omega) + 0.1)\sigma_d}. \tag{E.3}$$

The scalar ω is learned. The base decoder bias is fixed at $b_0 = -2.2$, while the cardinality head predicts a column-level logit

$$a_e = g_\psi\big([\log_0^{c_\star}(z_e^h); \|z_e^h\|_2]\big), \qquad \widehat{s}_e = N\sigma(a_e). \tag{E.4}$$

Because the implementation adds $a_e - b_0$ to a base logit containing b_0, the final soft incidence score simplifies to

$$P_{i,e} = \sigma(r_{i,e} + a_e). \tag{E.5}$$

These scores jointly depend on the sample-level distance statistics and are used for reconstruction and membership ranking. They are not independently sampled as the final binary hypergraph.

E.2 Phase-I Reconstruction Losses

Let $s_e(H) = \sum_i H_{i,e}$ be the target cardinality. The Phase-I objective is

$$\mathcal{L}_{\mathrm{AE}} = \mathcal{L}_{\mathrm{BCE}} + \lambda_{\mathrm{topo}}\mathcal{L}_{\mathrm{topo}} + \lambda_{\mathrm{boundary}}\mathcal{L}_{\mathrm{boundary}} + \lambda_{\mathrm{size}}\mathcal{L}_{\mathrm{size}}. \tag{E.6}$$

The loss balancing coefficients are set to $\lambda_{\mathrm{topo}} = 0.1$, $\lambda_{\mathrm{boundary}} = 0.01$, and $\lambda_{\mathrm{size}} = 0.1$, which act as auxiliary reconstruction objectives and radial regularization to match the gradient scales across different losses. Additionally, to handle class imbalance in incidence prediction, the positive-class weight in the binary cross-entropy loss is set to $w_+ = 9$:

$$\mathcal{L}_{\mathrm{BCE}} = -\operatorname{Mean}_{i,e}\big[w_+ H_{i,e}\log P_{i,e} + (1 - H_{i,e})\log(1 - P_{i,e})\big]. \tag{E.7}$$

No slot mask is passed to this loss in the evaluated fixed-slot training path. The topology term is the sum of three mean-squared errors,

$$\begin{aligned}\mathcal{L}_{\mathrm{topo}} = \ & \mathrm{MSE}\left(\frac{1}{M}\sum_e P_{:,e}, \frac{1}{M}\sum_e H_{:,e}\right) \\ & + \mathrm{MSE}\left(\frac{1}{N}\sum_i P_{i,:}, \frac{1}{N}\sum_i H_{i,:}\right) + \mathrm{MSE}(\mathrm{Mean}(P), \mathrm{Mean}(H))\,.\end{aligned} \tag{E.8}$$

For node and hyperedge Poincaré tokens, the boundary term averages $[\max(0, \|z\|_2 - 0.9)]^2 + [\max(0, 0.1 - \|z\|_2)]^2$ over tokens. The size-head term is

$$\mathcal{L}_{\mathrm{size}} = \mathrm{SmoothL1}\left(\frac{\widehat{s}}{N}, \frac{s(H)}{N}\right), \tag{E.9}$$

which is evaluated over all fixed hyperedge slots, including empty ones.

E.3 Validation-Calibrated Cardinality Top-k

For a candidate scale factor g, each slot receives

$$k_e(g) = \mathrm{clip}_{[0,N]}(\mathrm{round}(g\widehat{s}_e))\,. \tag{E.10}$$

Values below two are reset to zero to prevent trivial singletons. Otherwise, the decoder selects the $k_e(g)$ nodes with the highest membership probabilities $P_{i,e}$. Hyperedge columns are processed in sequence; if a newly formed node set duplicates a previously assigned active hyperedge, the redundant column is zeroed out.

To determine the optimal scale, we perform a grid search for $g \in \{0.25, 0.275, \ldots, 1.25\}$ on a validation calibration batch. The scalar scale factor γ is selected to minimize a joint validation score accounting for normalized structural discrepancies (node degree, active hyperedge size, edge count, and density) alongside invalid/singleton penalties. Once calibrated, γ is fixed for inference:

$$\widehat{H} = \mathcal{Q}_\gamma(P, \widehat{s}). \tag{E.11}$$

Crucially, the held-out test set is excluded during this calibration step, ensuring a deterministic, fully zero-shot inference pipeline.

E.4 Fixed Slots and Hyperedge Ordering

HHGD operates on an ordered, fixed-width incidence representation. During preprocessing, node indices within each hyperedge are sorted, and the resulting tuples are canonicalized via lexicographical ordering under the dataset's native node labeling. Training directly aligns corresponding incidence columns, bypassing costly combinatorial optimization over hyperedge permutations. At generation time, columns with $k_e < 2$ or duplicate membership sets are automatically rendered inactive. This fixed-order canonicalization provides a stable representation for hyperedge learning while ensuring high computational efficiency.

F Implementation and Evaluation Details

F.1 Model Configuration

The proposed frozen model uses token dimension $d = 64$, hidden dimension 128, 4 attention heads, 2 HGCN encoder layers, and 4 HHGD denoising layers, with dropout 0.1 and fixed interface curvature magnitude $c_\star = 1$. The main HHGD model combines the incidence-aware HGCN encoder, the Poincaré-space denoising network in Appendix D, and the cardinality-aware Poincaré-distance decoder in Appendix E. The diffusion schedule contains $T = 1000$ cosine steps.

F.2 Training Protocol

Training is performed in two stages using AdamW with weight decay 10^{-4}.

Phase I: autoencoder training. The encoder and decoder are trained jointly for 50 epochs at learning rate 10^{-3} and batch size 8:

$$H \xrightarrow{\mathrm{Enc}_\phi} Z_0 \xrightarrow{\mathrm{Dec}_\psi} (P, \widehat{s}). \tag{F.1}$$

They minimize the four-term reconstruction objective in Appendix E.

Phase II: diffusion-prior training. Both the encoder and decoder are frozen. For 100 epochs at learning rate 10^{-4}, the HHGD denoiser predicts the noise added to encoder-induced tangent coordinates under a fixed uniform context

$$C = M^{-1}\mathbf{1}_N\mathbf{1}_M^\top : \tag{F.2}$$

$$\mathcal{L}_{\mathrm{diff}} = \mathbb{E}_{H,t,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(\widetilde{V}_t, t; C)\right\|_2^2\right]. \tag{F.3}$$

The incidence matrix H is used to produce the encoder-induced coordinates and the corresponding training target, but is not passed to the denoiser. The same context C is used during training and unconditional sampling. The global gradient-norm clipping threshold is 1.0; the implementation uses a stricter half-threshold for the Phase-II score-network update and rejects non-finite batches. In all reported experiments, no non-finite batches were encountered during this phase.

F.3 Calibration and Sampling Protocol

After training, an independent batch of 50 unconditional soft samples is used to select the cardinality scale γ on validation data. The 41-point grid and validation criterion are specified in Appendix E; the selected value is then frozen. Formal generation begins with

$$V_T \sim \mathcal{N}(0, I), \qquad Z_T = \mathrm{Proj}_{c_\star}\left(\exp_0^{c_\star}(V_T)\right), \tag{F.4}$$

where each sample has token shape $(N + M) \times d$. The model applies 50-step DDIM sampling with $\eta = 0$ and the uniform generation context, passes the final ball-valued tokens to the frozen decoder, and returns

$$(P, \widehat{s}) = \mathrm{Dec}_\psi(\widehat{Z}_0), \qquad \widehat{H} = \mathcal{Q}_\gamma(P, \widehat{s}). \tag{F.5}$$

Validation screening uses 50 generated samples per run. After the checkpoints and cardinality scale are frozen, each held-out run generates 100 samples in batches of 25. No held-out hypergraph is used as a model input or to select γ.

F.4 Datasets

The eight datasets and their fixed slot budgets are summarized in Table 5. Every processed dataset contains 350 training, 50 validation, and 100 held-out test hypergraphs. Incidence tensors have a single dataset-specific shape, while node and hyperedge masks identify the active submatrix of each sample.

For the four synthetic families, each accepted hypergraph is generated from an independently seeded draw. Exact duplicate incidence structures are rejected, and the accepted samples are assigned to

Table 5: Summary of the eight hypergraph datasets. $N_{\max}$ and $M_{\max}$ denote the dataset-specific node and hyperedge slot budgets after padding; masks identify the active nodes and hyperedges in each sample.

Dataset	Regime	$N_{\max}$	$M_{\max}$	Train/Val/Test
Erdős–Rényi (ER)	Synthetic	32	114	350/50/100
Hypergraph SBM	Synthetic	32	101	350/50/100
Tree	Synthetic	32	31	350/50/100
Ego	Synthetic	130	87	350/50/100
Contact Primary School	Real-world temporal	64	116	350/50/100
NDC Substances	Real-world temporal	64	120	350/50/100
Email-Eu	Real-world temporal	64	128	350/50/100
Tags-Math	Real-world temporal	64	128	350/50/100

the training, validation, and test splits in that order. The slot budgets are the maximum node and hyperedge counts observed over the frozen 500-sample dataset, so no synthetic incidence is cropped to fit the representation.

For the four temporal real-world sources, events are first ordered by timestamp and original event identifier and divided approximately $70\%/10\%/20\%$ into training, validation, and test sources without splitting equal timestamps. Connected local hypergraphs are then sampled separately within each source split. Sampling admits a source hyperedge only when its complete membership set fits the budget; it never crops a hyperedge. Duplicate columns and duplicate processed incidence matrices are rejected. The real-world slot budgets are chosen from training-source candidates using bounded 95th-percentile rules; validation and test data do not set these budgets. A bounded sample may omit part of a larger local neighborhood, but every retained hyperedge is complete. Model fitting uses the training split, model selection and cardinality calibration use only validation data, and the test split is reserved for the final evaluation.

F.5 Baselines and Ablations

Table 2 summarizes the 2×2 ablation study formed by the Cartesian product of two encoders (HGCN and Euclidean) and two score networks (Hyperbolic and Lightweight). All four variants share the same cardinality-aware Poincar'e decoder, training budget, data splits, and seed pairs. This setup explicitly isolates component choices within a common decoder rather than comparing entirely Euclidean versus entirely hyperbolic pipelines.

Under the fixed validation protocol, Euc+Hyperbolic obtains the lowest aggregate validation score. However, the difference from HGCN+Hyperbolic is small relative to the variation across the three seed pairs and does not support a reliable ranking. We therefore use HGCN+Hyperbolic as the named HHGD configuration because it provides a geometrically coherent pipeline: the incidence-aware HGCN encoder, hyperbolic denoiser, and Poincaré decoder share the same non-Euclidean inductive bias. This choice is made before held-out evaluation and is not based on test results. The displayed comparisons show the effects of these component choices.

Throughout the paper, **HHGD** denotes the HGCN+Hyperbolic configuration.

The external comparison includes HyperPA, HyperLap+, HYGENE, VAE, GAN, and incidence-image diffusion, as described in Section 4.3. Every executable method is fitted or trained on the same frozen training files and evaluated with the same artifact validator and metric implementation; values from the methods' original papers are not inserted into our tables. HyperLap+ fits its templates from the training split only. HYGENE on Email-Eu is reported as OOM because its validation run exceeded the 24 GB accelerator-memory limit even at batch size one; no pilot or incomplete output is substituted.

F.6 Evaluation Metrics

All distributional statistics are computed on the active submatrix selected by the node and hyperedge masks. Unless stated otherwise, distances are 1-Wasserstein distances (W_1), for which lower values are better.

Active hyperedge count. For each hypergraph, the evaluator records the number of active hyperedge columns and compares the resulting per-hypergraph distributions.

Node degree and hyperedge size. The evaluator pools $d_i = \sum_e H_{i,e}$ across active nodes and $k_e = \sum_i H_{i,e}$ across active hyperedges. Zero-valued node degrees and hyperedge sizes are retained in these two pooled distributions.

Pair co-degree and hyperedge intersection. For every distinct node pair, co-degree is the number of shared hyperedges; for every distinct hyperedge pair, intersection size is the number of shared nodes. Only positive off-diagonal values enter these two pooled distributions. Thus, a small distance on either metric does not by itself measure the frequency of zero-overlap pairs.

Spectral MMD. For each hypergraph, the evaluator forms the normalized hypergraph Laplacian and converts its eigenvalues to a normalized 200-bin histogram on $[-10^{-5}, 2]$. Spectral is the biased squared maximum mean discrepancy between the real and generated histogram sets, using a Gaussian kernel applied to total-variation distance with bandwidth one. It is a global spectral summary, not a direct measure of semantic hierarchy recovery.

Additional structural diagnostics. The appendix evaluation additionally reports W_1 distances for incidence density and the number of connected components. Closeness, betweenness, and harmonic centralities are computed on the weighted clique projection, where an edge's length is the reciprocal of its node-pair co-degree, and each centrality distribution is compared separately. We do not collapse these quantities into a single centrality score, and the evaluator does not report a clustering coefficient metric.

Strict validity. A generated sample is valid only if its incidence tensor is binary and nonempty, its active-node count matches the declared target, it has no nonzero entries outside the masks, and every active hyperedge is nonempty, has cardinality at least two, and has a membership set distinct from every other active hyperedge in the sample. Validity is the fraction of generated *samples* satisfying all of these conditions, not the fraction of individually valid hyperedges.

Exact uniqueness and novelty. Each active incidence matrix is converted to a bipartite graph whose node and hyperedge vertex types are preserved. Weisfeiler–Lehman hashes are used only to form candidate buckets, followed by exact typed-graph isomorphism tests. Uniqueness is the fraction of generated samples belonging to distinct isomorphism classes, and novelty is the fraction not isomorphic to any training hypergraph. Both are invariant to row and column permutations. High uniqueness or novelty does not imply validity or structural fidelity.

All structural, uniqueness, and novelty metrics are computed over all 100 generated samples in a run rather than only its valid subset. Each table entry is the mean and sample standard deviation over the three fixed seed pairs.

G Additional External-Baseline Results

Table 6 presents the complementary evaluation on the remaining four datasets under the identical protocol, providing a comprehensive assessment across diverse hypergraph topologies.

Table 6: Remaining external-baseline results: SBM, Ego, NDC, and Email-Eu. Means ± SD over three seeds. Results are means ± sample standard deviations over three seeds, with 100 samples per run. Bold indicates the best displayed mean, including ties, without implying significance. Higher validity and uniqueness and lower structural distances are better. Novelty is 1.000 ± 0.000 for all completed runs and is omitted. "OOM" indicates out-of-memory error on a 24 GB GPU during training/inference.

Method	Valid (%)↑	Unique (%)↑	EdgeCnt↓	NodeDeg↓	EdgeSize↓	PairDeg↓	Intersect↓	Spectral↓
Ego								
HyperPA	$\mathbf{49.3}_{\pm 5.1}$	$\mathbf{100.0}_{\pm 0.0}$	$14.35_{\pm 1.77}$	$1.943_{\pm 0.038}$	$\mathbf{0.045}_{\pm 0.013}$	$1.740_{\pm 0.070}$	$0.657_{\pm 0.010}$	$0.084_{\pm 0.001}$
HyperLap+	$46.3_{\pm 0.6}$	$\mathbf{100.0}_{\pm 0.0}$	$5.32_{\pm 0.21}$	$0.885_{\pm 0.014}$	$0.210_{\pm 0.021}$	$\mathbf{0.033}_{\pm 0.002}$	$\mathbf{0.048}_{\pm 0.004}$	$\mathbf{0.029}_{\pm 0.000}$
HYGENE	$14.3_{\pm 10.8}$	$98.3_{\pm 0.6}$	$34.26_{\pm 3.03}$	$\mathbf{0.788}_{\pm 0.112}$	$2.438_{\pm 0.697}$	$0.212_{\pm 0.007}$	$0.388_{\pm 0.489}$	$0.132_{\pm 0.043}$
VAE	$0.7_{\pm 0.6}$	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{3.49}_{\pm 0.49}$	$2.069_{\pm 0.243}$	$1.818_{\pm 0.460}$	$0.102_{\pm 0.128}$	$0.375_{\pm 0.192}$	$0.053_{\pm 0.003}$
GAN	$0.0_{\pm 0.0}$	$99.7_{\pm 0.6}$	$5.28_{\pm 1.02}$	$1.416_{\pm 0.328}$	$1.093_{\pm 0.165}$	$0.149_{\pm 0.024}$	$0.063_{\pm 0.040}$	$0.054_{\pm 0.012}$
Inc. Diff.	$3.7_{\pm 3.2}$	$\mathbf{100.0}_{\pm 0.0}$	$17.14_{\pm 0.38}$	$1.707_{\pm 0.046}$	$1.173_{\pm 0.023}$	$0.129_{\pm 0.009}$	$0.066_{\pm 0.013}$	$0.135_{\pm 0.004}$
HHGD	$5.0_{\pm 6.1}$	$\mathbf{100.0}_{\pm 0.0}$	$8.04_{\pm 4.67}$	$1.902_{\pm 0.466}$	$1.041_{\pm 0.841}$	$1.881_{\pm 2.497}$	$0.484_{\pm 0.644}$	$0.118_{\pm 0.053}$
Hypergraph SBM								
HyperPA	$0.3_{\pm 0.6}$	$\mathbf{100.0}_{\pm 0.0}$	$3.79_{\pm 0.59}$	$4.369_{\pm 0.064}$	$0.015_{\pm 0.000}$	$1.113_{\pm 0.036}$	$0.172_{\pm 0.008}$	$0.165_{\pm 0.004}$
HyperLap+	$28.0_{\pm 1.7}$	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{2.35}_{\pm 0.66}$	$\mathbf{0.456}_{\pm 0.064}$	$\mathbf{0.000}_{\pm 0.000}$	$0.043_{\pm 0.006}$	$0.014_{\pm 0.002}$	$0.010_{\pm 0.000}$
HYGENE	$44.7_{\pm 6.7}$	$97.3_{\pm 4.6}$	$31.44_{\pm 7.12}$	$2.525_{\pm 0.429}$	$0.482_{\pm 0.230}$	$0.187_{\pm 0.044}$	$\mathbf{0.013}_{\pm 0.010}$	$0.026_{\pm 0.014}$
VAE	$0.7_{\pm 0.6}$	$\mathbf{100.0}_{\pm 0.0}$	$4.22_{\pm 0.48}$	$0.601_{\pm 0.379}$	$1.201_{\pm 0.051}$	$0.055_{\pm 0.040}$	$0.067_{\pm 0.021}$	$\mathbf{0.008}_{\pm 0.001}$
GAN	$0.0_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$5.65_{\pm 0.82}$	$0.756_{\pm 0.095}$	$1.313_{\pm 0.032}$	$\mathbf{0.015}_{\pm 0.010}$	$0.043_{\pm 0.006}$	$\mathbf{0.008}_{\pm 0.001}$
Inc. Diff.	$1.7_{\pm 1.2}$	$\mathbf{100.0}_{\pm 0.0}$	$27.57_{\pm 2.17}$	$4.170_{\pm 1.293}$	$1.638_{\pm 0.178}$	$0.677_{\pm 0.260}$	$0.203_{\pm 0.075}$	$0.017_{\pm 0.001}$
HHGD	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$6.63_{\pm 3.79}$	$3.041_{\pm 0.374}$	$0.489_{\pm 0.033}$	$0.446_{\pm 0.201}$	$0.027_{\pm 0.032}$	$0.137_{\pm 0.022}$
NDC Substances								
HyperPA	$\mathbf{93.0}_{\pm 1.0}$	$71.0_{\pm 1.7}$	$8.93_{\pm 0.40}$	$1.487_{\pm 0.143}$	$\mathbf{1.252}_{\pm 0.036}$	$0.908_{\pm 0.201}$	$0.598_{\pm 0.054}$	$0.078_{\pm 0.002}$
HyperLap+	$46.7_{\pm 4.5}$	$\mathbf{100.0}_{\pm 0.0}$	$7.03_{\pm 0.45}$	$\mathbf{1.222}_{\pm 0.132}$	$2.188_{\pm 0.856}$	$1.050_{\pm 0.686}$	$1.702_{\pm 1.224}$	$\mathbf{0.011}_{\pm 0.002}$
HYGENE	$20.3_{\pm 5.1}$	$99.3_{\pm 1.2}$	$12.86_{\pm 4.17}$	$5.505_{\pm 5.440}$	$7.173_{\pm 4.488}$	$12.690_{\pm 14.497}$	$13.018_{\pm 9.683}$	$0.044_{\pm 0.008}$
VAE	$0.7_{\pm 1.2}$	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{6.95}_{\pm 2.93}$	$2.176_{\pm 1.261}$	$2.250_{\pm 1.784}$	$1.021_{\pm 1.369}$	$1.246_{\pm 1.492}$	$0.022_{\pm 0.004}$
GAN	$0.3_{\pm 0.6}$	$\mathbf{100.0}_{\pm 0.0}$	$19.36_{\pm 3.37}$	$1.695_{\pm 0.284}$	$1.616_{\pm 0.137}$	$\mathbf{0.267}_{\pm 0.083}$	$\mathbf{0.431}_{\pm 0.079}$	$0.048_{\pm 0.005}$
Inc. Diff.	$0.0_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$48.01_{\pm 8.78}$	$3.562_{\pm 0.774}$	$2.410_{\pm 0.283}$	$0.459_{\pm 0.071}$	$0.892_{\pm 0.036}$	$0.173_{\pm 0.019}$
HHGD	$31.7_{\pm 1.5}$	$\mathbf{100.0}_{\pm 0.0}$	$14.10_{\pm 1.20}$	$2.823_{\pm 0.197}$	$3.289_{\pm 0.063}$	$0.497_{\pm 0.196}$	$0.826_{\pm 0.058}$	$0.101_{\pm 0.005}$
Email-Eu								
HyperPA	$\mathbf{45.3}_{\pm 1.2}$	$96.7_{\pm 0.6}$	$\mathbf{10.01}_{\pm 1.13}$	$2.130_{\pm 0.371}$	$\mathbf{0.397}_{\pm 0.040}$	$0.629_{\pm 0.158}$	$0.212_{\pm 0.012}$	$\mathbf{0.043}_{\pm 0.002}$
HyperLap+	$1.3_{\pm 0.6}$	$\mathbf{100.0}_{\pm 0.0}$	$23.47_{\pm 4.47}$	$2.621_{\pm 0.155}$	$0.425_{\pm 0.013}$	$\mathbf{0.110}_{\pm 0.009}$	$0.384_{\pm 0.044}$	$0.049_{\pm 0.003}$
HYGENE					*OOM*			
VAE	$0.0_{\pm 0.0}$	$\mathbf{100.0}_{\pm 0.0}$	$16.18_{\pm 2.23}$	$\mathbf{1.554}_{\pm 0.146}$	$0.966_{\pm 0.230}$	$0.995_{\pm 0.066}$	$\mathbf{0.123}_{\pm 0.018}$	$0.084_{\pm 0.007}$
GAN	$9.3_{\pm 12.9}$	$99.3_{\pm 1.2}$	$21.91_{\pm 9.93}$	$6.369_{\pm 4.581}$	$2.007_{\pm 1.226}$	$3.922_{\pm 1.766}$	$1.895_{\pm 0.675}$	$0.104_{\pm 0.011}$
Inc. Diff.	$12.3_{\pm 6.0}$	$\mathbf{100.0}_{\pm 0.0}$	$58.65_{\pm 2.46}$	$7.524_{\pm 1.071}$	$1.068_{\pm 0.233}$	$0.925_{\pm 0.137}$	$0.131_{\pm 0.037}$	$0.128_{\pm 0.003}$
HHGD	$36.0_{\pm 1.0}$	$\mathbf{100.0}_{\pm 0.0}$	$18.21_{\pm 0.52}$	$2.745_{\pm 0.320}$	$2.031_{\pm 0.200}$	$0.676_{\pm 0.331}$	$0.197_{\pm 0.004}$	$0.236_{\pm 0.007}$

H Additional Ablation Results on Synthetic Datasets

To provide a comprehensive evaluation within main-text space limits, Table 7 reports the ablation performance on the remaining four datasets (ER, SBM, Tree, and Ego) under the same experimental protocol. Across all configurations, models maintain near-perfect novelty and uniqueness (1.000 ± 0.000), with the sole exception of Euc+Light on SBM (uniqueness 0.983 ± 0.029). All structural metrics—including W_1 distances and spectral MMD—are computed over all generated samples, including invalid ones.

Table 7: Encoder–score ablation on the remaining four datasets (ER, SBM, Tree, and Ego), completing the evaluation in Table 3. Entries report three-seed means ± sample standard deviations over 100 generated samples per run. Bold indicates the best mean within each dataset.

Encoder+Score	Valid (%)↑	EdgeCnt↓	NodeDeg↓	EdgeSize↓	PairDeg↓	Intersect↓	Spectral↓
Erdős–Rényi (ER)							
HHGD	$\mathbf{100.0}_{\pm 0.0}$	$16.52_{\pm 6.01}$	$2.806_{\pm 0.247}$	$\mathbf{0.179}_{\pm 0.067}$	$0.400_{\pm 0.103}$	$0.086_{\pm 0.026}$	$0.046_{\pm 0.012}$
Euc+Hyp	$\mathbf{100.0}_{\pm 0.0}$	$15.40_{\pm 4.41}$	$2.776_{\pm 0.219}$	$0.181_{\pm 0.051}$	$0.418_{\pm 0.069}$	$0.090_{\pm 0.018}$	$0.043_{\pm 0.008}$
HGCN+Light	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{13.45}_{\pm 9.07}$	$2.755_{\pm 0.329}$	$0.351_{\pm 0.121}$	$0.135_{\pm 0.133}$	$0.015_{\pm 0.020}$	$0.038_{\pm 0.015}$
Euc+Light	$\mathbf{100.0}_{\pm 0.0}$	$13.52_{\pm 6.67}$	$\mathbf{2.729}_{\pm 0.257}$	$0.367_{\pm 0.062}$	$\mathbf{0.120}_{\pm 0.067}$	$\mathbf{0.012}_{\pm 0.009}$	$\mathbf{0.037}_{\pm 0.011}$
Hypergraph SBM							
HHGD	$\mathbf{100.0}_{\pm 0.0}$	$6.63_{\pm 3.79}$	$3.041_{\pm 0.374}$	$\mathbf{0.489}_{\pm 0.033}$	$0.446_{\pm 0.201}$	$\mathbf{0.027}_{\pm 0.032}$	$\mathbf{0.137}_{\pm 0.022}$
Euc+Hyp	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{6.37}_{\pm 4.58}$	$\mathbf{3.021}_{\pm 0.402}$	$0.497_{\pm 0.029}$	$0.426_{\pm 0.181}$	$0.028_{\pm 0.029}$	$0.138_{\pm 0.025}$
HGCN+Light	$\mathbf{100.0}_{\pm 0.0}$	$22.10_{\pm 24.51}$	$3.286_{\pm 1.640}$	$0.929_{\pm 0.061}$	$\mathbf{0.316}_{\pm 0.063}$	$0.100_{\pm 0.015}$	$0.206_{\pm 0.185}$
Euc+Light	$\mathbf{100.0}_{\pm 0.0}$	$23.21_{\pm 25.26}$	$3.367_{\pm 1.675}$	$0.934_{\pm 0.057}$	$0.321_{\pm 0.058}$	$0.101_{\pm 0.014}$	$0.219_{\pm 0.200}$
Tree							
HHGD	$\mathbf{100.0}_{\pm 0.0}$	$1.80_{\pm 0.67}$	$\mathbf{0.601}_{\pm 0.072}$	$0.181_{\pm 0.025}$	$0.126_{\pm 0.018}$	$0.075_{\pm 0.011}$	$0.083_{\pm 0.021}$
Euc+Hyp	$\mathbf{100.0}_{\pm 0.0}$	$2.01_{\pm 0.39}$	$0.609_{\pm 0.067}$	$0.174_{\pm 0.019}$	$0.125_{\pm 0.009}$	$0.075_{\pm 0.004}$	$0.087_{\pm 0.013}$
HGCN+Light	$\mathbf{100.0}_{\pm 0.0}$	$1.06_{\pm 0.09}$	$0.652_{\pm 0.016}$	$0.075_{\pm 0.022}$	$0.071_{\pm 0.008}$	$0.038_{\pm 0.006}$	$\mathbf{0.079}_{\pm 0.002}$
Euc+Light	$\mathbf{100.0}_{\pm 0.0}$	$\mathbf{1.04}_{\pm 0.15}$	$0.654_{\pm 0.027}$	$\mathbf{0.074}_{\pm 0.053}$	$\mathbf{0.070}_{\pm 0.019}$	$\mathbf{0.037}_{\pm 0.012}$	$\mathbf{0.079}_{\pm 0.003}$
Ego							
HHGD	$5.0_{\pm 6.1}$	$\mathbf{8.04}_{\pm 4.67}$	$1.902_{\pm 0.466}$	$1.041_{\pm 0.841}$	$1.881_{\pm 2.497}$	$0.484_{\pm 0.644}$	$0.118_{\pm 0.053}$
Euc+Hyp	$\mathbf{5.3}_{\pm 7.5}$	$8.72_{\pm 6.20}$	$1.980_{\pm 0.440}$	$\mathbf{0.963}_{\pm 0.418}$	$2.203_{\pm 2.885}$	$0.667_{\pm 0.807}$	$0.114_{\pm 0.065}$
HGCN+Light	$0.7_{\pm 0.6}$	$25.18_{\pm 0.19}$	$1.200_{\pm 0.038}$	$1.578_{\pm 0.022}$	$\mathbf{0.136}_{\pm 0.003}$	$\mathbf{0.008}_{\pm 0.001}$	$\mathbf{0.048}_{\pm 0.006}$
Euc+Light	$0.0_{\pm 0.0}$	$24.87_{\pm 0.29}$	$\mathbf{1.179}_{\pm 0.039}$	$1.611_{\pm 0.065}$	$0.141_{\pm 0.012}$	$0.010_{\pm 0.006}$	$0.049_{\pm 0.005}$