Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Autoencoder–Graph Neural Network Framework for Robust 3D Pore Connectivity Classification in Noisy Volumetric Images

Kishore S
Indian Institute of Technology Roorkee
Mayank Goswami
mayankfph@iitr.ac.in

Indian Institute of Technology Roorkee

Research Article

Keywords: pore connectivity, μCT, graph neural networks, GraphSAGE, autoencoders, synthetic phantoms

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11652673/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Autoencoder–Graph Neural Network Framework for Robust 3D Pore Connectivity Classification in Noisy Volumetric Images

Kishore S[1] and Mayank Goswami,[1,2]

[1]Department of Physics, IIT Roorkee
[2]MF School of AI and DS, IIT Roorkee
mayank.goswami@ph.iitr.ac.in

***Abstract*—Pore connectivity is a critical property governing transport of fluids, gases and ions in porous materials. Connectivity is often inferred from micro-computed tomography (µCT) volumes via thresholding and connected-component analysis, but this pipeline is sensitive to noise and limited resolution, producing both false connections and missed links. This study tests whether graph-based learning, using geometric features and 3D image representations, can correct such errors.**

We generate 250 synthetic 128^3-voxel phantoms with known ground-truth connectivity, degrade them with point-spread blur and noise, and evaluate ten models spanning classical machine learning, 3D CNNs, GCN, GraphSAGE and autoencoder–GNN fusion. The best models (GraphSAGE and AE+GraphSAGE) achieve F1 scores of 0.954 and 0.957, compared with 0.468 for the raw connected-component baseline. Improvements are driven primarily by geometric features (neck width, centroid distance and pore volume), while frozen autoencoder embeddings provide negligible benefit. However, improved edge classification does not consistently translate into better cluster agreement or percolation prediction.

***Index Terms*—pore connectivity, µCT, graph neural networks, GraphSAGE, autoencoders, synthetic phantoms**

1 Introduction

1.1 Why pore connectivity matters

In porous materials, transport depends on both porosity and how pores are interconnected. Connectivity controls flow pathways and percolation, so samples with similar porosity can exhibit very different permeability.

1.2 The problem with classical connectivity analysis

A typical µCT workflow thresholds the volume into a pore/solid mask and infers connectivity with 26-connected components. Noise can introduce spurious links, while limited resolution can erase thin necks and split truly connected pores, biasing downstream transport estimates.

1.3 Why a learned correction, and why synthetic data

Connectivity is naturally a graph problem: pores are nodes and candidate links are edges. We therefore combine local 3D image features (via an autoencoder) with network context (via a GNN).

Because real µCT volumes rarely have verified ground-truth connectivity, we train and evaluate on synthetic 3D phantoms with exactly known connections.

Corresponding author: mayank.goswami@ph.iitr.ac.in

1.4 Research objectives

1. Develop a synthetic 3D pore-network generator with known ground-truth connectivity and realistic imaging degradation.
2. Quantify the errors of classical Otsu thresholding and 26-connectivity analysis under varying noise and segmentation thresholds.
3. Compare ten models covering classical, tabular, image-based, graph-based, and AE-GNN fusion approaches.
4. Evaluate robustness to noise, neck width, threshold variations, and unseen geometries, alongside feature and architecture ablations.

1.5 Contributions

- Develop a reproducible benchmark of 250 synthetic 128^3-voxel volumes with verified connectivity graphs and ten evaluated model variants.
- Demonstrate that classical connectivity analysis suffers primarily from false merging (precision 0.309 at recall 0.967), which graph-based learning can substantially reduce.
- Show that frozen, reconstruction-trained autoencoder embeddings provide negligible improvements over graph-only models, with AE–GCN exhibiting poor robustness to unseen noise.
- Highlight that improved edge-level F1 does not necessarily translate into better cluster agreement or percolation accuracy.

2 Background and Related Work

Marker-controlled watershed segmentation of the distance transform is the established route from a binary pore image to a pore network (Gostick, 2017), and PoreSpy (Gostick et al., 2019) packages these classical operations for porous-media images; our node segmentation follows the same watershed strategy. On the learning side, graph convolutional networks (Kipf and Welling, 2017) and GraphSAGE (Hamilton et al., 2017) are the two graph layers used here, and the task of predicting whether a pair of nodes is linked is an instance of GNN link prediction (Zhang and Chen, 2018). Variational autoencoders (Kingma and Welling, 2013) provide the probabilistic alternative to our plain autoencoder. Recent work has embedded GNNs inside pore-network solvers for permeability prediction (Zhao and Xiao) and compared

CNNs with equivariant GNNs for transport properties of µCT images (Cai et al., 2021). In contrast to those regression-style targets, our target is edge-level connectivity classification, and we study the specific failure modes of the classical segmentation pipeline.

3 Methodology

The pipeline has seven stages (Figure 1 summarises the data flow in words): phantom generation, imaging and noise simulation, classical segmentation, node and candidate-edge construction with feature extraction, dataset assembly with leakage checks, model training, and evaluation. All random draws are seeded from a single configuration (seed 42).

3.1 Synthetic phantom generator with exact ground truth

Each pore is modelled as a randomly oriented superellipsoid. In its local coordinate system, a point x lies inside the pore if

$$\left|\frac{x_1}{a_1}\right|^p + \left|\frac{x_2}{a_2}\right|^p + \left|\frac{x_3}{a_3}\right|^p \le 1$$

where $a_1, a_2, a_3$are the semi-axes, p controls the shape, and p=2 gives an ellipsoid. The orientation is determined by a rotation matrix R. The semi-axes are sampled as

$$a_k \sim U(4, 9) \text{ voxels}$$

with the constraint

$$\sum_k \epsilon_k = 0$$

so that volume is preserved while elongation varies, with an aspect ratio of up to 1.5.

An exact support function determines the distance from the pore centre to its surface along any direction, allowing pores to be positioned with precisely controlled separation and neck lengths.

Connections are modelled as cylindrical necks with radius

$$w \sim U(1, 3.5) \text{ voxels}$$

and length

$$L = \begin{cases} 0, & \text{with probability } 0.25, \\ U(1, 8) \text{ voxels}, & \text{otherwise.} \end{cases}$$

Unconnected pores and necks are maintained at approximately 3 voxels or more apart. An independent raster-level validation recomputes the 26-connected components of the clean, noise-free void mask. A phantom is retained only if these components match the generator's ground-truth clusters and the measured minimum separation is at least 2.5 voxels.

Pores are organised into four configurations: isolated singles (30%), pairs (20%), chains of 3–6 pores (20%), and clusters of 4–7 pores with optional loops (30%). Each volume is independently assigned one of three configurations: ordinary (40%), percolating (35%), or near-miss (25%). Percolating volumes contain a continuous top-to-bottom pore chain, whereas near-miss volumes contain a similar chain with exactly one neck omitted.

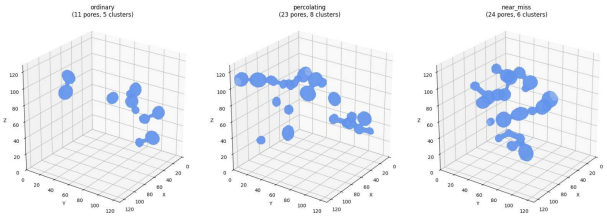


Fig. 1. Representative synthetic phantoms showing ordinary, percolating and near-miss configurations, rendered as isosurfaces of the void space. Titles report the number of pores and ground-truth clusters.

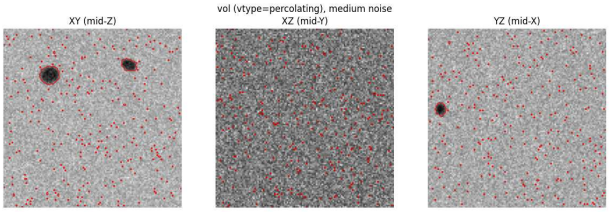


Fig. 2. Noisy slices (XY, XZ, YZ) of a percolating volume at medium noise after Otsu thresholding (Otsu threshold 0.691). Red marks voxels below threshold. In this example the classical CC analysis percolates correctly and detects 21 nodes (23 true pores) with 35 candidate edges (9 true connections).

3.2 Imaging and noise model

The clean binary phantom (I_0, where 1 represents solid and 0 represents void) is degraded through three stages to simulate µCT imaging:

(i) Gaussian point-spread blur with σ_{psf} = 0.8 voxels;

(ii) Poisson photon noise $I_p = \frac{Poisson(I_b N)}{N}$, so that variance scales with signal;

(iii) Additive Gaussian detector noise N(0, σ_g^2). Training, validation and test volumes draw (σ_g, $N_{photons}$) log-uniformly per volume from σ_g [0.04, 0.25] and N [500, 40].

For controlled evaluation sweeps, four named levels are fixed, low (0.05, 400), medium (0.12, 150), high (0.20, 60), very high (0.30, 30), plus an **unseen** level (0.40, 15) that is harsher than anything seen in training.

3.3 Classical pipeline (Model 0) and node segmentation

The volume is percentile-normalised (0.5–99.5%), thresholder with Otsu's method, which chooses a threshold maximising the between-class variance cleaned with a morphological opening, and labelled under 26-connectivity. Components smaller than 150 voxels are discarded. This minimum-blob-size filter was set deliberately (the smallest real pore is about 268 voxels; noise speckle clusters are 20–60 voxels); an earlier setting of 20 voxels produced thousands of spurious nodes per volume at high noise. Model 0 uses only a single indicator, whether two candidate pores share a raw component, as its connectivity "probability". Where noise merges several real pores into one component, a marker-controlled watershed on the Euclidean distance transform (h-maxima markers, h = 1) re-separates them into candidate nodes.

3.4 Candidate graph and features

Every pair of nodes whose surfaces lie within 20 voxels becomes a candidate edge (not all pairs). Each edge carries 33 handcrafted features in four groups: **distance**

(centroid distance raw and size-normalised, surface gap, direction cosine with the percolation axis), **volume** (log pore volumes and their ratio), **neck** (the bottleneck or maximin radius: the largest distance-transform level t at which the two pore peaks remain 26-connected), and **intensity profile** (mean, min, max, variance, maximum gradient, mid-point value, fraction and longest run above threshold, sampled along the line between the nearest surface points on raw and smoothed volumes, plus a MAD-based local noise estimate). Nodes carry 19 geometric and intensity descriptors plus 2 contextual ones. For image-based models, a 32^3 patch re-oriented so its local z-axis points from pore i to pore j is cut around every candidate edge.

3.5 *Dataset, split and leakage control*

The 250 volume indices are permuted once and split 70/15/15 at the **volume level** (175 / 38 / 37). All edges, patches and node features of a volume stay in one split. A hard assertion verifies that the three id sets are disjoint, cover all 250 volumes and that no two volumes share a generator seed key (the notebook confirms: "OK: 175 train / 38 val / 37 test volumes, 250 unique phantom seeds, no overlap"). Feature standardisation is fitted on the training split only; every decision threshold is tuned on the validation split only.

3.6 *Models compared*

Autoencoder. The 3D convolutional autoencoder consists of an encoder with three stride-2 convolutional layers, reducing the spatial dimensions from 32^3 to 4^3, with channel dimensions increasing from $1 \to 16 \to 32 \to 64$. A linear layer then projects the encoded representation into a \(d\)-dimensional latent vector. The decoder mirrors the encoder and ends with a sigmoid activation to reconstruct the input patch. The autoencoder is trained using mean-squared error (MSE) reconstruction loss:

$$\mathscr{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N} \left\| X_i - \widehat{X}_i \right\|_2^2$$

Training uses every candidate patch from the training split, irrespective of its connectivity label. This label-agnostic strategy prevents connectivity labels from influencing the learned representation. After training, the autoencoder is frozen, and its latent embeddings are used as input to the fusion models.

Graph Models. The candidate pore network is represented as a graph, where pores are nodes and candidate connections are edges. For each candidate edge \((i,j)\), the edge-level representation is constructed as

$$\mathbf{r}_{ij} = \left[\mathbf{h}_i \parallel \mathbf{h}_j \parallel \mathbf{e}_{ij} \parallel \mathbf{z}_{ij}^{AE} \right]$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are the GNN node embeddings, $\mathbf{e}_{ij}$ contains handcrafted geometric edge features, $\mathbf{z}_{ij}^{AE}$ is the frozen autoencoder latent representation, and $\parallel$ denotes concatenation. The AE embedding is included only in fusion models.

The resulting edge representation is passed through a $64 \to 16 \to 1$ multilayer perceptron (MLP) with a sigmoid output to estimate the connection probability: $p_{ij} = \sigma\left(MLP\left(\mathbf{r}_{ij}\right)\right)$

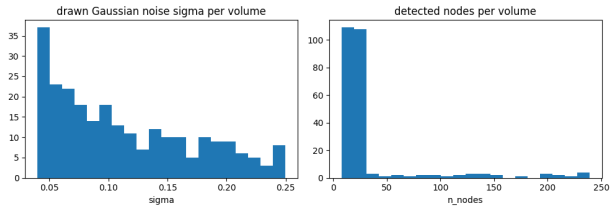


Fig. 3. Dataset summary: distribution of the Gaussian noise σ drawn per volume (left) and of detected nodes per volume (right).

Two graph architectures are evaluated:

1. Graph Convolutional Network (GCN):

The GCN updates node representations using normalized adjacency information:

$$H^{(l+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

where $\widetilde{A} = A + I$ is the adjacency matrix with self-loops, $\widetilde{D}$ is its degree matrix, $H^{(l)}$ represents node embeddings at layer l , and $W^{(l)}$ (is the trainable weight matrix.

3.7 *Training and evaluation metrics*

Connected edges are the minority class, so all learned models use class-weighted binary cross-entropy, and all reported precision, recall and F1 refer to the **connected** class (never accuracy). Early stopping and model selection use validation F1. For cluster-level evaluation, edges with $p_{ij} \geq \tau$ are kept and a union-find over the kept edges gives predicted clusters, compared with the true clusters using the Adjusted Rand Index (ARI) and Normalised Mutual Information (NMI), averaged over volumes. **Percolation accuracy** is the fraction of volumes for which the predicted top-to-bottom connectivity (a node on the z = 0 face and a node on the z = D−1 face in the same predicted component) equals the true percolation flag.

4 Results

4.1 *Dataset characteristics*

The candidate graph covered **100%** of the true edges on average (mean coverage 1.000), so the ceiling on recall is perfect and errors come from classification, not candidate generation. The volume-type mix is 96 ordinary, 83 percolating and 71 near-miss. The mean positive-edge fraction is 0.275 per volume. **An important caveat:** the test split has far fewer candidate edges per volume (54 vs. about 182) and a much higher positive rate (18.3% vs. 4–5%) than train and validation. F1 depends on class prevalence, so test F1 values are not directly comparable with validation F1 values (compare Table 4 with Section 4.3), and thresholds tuned on validation face a different prevalence at test time.

4.2 *Training behaviour*

Table IV summarises the training behaviour of the evaluated models.

TABLE I. Key experimental configuration (full preset).

Parameter	Value
Volumes / size / pores per volume	250 / 128³ voxels / 10–30
Split (volume-level)	70% / 15% / 15% = 175 / 38 / 37
Pore equivalent radius; neck radius; neck length	4–9 voxels; 1–3.5 voxels; 0 or 1–8 voxels
PSF σ; training Gaussian σ_g; training photons	0.8; 0.04–0.25; 500–40
Minimum component / node volume	150 voxels
Candidate radius (default)	12 voxels (max 20)
AE latent dimension; AE epochs; AE learning rate	64; 60; 1e-3 (858,657 parameters)
GNN layers; hidden size; epochs; learning rate; patience	2; 32; 150; 2e-3; 25
Loss	Class-weighted BCE, w = #neg / #pos (train)
Model selection / threshold tuning	Validation F1 only

TABLE II. The evaluated models (plus the VAE variant) and the information each may use.

Model	Inputs	Graph	Image
Model 0: Raw CC	Shared raw connected component (no parameters)	No	Indirect
Distance rule	Centroid distance < τ_d (τ_d = 19 voxels, fitted on train)	No	No
Model 1: XGBoost	33 handcrafted edge features	No	No
Model 2: MLP	33 handcrafted edge features	No	No
Model 3: 3D CNN	32³ neck patch, trained end-to-end on labels	No	Yes
Model 4: GCN	Node + edge handcrafted features, 2 GCN layers	Yes	No
Model 5: GraphSAGE	Node + edge handcrafted features, 2 SAGE layers	Yes	No
Model 6: AE + MLP	Handcrafted features + frozen 64-d AE latent	No	Yes (frozen)
Model 7: AE + GCN	GCN + frozen AE latent + edge features	Yes	Yes (frozen)
Model 7b: AE + GraphSAGE	GraphSAGE + frozen AE latent + edge features	Yes	Yes (frozen)
VAE + GCN (Exp. 8)	As Model 7 but with a variational latent (β = 0.001)	Yes	Yes (frozen)

TABLE III. Dataset statistics by split (means per volume; edge totals and positive rate from the edge tables).

Split	Nodes	Cand. edges	Pos. edges	True pores	True edges	Total edges	Pos. rate
Train (175)	36.50	181.77	8.61	20.27	14.22	31,810	4.7%
Val (38)	39.32	183.61	7.45	20.79	14.24	6,977	4.1%
Test (37)	22.57	54.30	9.95	19.89	14.19	2,009	18.3%

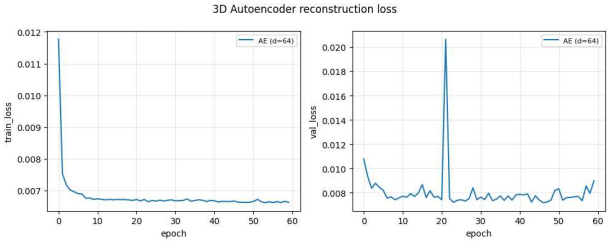


Fig. 4. Autoencoder reconstruction loss (train and validation). The loss saturates within a few epochs and the validation curve fluctuates.

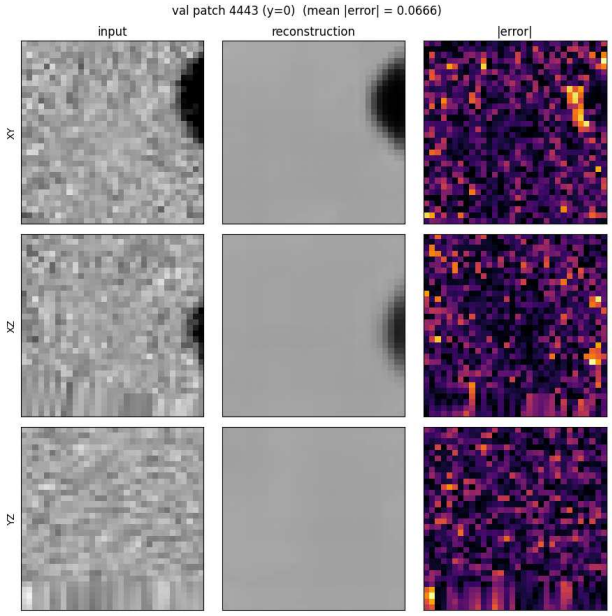


Fig. 5. Reconstruction of a validation neck patch by the autoencoder: input, reconstruction and absolute error.

4.3 Main comparison on the held-out test set (Experiment 2)

At the edge level the picture is clear. Raw CC has near-perfect recall (0.967) but precision of only 0.309: it labels many unconnected neighbouring pores as connected; i.e. false merging is its dominant error. The distance rule is a weak non-learned baseline (F1 0.656). Moving to learned models using the handcrafted features lifts F1 to 0.86–0.88, graph models raise it further, and the top two are **AE+GraphSAGE (F1 0.957, precision 0.947, recall 0.967)** and **GraphSAGE (F1 0.954, precision 0.919, recall 0.992)**. Their difference (0.003) is far below what a single seed can resolve, so they should be regarded as tied. The 3D CNN trained directly on labels (0.897) matches GCN and AE+GCN, showing that raw image patches carry usable signal when the network is trained supervised.

The metrics disagree, and this matters. Raw CC has the best cluster agreement (ARI 0.965, NMI 0.984) and shares the best percolation accuracy (0.919, i.e. 34 of 37 volumes) with AE+MLP, even though its edge F1 is the worst of the learned and non-learned models other than the distance rule and AE+MLP. The best edge-level models have ARI 0.83–0.85 and percolation accuracy 0.68. With only 37 test volumes, each volume is worth 2.7 percentage points of percolation accuracy, so differences of a few volumes are not conclusive; but the direction, learned models not beating raw CC on cluster and percolation metrics, is consistent across ARI, NMI and percolation. We offer possible explanations in Section 5.2 but the notebook does not contain the diagnostics (for example a per-volume-type confusion matrix) needed to confirm them.

TABLE IV. Training summary (validation F1 is the selection criterion).

Model	Epochs run	Best/final val. F1	Observation
MLP (M2)	150/150	0.737 (final)	Slowly improving; never early-stopped
AE (d = 64)	60/60	val loss 0.0074–0.0090	Train loss plateaus at ≈ 0.0066 by epoch 7
AE+MLP (M6)	stopped at 31	0.683	Early stopping triggered; poor generalisation
3D CNN (M3)	40/40	peak ≈ 0.842	Val loss rose 0.125→0.38 while train loss fell (overfitting)
GCN (M4)	stopped at 60	0.776	Stable
GraphSAGE (M5)	stopped at 95	0.873	Best validation F1 of all models
AE + GCN (M7)	stopped at 117	0.827	Slow convergence
AE + GraphSAGE (M7b)	stopped at 53	0.844	Fast convergence
VAE+GCN	stopped at 117	0.822	VAE stopped at epoch 43 (val loss 0.01338)

TABLE V. Test-set performance of all models (thresholds tuned on validation, applied once to test). Highlighted rows are the two best edge-level models.

Model	F1	Prec.	Rec.	ARI	NMI	Perc. acc.
Raw CC (Model 0)	0.468	0.309	0.967	0.965	0.984	0.919
Distance rule	0.656	0.651	0.660	0.566	0.816	0.568
XGBoost (Model 1)	0.883	0.800	0.986	0.955	0.979	0.865
MLP handcrafted (Model 2)	0.856	0.788	0.938	0.887	0.958	0.838
3D CNN (Model 3)	0.897	0.855	0.943	0.835	0.918	0.703
GCN (Model 4)	0.899	0.840	0.967	0.909	0.938	0.811
GraphSAGE (Model 5)	0.954	0.919	0.992	0.850	0.921	0.676
AE + MLP (Model 6)	0.651	0.503	0.921	0.901	0.959	0.919
AE + GCN (Model 7)	0.896	0.901	0.891	0.793	0.897	0.703
AE+GraphSAGE (Model 7b)	0.957	0.947	0.967	0.825	0.914	0.676

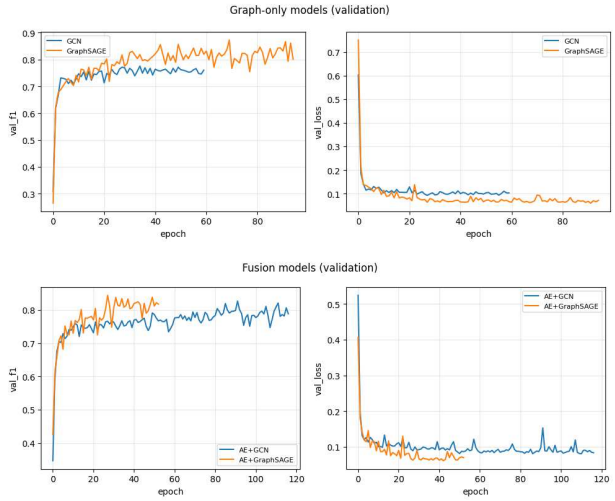


Fig. 6. Validation F1 and validation loss for the graph-only models (GCN, GraphSAGE; top) and the fusion models (AE+GCN, AE+GraphSAGE; bottom).

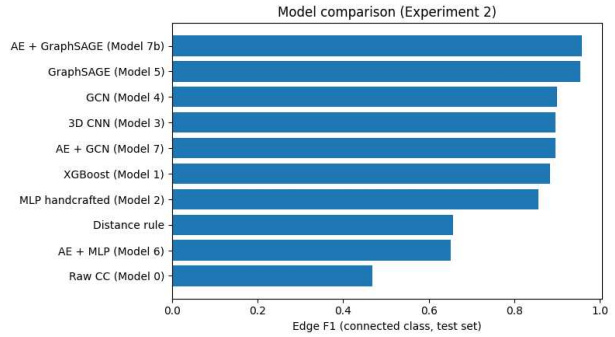


Fig. 7. Edge F1 (connected class) of all evaluated models on the test set.

4.4 Robustness to noise (Experiments 1 and 3)

Each model was evaluated, without retraining, on 20 fresh volumes at each fixed noise level (Table 7, Figures 9 and 10). All models degrade from low to medium noise, but the learned graph models degrade far less than

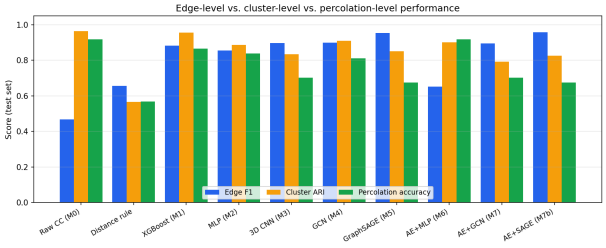


Fig. 8. Edge F1, cluster ARI and percolation accuracy for each model on the test set (values from Table V). Higher edge F1 does not necessarily imply higher ARI or percolation accuracy.

TABLE VI. Percolation accuracy on the test set (37 volumes).

Model	Perc. acc.	Correct
Raw CC (Model 0)	0.919	34 / 37
AE + MLP (Model 6)	0.919	34 / 37
XGBoost (Model 1)	0.865	32 / 37
MLP handcrafted (Model 2)	0.838	31 / 37
GCN (Model 4)	0.811	30 / 37
3D CNN (Model 3)	0.703	26 / 37
AE + GCN (Model 7)	0.703	26 / 37
GraphSAGE (Model 5)	0.676	25 / 37
AE + GraphSAGE (Model 7b)	0.676	25 / 37
Distance rule	0.568	21 / 37

raw CC. At medium to very-high noise, GraphSAGE and AE+GraphSAGE keep F1 between 0.82 and 0.88, XGBoost, MLP, CNN and GCN stay around 0.72–0.80, whereas raw CC sits at 0.48–0.55. Averaged over the five levels, AE+GraphSAGE leads (0.886), followed by GraphSAGE (0.876), while raw CC averages 0.558.

Two points stand out. First, the AE+GCN fusion model **collapses to F1 0.377** at the unseen noise level, whereas the same encoder with GraphSAGE keeps 0.869 and the graph-only GCN keeps 0.743. This suggests that, at least with GCN aggregation, the frozen image embedding becomes an unreliable input when the noise distribution moves outside training. Second, the distance rule is nearly noise-independent (as expected for a rule that ignores the

TABLE VII. Edge F1 by noise level. "Unseen" (σ = 0.40, 15 photons) is harsher than any training condition.

Model	Low	Med.	High	Very high	Unseen	Mean
Raw CC (M0)	0.708	0.508	0.480	0.549	0.547	0.558
Distance rule	0.767	0.612	0.539	0.596	0.640	0.631
XGBoost (M1)	0.977	0.779	0.721	0.778	0.785	0.808
MLP (M2)	0.975	0.718	0.716	0.780	0.785	0.795
3D CNN (M3)	0.984	0.802	0.736	0.785	0.750	0.811
GCN (M4)	0.977	0.775	0.726	0.784	0.743	0.801
GraphSAGE (M5)	0.983	0.879	0.839	0.873	0.804	0.876
AE+MLP (M6)	0.873	0.657	0.641	0.680	0.686	0.707
AE+GCN (M7)	0.971	0.822	0.787	0.733	0.377	0.738
AE+SAGE (M7b)	0.981	0.881	0.822	0.879	0.869	0.886

TABLE VIII. Recall of the connected class versus true neck radius (voxels).

Model	w = 1	w = 2	w = 3	w = 4	w = 5
3D CNN (M3)	0.831	0.915	0.936	0.679	0.731
AE+GCN (M7)	0.840	0.889	0.818	0.811	0.692
AE+SAGE (M7b)	0.853	0.897	0.982	0.981	1.000
AE+MLP (M6)	0.511	0.769	0.909	0.849	1.000
Distance rule	0.636	0.615	0.618	0.509	0.654
GCN (M4)	0.982	0.957	0.936	0.906	0.885
GraphSAGE (M5)	0.916	0.983	0.991	1.000	1.000
MLP (M2)	0.422	0.795	0.855	0.717	0.769
Raw CC (M0)	1.000	1.000	1.000	1.000	1.000
XGBoost (M1)	0.960	0.983	0.982	0.943	0.923

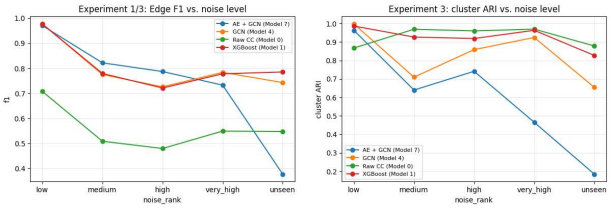


Fig. 9. Edge F1 (left) and cluster ARI (right) versus noise level for Raw CC, XGBoost, GCN and AE+GCN. Raw CC is consistently far below the learned models in F1; AE+GCN drops sharply at the unseen noise level.

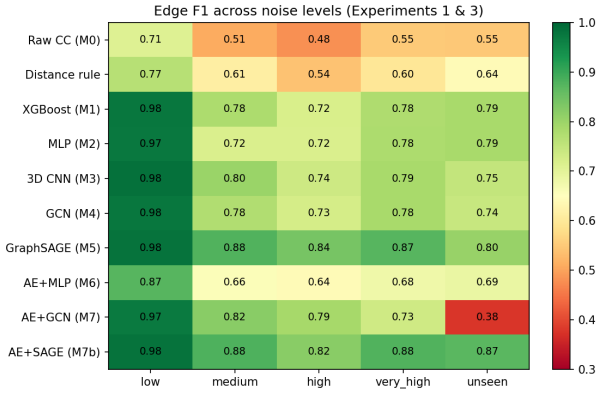


Fig. 10. Heat map of edge F1 for all models across noise levels (from Table VII). The red cell marks the AE+GCN collapse at unseen noise (0.377).

image) but is much weaker than learned models.

4.5 *Neck-width sensitivity (Experiment 4)*

Using 20 volumes per neck radius (1–5 voxels, medium noise), the notebook reports recall by neck width (Table 8) and plots F1 (Figure 11, left). Raw CC has recall 1.0 at every width, so its F1 is limited entirely by precision. Contrary to the intuition that thin necks are hardest, learned-model recall is generally **lower for wider necks** (for example GCN 0.982 at w = 1 versus 0.885 at w = 5; AE+GCN 0.840 versus 0.692), whereas AE+GraphSAGE and GraphSAGE reach recall of 1.0 at the widest necks. Reading the plot, F1 for GCN falls from about 0.87 at w = 1 to about 0.67–0.69 for w ≥ 2, AE+GCN falls from about 0.84 to about 0.62 at w = 5, and raw CC falls from about 0.58 to about 0.33 (approximate values read from the figure). A plausible but untested explanation is that wide necks merge two pores into one node during watershed splitting, so the pair never appears as a candidate edge; we did not verify this.

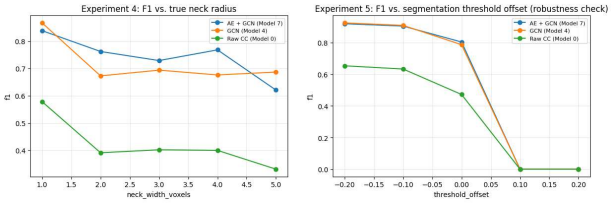


Fig. 11. Left: F1 versus true neck radius (Experiment 4). Right: F1 versus segmentation-threshold offset (Experiment 5). Both for Raw CC, GCN and AE+GCN.

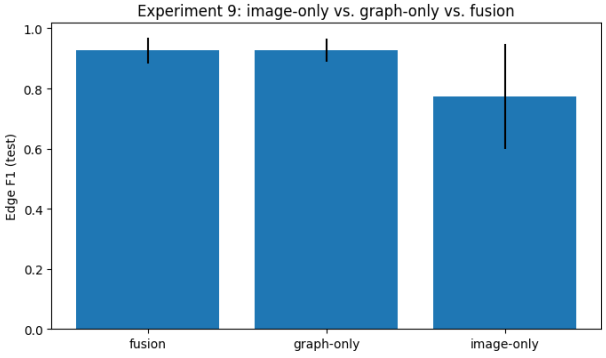


Fig. 12. Mean edge F1 per arm (image-only 0.774, graph-only 0.927, fusion 0.927). Error bars show the spread between the two models in each arm.

4.6 *Segmentation-threshold sensitivity (Experiment 5)*

The Otsu threshold was shifted by offsets from −0.2 to +0.2 (normalised units) at medium noise, 20 volumes per offset (Figure 11, right). Negative offsets (a more conservative void mask) **improved** every model: approximate F1 was 0.92 for GCN and AE+GCN and about 0.65 for raw CC at −0.2, versus about 0.79–0.80 and 0.47 at zero offset. At offsets of +0.1 and +0.2 **F1 fell to 0 for all models, including raw CC**. Because the baseline fails identically, this points to a breakdown of the segmentation and graph construction at those offsets rather than a model-specific failure, but the notebook does not diagnose the cause (for instance, whether positive edges disappear or the candidate graph degenerates), so the F1 = 0 values should be investigated before being interpreted physically.

4.7 *Image-only vs. graph-only vs. fusion (Experiment 9)*

Graph-only and fusion arms have identical mean F1 (0.927); the image-only arm is lower (0.774) because of the poor AE+MLP model. Pairing each graph model with its fused version isolates the AE contribution: GCN → AE+GCN changes F1 by −0.003 (0.899 → 0.896; recall

TABLE IX. Grouping of the models into three information arms (test set).

Arm	Model	F1	Prec.	Rec.	ARI	Perc. acc.
Image-only	3D CNN (M3)	0.897	0.855	0.943	0.835	0.703
Image-only	AE + MLP (M6)	0.651	0.503	0.921	0.901	0.919
Graph-only	GCN (M4)	0.899	0.840	0.967	0.909	0.811
Graph-only	GraphSAGE (M5)	0.954	0.919	0.992	0.850	0.676
Fusion	AE + GCN (M7)	0.896	0.901	0.891	0.793	0.703
Fusion	AE + GraphSAGE (M7b)	0.957	0.947	0.967	0.825	0.676

falls from 0.967 to 0.891 while precision rises from 0.840 to 0.901), and GraphSAGE → AE+GraphSAGE by +0.003 (0.954 → 0.957; precision 0.919 → 0.947, recall 0.992 → 0.967). The frozen AE therefore trades recall for precision but does not improve overall edge F1 in a way that can be distinguished from noise.

4.8 Generalisation to unseen conditions (Experiment 10)

Graph models transfer well to geometric shifts: GraphSAGE and AE+GraphSAGE exceed 0.95 for more pores and larger pores and stay above 0.90 for bigger clusters. Model-specific fragilities are informative. The distance rule scores **0.000** on large pores (a fixed 19-voxel centroid-distance cut-off cannot apply when pores are bigger and centroids farther apart); XGBoost drops to 0.645 on large pores, suggesting reliance on absolute-scale features, while GraphSAGE and the CNN stay above 0.99; and AE+GCN fails on unseen noise. These results indicate that the geometric prior learned from the training generator is not catastrophically brittle, but they were obtained on 20 volumes per condition and a single training seed.

4.9 Feature ablation (Experiment 6)

Removing the **distance** group costs the most (F1 0.859 → 0.574), followed by **neck width** (→ 0.622) and **volume** (→ 0.694). Removing the **intensity-profile** group changes F1 by only 0.008, so once geometry is known the along-the-neck intensity samples add little. Adding the AE embedding to the same MLP **reduces F1 from 0.856 to 0.651** (precision 0.788 → 0.503), consistent with the 64 extra dimensions encouraging overfitting (the model early-stopped at epoch 31). The near-identical values of the full ablation row (0.859) and Model 2 (0.856), which are the same configuration, give a rough indication (about 0.003) of run-to-run noise. Interestingly, the "− volume" model has the best ARI (0.974) and percolation accuracy (0.946) in the table despite a much lower F1, another instance of edge F1 and connectivity-level metrics diverging.

4.10 AE versus VAE (Experiment 8)

The VAE variant is marginally better on every cluster-level measure (ARI 0.854 vs. 0.793; percolation accuracy 0.784 vs. 0.703) and equal in F1 (0.900 vs. 0.896). With a single seed and 37 test volumes this is a suggestive trend, not a demonstrated advantage.

4.11 Latent dimension (Experiment 7) and candidate radius

F1 varies only between 0.896 and 0.917 across latent widths 16–128 with no monotone trend; ARI is highest at d = 128 (0.881) but the pattern is irregular. Given the 0.006 seed-to-seed standard deviation measured for GCN (Section 4.12) and a single seed per width, we cannot claim any latent size is better; the representation appears neither clearly under- nor over-compressed. For the candidate radius, coverage reaches 0.999 at 8 voxels and 1.000 from 12 voxels, while the number of candidate edges roughly doubles from 8 to 20 voxels (1,168 → 2,142) and F1 drifts down slowly (0.921 → 0.909). A radius of 8–12 voxels therefore captures essentially all true edges at lower cost; at 4 voxels about 2% of true edges are lost (coverage 0.981), though F1 on the retained edges is highest.

4.12 4.12 Multi-seed stability

Retraining the GCN with three seeds gave test F1 of 0.901, 0.904 and 0.890, i.e. **0.898 ± 0.006** (mean ± standard deviation), in line with the 0.899 of the main run. This is the only model for which seed variability was measured; gaps between top models in Table 5 (for example 0.954 vs. 0.957) are well within what such variability could produce, and comparisons across other models should be treated accordingly.

4.13 4.13 Qualitative results

In test volume 6 the notebook found 6 candidate edges where raw CC wrongly connected a pair and the fusion model corrected it, 0 edges where CC missed a connection and the model recovered it, and 4 edges that the model still gets wrong. The per-edge output table for this volume illustrates the correction: for pair (0, 1), the ground truth is "not connected", raw CC predicts connected, the AE+GCN probability is 0.107 (rejected) while the graph-only GCN probability is 0.945 (accepted). The fusion model fixes this particular error that its graph-only counterpart makes, though the pair (1, 5) shows both learned models wrongly accepting a pair that is not connected (probabilities 0.9895 and 0.9475, ground truth 0).

5 Discussion

5.1 What the evidence supports

1. **False merging is the classical pipeline's main weakness here.** Raw CC finds nearly every true connection (recall 0.967 on the test set; 1.0 across all neck widths) but accepts many false ones (precision 0.309). Any improvement must therefore come from discriminating real necks from near-misses, and learned models do this: GraphSAGE-family models reach precision of 0.92–0.95 at recall 0.97–0.99.
2. **Geometry, not appearance, carries most of the signal.** The ablation shows that centroid distance, neck width and pore volume are indispensable,

TABLE X. Edge F1 on out-of-distribution sets: unseen noise (σ = 0.40), more pores than training (35–45), larger pores (r_{eq} 10–14 vs. 4–9), and bigger clusters (8–12 pores vs. 4–7).

Model	Unseen noise	More pores	Large pores	Big clusters
3D CNN (M3)	0.750	0.947	0.996	0.873
AE+GCN (M7)	0.377	0.936	0.943	0.887
AE+SAGE (M7b)	0.869	0.961	0.983	0.906
AE+MLP (M6)	0.686	0.843	0.949	0.749
Distance rule	0.640	0.773	0.000	0.670
GCN (M4)	0.743	0.928	0.981	0.848
GraphSAGE (M5)	0.804	0.955	0.991	0.917
MLP (M2)	0.785	0.921	0.893	0.831
Raw CC (M0)	0.547	0.616	0.817	0.496
XGBoost (M1)	0.785	0.928	0.645	0.861

TABLE XI. Handcrafted-feature-group ablation (MLP retrained per row; single seed) and the AE-embedding contrast.

Configuration	F1	Prec.	Rec.	ARI	Perc. acc.
Full (all four groups)	0.859	0.778	0.959	0.917	0.865
– neck width	0.622	0.575	0.677	0.642	0.838
– intensity profile	0.851	0.768	0.954	0.927	0.811
– distance	0.574	0.428	0.875	0.853	0.838
– volume	0.694	0.540	0.970	0.974	0.946
Full, no AE embedding (= Model 2)	0.856	0.788	0.938	0.887	0.838
Full + AE embedding (= Model 6)	0.651	0.503	0.921	0.901	0.919

TABLE XII. Plain autoencoder versus variational autoencoder embeddings feeding the same GCN fusion model.

Model	F1	Prec.	Rec.	ARI	NMI	Perc. acc.
AE + GCN	0.896	0.901	0.891	0.793	0.897	0.703
VAE + GCN	0.900	0.874	0.927	0.854	0.924	0.784

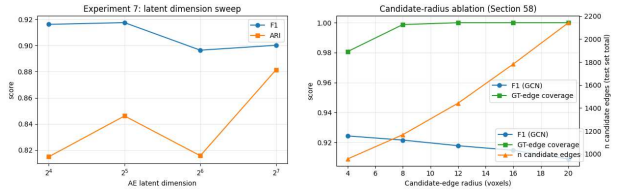


Fig. 13. Left: effect of the AE latent dimension on F1 and ARI (Experiment 7). Right: effect of the candidate-edge radius on F1, true-edge coverage and number of candidate edges.

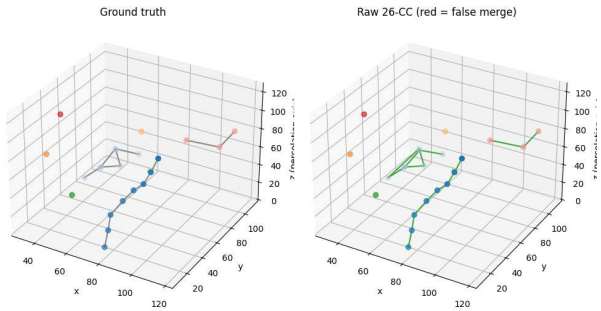


Fig. 14. Ground-truth connectivity graph (left) versus the graph implied by raw 26-CC (right) for a demonstration phantom. Node colours mark true clusters; the raw-CC graph adds false edges (green) that do not exist in the ground truth.

while the intensity profile is nearly redundant. The bottleneck neck radius, computed from the distance transform, is the single feature most closely tied to physical connectivity.

3. **Graph context helps further.** GraphSAGE (0.954) and GCN (0.899) outperform the same features without a graph (MLP 0.856, XGBoost 0.883), at least at the edge level. GraphSAGE's separate self and neighbour weights appear to suit this task better than GCN's symmetric normalisation.
4. **A frozen, reconstruction-trained autoencoder is not the source of the gain.** Fusion changes F1 by –0.003 (GCN) and +0.003 (GraphSAGE), degrades

TABLE XIII. Latent-dimension sweep (AE+GCN) and candidate-edge radius ablation (trained GCN).

AE d	F1	ARI	Radius	Cand. edges	Coverage	F1
16	0.916	0.815	4	957	0.981	0.924
32	0.917	0.846	8	1,168	0.999	0.921
64	0.896	0.816	12	1,442	1.000	0.918
128	0.900	0.881	16	1,782	1.000	0.915
			20	2,142	1.000	0.909

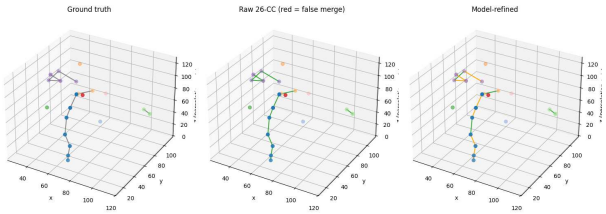


Fig. 15. Ground truth, raw 26-CC, and the model-refined (AE+GCN) graph for a test volume. In the refined panel, green = correct connection, red = false connection, and orange = missed true connection.

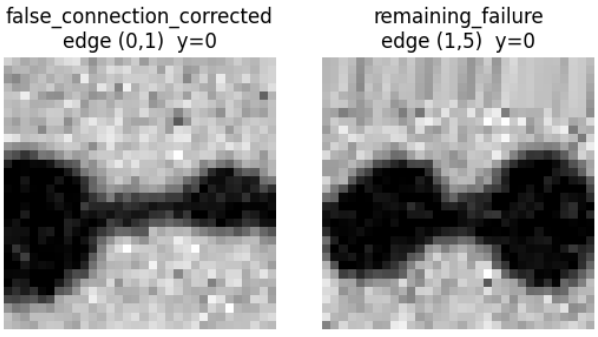


Fig. 16. Mid-slices of neck patches for an example false connection that was corrected (left) and a remaining failure (right) in a test volume.

an MLP by 0.205, and makes GCN brittle under unseen noise. Meanwhile an end-to-end 3D CNN reaches 0.897, so image information is useful when trained for the task. The likely issue is the objective: reconstruction loss is dominated by bulk intensity, and its curve saturated at about 0.0066 within a few epochs (Figure 4), so the latent code may not preserve the thin-neck cues that matter.

5. **Robustness is good for geometry shifts, fragile for noise shifts in specific models.** GraphSAGE and AE+GraphSAGE degrade gracefully; AE+GCN does not.

5.2 Why edge F1 did not improve cluster and percolation metrics

This is the most important caveat of the study. Raw CC has ARI 0.965 and percolation accuracy 0.919, while GraphSAGE and AE+GraphSAGE, the best edge classifiers, have ARI 0.85/0.83 and percolation accuracy 0.68. We can suggest hypotheses, none of which is tested in the notebook: (i) cluster and percolation metrics are all-or-nothing aggregates of edges, so a single wrong edge can merge two clusters or create or break a percolation path, and edge F1 gives no credit for how errors are distributed across volumes; (ii) in near-miss volumes exactly one neck is missing, so one false-positive edge across the gap changes the percolation answer, and learned models that accept a plausible-looking close pair are exposed to this specific error whereas raw CC rarely bridges a clean gap; (iii) thresholds were tuned on validation edge F1 at 4% prevalence and not on connectivity-level objectives; (iv) 37 test volumes give coarse estimates. A per-volume-type breakdown of percolation errors, and tuning the threshold or loss directly for cluster or percolation accuracy, are the natural next steps.

5.3 Threats to validity

- **Small evaluation sets.** Sweeps use 20 volumes per condition (4 in the smoke preset); the test set has 37 volumes and 2,009 edges.
- **Distribution difference between splits.** The test split has a much higher positive-edge rate (18.3%) and fewer candidate edges than train and validation, so test F1 is not comparable to validation F1, and the effect of prevalence on thresholds is not separated from model quality.
- **Single imaging model.** All noise comes from one PSF + Poisson + Gaussian model; real scanners also have ring artefacts, beam hardening and non-Gaussian structure.

6 Limitations

- **Synthetic-to-real gap.** Everything here concerns synthetic phantoms with idealised super-ellipsoid pores and cylindrical necks. The results show whether this class of model can correct the simulated failure modes; they are not evidence of performance on any real material or scanner, which requires separate validation.
- **Physical resolution limit.** No learned model can recover a neck that has been entirely erased by the blur; apparent recovery of such cases would more likely reflect priors of the generator than real information. The out-of-distribution results (Section 4.8) are a partial check on that risk.
- **Class imbalance.** Connected edges are the minority (about 4–5% of train/val candidate edges), so all reported metrics are for the connected class and accuracy is deliberately not used.

7 Conclusion

We set out to test whether a learned correction can repair the systematic errors that classical thresholding and connected-component analysis make when determining pore connectivity in noisy 3D images. On exactly known synthetic ground truth, the answer at the level of individual connections is yes: raw 26-CC labels nearly all true connections but is wrong about two thirds of its positive predictions (F1 0.468), while graph-based models such as GraphSAGE and AE+GraphSAGE reach F1 of about 0.95–0.96 with precision of 0.92–0.95 and recall of 0.97–0.99. They also keep F1 between 0.72 and 0.88 where raw CC falls to 0.48–0.55 across medium to very-high noise, and they generalise well to more crowded volumes, larger pores and larger clusters.

The gain is attributable to geometric neck features and graph context, not to the autoencoder. The frozen AE embedding added nothing measurable to graph models, harmed an MLP, and made the GCN fusion model collapse under unseen noise; an end-to-end 3D CNN (F1 0.897) shows the image does contain useful information when trained supervised. Finally, improvements at the edge level did not carry over to cluster agreement or percolation accuracy on our 37-volume test set, where the unlearned CC baseline remained competitive (ARI 0.965, percolation accuracy 0.919). The practical message is therefore a measured one: learned graph models are a promising tool for removing false connections, but their value for connectivity-level properties must be shown with connectivity-level objectives and more extensive evaluation before it is claimed.

7.1 Future work

- Diagnose the percolation errors by volume type (ordinary, percolating, near-miss) and tune thresholds or train losses directly for cluster and percolation accuracy.
- Repeat all models over multiple seeds, enlarge the test set, and match class prevalence between validation and test.
- Train the image branch end-to-end (CNN encoder + GraphSAGE) or pre-train it with objectives targeted at necks (contrastive or neck-focused) instead of a frozen reconstruction AE.
- Investigate the F1 = 0 collapse at positive threshold offsets and the recall drop at wide necks.
- Validate on real μCT scans and compare with pore-network tools such as PoreSpy.

8 Acknowledgment

SMILE Facility IITR for data.

9 Video

https://youtu.be/b7M5xkNPdG8

10 References

1. Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. ICLR 2017. arXiv:1609.02907.
2. Hamilton, W. L., Ying, R. and Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. NeurIPS 2017. arXiv:1706.02216.
3. Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. arXiv:1312.6114.
4. Fey, M. and Lenssen, J. E. (2019). Fast Graph Representation Learning with PyTorch Geometric. ICLR Workshop on Representation Learning on Graphs and Manifolds. arXiv:1903.02428.
5. Gostick, J. T. (2017). Versatile and Efficient Pore Network Extraction Method Using Marker-Based Watershed Segmentation. Physical Review E, 96, 023307.
6. Gostick, J. T., Khan, Z. A., Tranter, T. G., Kok, M. D. R., Agnaou, M., Sadeghi, M. and Jervis, R. (2019). PoreSpy: A Python Toolkit for Quantitative Analysis of Porous Media Images. Journal of Open Source Software, 4(37), 1296.
7. Zhao, Q. and Xiao, H. An End-to-End Differentiable, Graph Neural Network–Embedded Pore Network Model for Permeability Prediction. arXiv:2509.13841.
8. Cai, C., Vlassis, N., Magee, L., Ma, R., Xiong, Z., Bahmani, B., Wong, T.-F., Wang, Y. and Sun, W. C. (2021). Equivariant Geometric Learning for Digital Rock Physics: Estimating Formation Factor and Effective Permeability Tensors from Morse Graph. arXiv:2104.05608.
9. Zhang, M. and Chen, Y. (2018). Link Prediction Based on Graph Neural Networks. NeurIPS 2018. arXiv:1802.09691.
10. Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics.
11. Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. ACM SIGGRAPH Computer Graphics.
12. Hubert, L. and Arabie, P. (1985). Comparing partitions. Journal of Classification.