TOPOLOGYUNDERSTANDINGNET (TUN): GEOMETRY-CONDITIONED CLASSIFICATION OF PERSISTENCE DIAGRAM POINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Persistence diagrams (PDs) summarize topological changes across a filtration, but deciding which individual entries reflect source structure rather than sampling artifacts remains difficult because persistence alone does not encode the geometry that generated the diagram. We formulate this problem as point-wise significance classification and construct a supervised benchmark spanning planar shapes, CAD models, triply periodic minimal surfaces, and zeolite frameworks. Its labels combine known topological counts with instance-specific geometric review, supported by a conditional analysis of an early-birth criterion. We then propose TopologyUnderstandingNet (TUN), which models interactions among diagram entries and conditionally refines each prediction using a point-cloud representation of the source geometry. Experiments show that TUN achieves the highest average classification performance and improves over its matched PD-only counterpart across all four categories. In aggregate CAD source-identity retrieval, equal-size subdiagrams selected by TUN also outperform persistence ranking, indicating that geometry-conditioned predictions preserve source information not captured by persistence alone.

1 INTRODUCTION

Persistent homology tracks connected components, loops, and voids across scale and records their birth and death values in a persistence diagram (PD) (Edelsbrunner & Harer, 2010). For point clouds, PDs support geometric analysis in robotics (Zhang & Singh, 2014), medical imaging (Singh et al., 2023), and computational biology (Xia & Wei, 2014). Their use, however, requires deciding which individual entries capture source structure rather than sampling artifacts. Appendix A reviews the required mathematical background.

The standard heuristic treats high-persistence entries as significant and points near the diagonal as noise (Fasy et al., 2014). Persistence alone does not always resolve this distinction (Bubenik et al., 2020): the decision can also depend on absolute birth and death scales, the other diagram entries, and the source geometry. Recent learned filtering combines point-cloud and PD representations to predict a diagram-specific threshold or soft mask for downstream recognition (Kudeshia & Poovvancheri, 2026). This motivates a predictor that uses the same context while returning a label for every retained entry.

We formulate this target as supervised classification of retained entries in one-dimensional persistence diagrams (1-PDs) computed by alpha filtration (Edelsbrunner & Harer, 2010). The benchmark initializes labels from geometric/topological metadata and persistence ranking, followed by an early-birth review; "significant" denotes this operational task label. Because the map from a filtered object to its PD is not generally injective, the source point cloud supplies information unavailable from the diagram alone. Persistence-threshold methods (Fasy et al., 2014; He et al., 2023) provide reference rules.

Figure 1 provides a geometric check using volume-optimal representative cycles (Obayashi, 2018). The two positive entries in panel (c) yield cycles supported near the sampled shape's loops, whereas the high-persistence negative entry in panel (d) yields a cycle that departs substantially from the surface. These representatives illustrate, but do not generate, the benchmark labels. Figure 2 gives the

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.

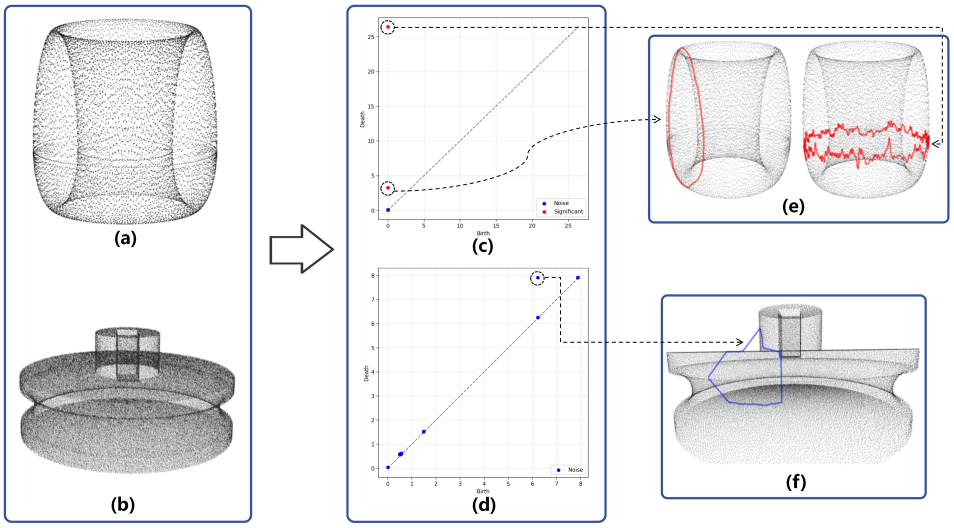

Figure 1: An illustration of the operational point-labeling task in 1-PDs. (a) and (b): point clouds sampled from CAD models with first Betti numbers 2 and 0. (c) and (d): the corresponding 1-PDs and benchmark labels, where red and blue denote positive and negative entries. (e): volume-optimal representative cycles for the two positive entries in (c), both supported near loops of the sampled shape. (f): a representative cycle for the most persistent negative entry in (d); much of the cycle lies away from the sampled surface.

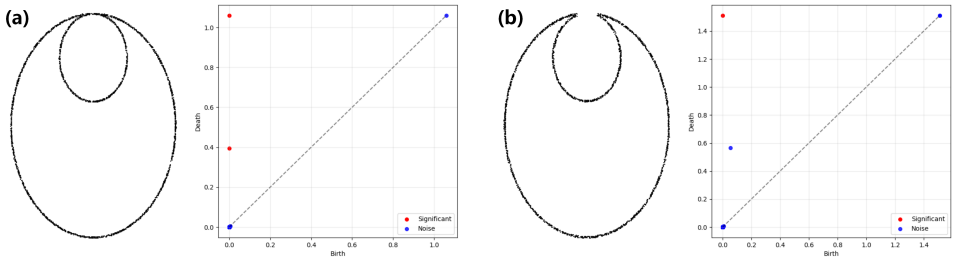

Figure 2: Examples illustrating the ambiguity of point labeling in 1-PDs. (a) and (b): two 1-PDs derived from similar planar shapes but assigned distinct operational label sets.

complementary case: visually similar 1-PDs receive different operational labels when their source geometries differ.

We introduce **TopologyUnderstandingNet (TUN)** for this point-wise binary classification task. TUN learns a default decision from masked diagram context and uses source geometry to apply a gated, candidate-specific correction. The four-category benchmark initializes labels from known topological counts and persistence ranking, then applies an early-birth review before training and testing.

Our contributions are: (1) We formulate a supervised benchmark for predicting operational labels on retained points in 1-PDs across four point-cloud categories and give a conditional early-birth separation result for its label construction. (2) We develop TopologyUnderstandingNet (TUN), which combines masked PD encoding and point-cloud context through gated residual conditioning. (3) Across ten runs, TUN reaches 0.9965 unweighted mean F1, compared with 0.9933 for DeepSets, and improves fixed-budget CAD retrieval mAP from 0.3215 to 0.3540 over top-persistence.

2 Related Work

2.1 Topological Data Analysis (TDA)

Topological data analysis studies the shape of data through constructions such as persistent homology (Edelsbrunner & Harer, 2008; 2010; Zomorodian & Carlsson, 2004; Wasserman, 2018). Persistent homology tracks the birth and death of features across a filtration and summarizes them in PDs. Prior work covers algorithms for several filtration families (Edelsbrunner & Harer, 2010; Maria et al., 2014), stability under perturbations (Cohen-Steiner et al., 2005), and applications in molecular science (Xia & Wei, 2014; Bramer & Wei, 2020), materials science (Hiraoka et al., 2016; Obayashi et al., 2022), geometric design and processing (Brüel-Gabrielsson et al., 2020; Dong et al., 2022; He et al., 2023; Chen & Lin, 2025), and medical image processing (Brito-Pacheco et al., 2026; Singh et al., 2023).

TDA has also been combined with machine learning for feature extraction and classification (Pun et al., 2022; Barnes et al., 2021). PersLay (Carrière et al., 2020) maps PDs to vector representations through learnable functions. Persformer (Reinauer et al., 2021) instead treats each diagram as an unordered point set and applies a Transformer without fixed vectorization. For point clouds, TopologyNet learns a direct map from geometry to persistence images (Zhou et al., 2022), whereas TopoLayer encapsulates PD vectorization in a module that augments point-cloud backbones (Guan et al., 2025). TOPF constructs node-level topological features on the source point cloud from algebraic-topological and differential-geometric quantities (Grande & Schaub, 2025). These methods produce diagram-level or source-point features rather than labels for individual PD entries, but their set-processing principles motivate the DeepSets and Set Transformer baselines used in our evaluation (Zaheer et al., 2017; Lee et al., 2019). Other studies incorporate PDs into loss functions to impose topological constraints during training (Wong & Vong, 2021; Hu et al., 2025; Jignasu et al., 2024).

These methods use a PD as a diagram-level representation or derive features on source points. TUN instead predicts a label for each retained PD entry.

2.2 Topology Understanding of PDs

The significance of points in PDs has been studied using statistical and geometric approaches. A common heuristic ranks points by persistence and declares the top k (or the single most persistent point) as significant, discarding the remainder as noise (Brüel-Gabrielsson et al., 2020; Dong et al., 2022). A statistical alternative constructs a confidence set around the diagonal through bootstrap resampling (Fasy et al., 2014). The result depends on the point-cloud density and user-selected parameters. Clustering-based strategies, such as k-means with $k = 2$, partition PD points into significant and noise classes (He et al., 2023; Chen & Lin, 2025); they cannot represent a diagram in which all retained entries belong to the same class. Both approaches use persistence as the primary importance measure. Persistence can omit relevant information about the geometry that generated the diagram (Bubenik et al., 2020): a low-persistence feature may correspond to a small geometric structure, while an outlier can produce a high-persistence artifact.

TopoGAT (Kudeshia & Poovvancheri, 2026) jointly encodes a point cloud and its H_1 and H_2 diagrams, predicts diagram-specific thresholds, and applies differentiable soft masks for classification and part segmentation. Its selection is optimized for the downstream task. TUN instead predicts supervised binary labels for every retained H_1 entry of a fixed alpha-filtration diagram, conditioned on source geometry.

Geometry can also enter earlier in the persistent-homology pipeline. Distance-to-measure and kernel-distance filtrations improve robustness before persistence is computed (Chazal et al., 2018); latent-graph and landmark-based approaches incorporate statistical structure before or during topological inference (Aupetit, 2005; Gamble & Heo, 2010). Adaptive filtration learning likewise uses point-cloud geometry to construct more informative diagrams (Nishikawa et al., 2023). TUN instead addresses a post-filtration interpretation problem: the alpha-filtration diagram is held fixed, and source geometry conditions the label assigned to each retained entry.

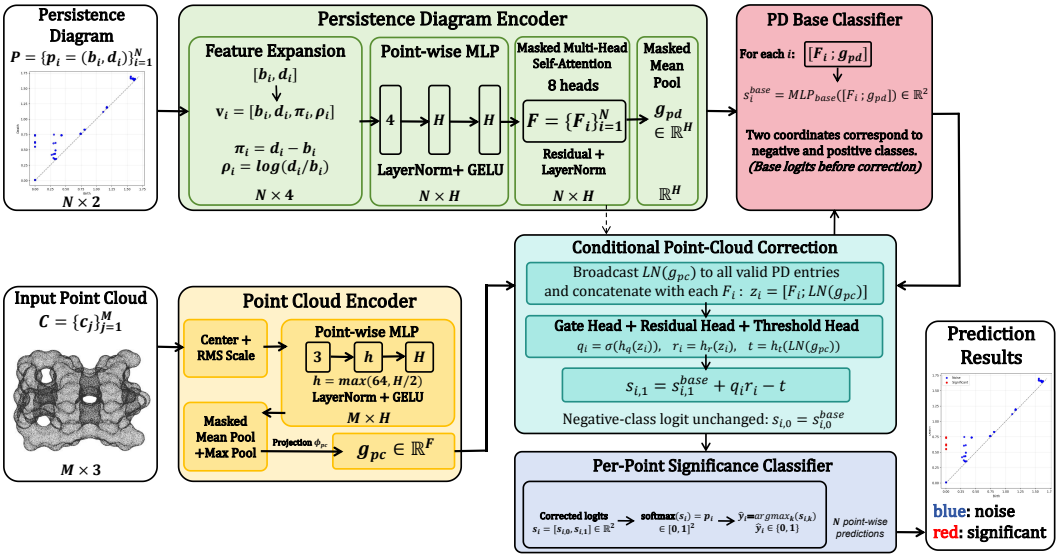

Figure 3: TUN architecture. Each retained birth–death pair is expanded to four descriptors and passed through a point-wise MLP, masked multi-head self-attention, residual normalization, and masked mean pooling. The resulting per-point features and global PD descriptor produce base two-class logits. In parallel, the source point cloud is centered, RMS-scaled, encoded point-wise, and summarized by masked mean and max pooling. Its projected descriptor drives a point-wise gated residual head and a sample-level threshold head that correct only the positive-class logit. A point-cloud count head and the summed positive probabilities provide the two count-supervision paths indicated by dashed arrows.

3 METHODS

TUN predicts labels for individual entries in PDs computed from alpha filtrations of point clouds. It uses two input modalities: a masked PD encoder provides candidate-specific and diagram-level features, and a point-cloud encoder supplies geometric context. The PD branch first produces point-wise base logits, which the point-cloud descriptor then adjusts conditionally. Figure 3 shows the data path and the training-time count supervision. Appendix B gives the implementation details and hyperparameters.

3.1 TOPOLOGICAL BASIS OF THE LABEL CONSTRUCTION

We first state the conditions under which an early-birth persistence rule identifies the one-dimensional classes inherited from an underlying shape. Let $X \subset \mathbb{R}^3$ be a compact tame set with positive reach τ and finite first Betti number β_1. Let C be a finite point cloud such that $d_H(C, X) \le \varepsilon < \tau/4$. The alpha complex of C is homotopy equivalent, at each filtration scale, to the corresponding union of balls (Edelsbrunner, 1995). The equality $\|d_X - d_C\|_\infty = d_H(X, C)$ then gives, by persistence-diagram stability, $d_B(D_1(X), D_1(C)) \le \varepsilon$ (Cohen-Steiner et al., 2005; Chazal et al., 2009).

Proposition 3.1 (Early-birth separation). *Under the conditions above, $D_1(C)$ contains β_1 classes satisfying*

$$b \le \varepsilon, \qquad d \ge \tau - \varepsilon, \qquad d - b \ge \tau - 2\varepsilon. \tag{1}$$

Every other class in $D_1(C)$ with $b \le \varepsilon$ has $d - b \le 2\varepsilon$. Consequently, the β_1 most persistent classes among the early-born entries are precisely the classes matched to the degree-one homology of X.

Proof. For $0 \le r < \tau$, nearest-point projection makes the offset X^r homotopy equivalent to X. Thus the ideal offset persistence has β_1 classes born at zero that survive to at least τ, with no additional degree-one births before τ. A bottleneck matching of cost at most ε maps these classes to entries satisfying Eq. (1). Any remaining sampled entry with $b \le \varepsilon$ cannot match an ideal class born at or

after τ and must instead match the diagonal, which implies $d - b \leq 2\varepsilon$. Since $\varepsilon < \tau/4$, we have $\tau - 2\varepsilon > 2\varepsilon$. □

The implementation uses GUDHI alpha-complex filtration values, which are squared radii (Maria et al., 2014). Writing $(\widetilde{b}, \widetilde{d}) = (b^2, d^2)$, an inherited entry satisfies $\widetilde{d} - \widetilde{b} \geq \tau^2 - 2\tau\varepsilon$, whereas an early-born diagonal-matched entry satisfies $\widetilde{d} - \widetilde{b} \leq 8\varepsilon^2$. The same condition $\varepsilon < \tau/4$ makes the first bound strictly larger, supporting ranking by native squared-radius persistence among early-born entries.

For the mesh-based data pipeline, let M denote the source mesh. If $d_H(M, X) \leq \delta_M$, the Poisson-disk sample covers M within δ, and each added perturbation has norm at most ν, then $d_H(C, X) \leq \delta_M + \delta + \nu$. Under the stated positive-reach and Hausdorff-error assumptions, this decomposition motivates the early-birth review rule. The per-instance quantities are not estimated, so the proposition does not certify individual labels.

3.2 Model Architecture

Persistence Diagram Encoder. Let $m_i^{\mathrm{pd}} \in \{0, 1\}$ indicate whether the i-th retained entry is valid. For each raw pair $p_i = (b_i, d_i)$, TUN constructs

$$\pi_i = d_i - b_i, \quad \rho_i = \log(d_i / \max(b_i, \varepsilon)), \quad v_i = [b_i, d_i, \pi_i, \rho_i] \in \mathbb{R}^4. \tag{2}$$

where $\varepsilon > 0$ prevents division by zero. In the implementation, birth and death values are clipped to a bounded range and non-finite derived values are replaced before encoding. The four-dimensional vector v_i is processed independently by a two-layer point-wise MLP with LayerNorm, GELU, and dropout. The resulting tokens are passed to an eight-head masked self-attention layer. The residual sum of the input tokens and the attention output is normalized by LayerNorm. Padding tokens are zeroed before attention, excluded as attention keys, and masked again after normalization. The encoder produces contextualized features F_i for valid PD entries and a permutation-invariant global descriptor

$$g_{\mathrm{pd}} = \frac{\sum_i m_i^{\mathrm{pd}} F_i}{\max\left(1, \sum_i m_i^{\mathrm{pd}}\right)}. \tag{3}$$

PD Base Classifier. The base classifier is applied to each valid PD entry using both its contextualized token and the global diagram descriptor. Specifically,

$$s_i^{\mathrm{base}} = \mathrm{MLP}_{\mathrm{base}}\left([F_i; g_{\mathrm{pd}}]\right) \in \mathbb{R}^2, \tag{4}$$

where the two output coordinates correspond to the negative and positive classes. This stage is the complete prediction path of the matched PD-Encoder only model used in the modality ablation.

Point Cloud Encoder. To incorporate geometric context, we use a masked PointNet-style encoder inspired by (Qi et al., 2017). Two-dimensional inputs are embedded in $\mathbb{R}^3$ with a zero z coordinate. Let m_j^{pc} denote the validity mask for point $c_j \in \mathbb{R}^3$. We center the valid points using

$$\bar{c} = \frac{\sum_j m_j^{\mathrm{pc}} c_j}{\max(1, \sum_j m_j^{\mathrm{pc}})}, \quad a = \left(\frac{\sum_j m_j^{\mathrm{pc}} \|c_j - \bar{c}\|_2^2}{\max(1, \sum_j m_j^{\mathrm{pc}})} + 10^{-8}\right)^{1/2}. \tag{5}$$

We then scale the points as $\widetilde{c}_j = m_j^{\mathrm{pc}}(c_j - \bar{c}) / \max(a, 10^{-4})$. A shared point-wise MLP maps the three coordinates through an intermediate width $h = \max(64, H/2)$ to width H, with LayerNorm and GELU after each linear transformation. If $h_j^{\mathrm{pc}} \in \mathbb{R}^H$ denotes the resulting feature, the two masked pooling outputs are

$$u_{\mathrm{mean}} = \frac{\sum_j m_j^{\mathrm{pc}} h_j^{\mathrm{pc}}}{\max(1, \sum_j m_j^{\mathrm{pc}})}, \quad u_{\max} = \max_{j : m_j^{\mathrm{pc}} = 1} h_j^{\mathrm{pc}}, \quad g_{\mathrm{pc}} = \phi_{\mathrm{pc}}([u_{\mathrm{mean}}; u_{\max}]) \in \mathbb{R}^F, \tag{6}$$

where ϕ_{pc} is the projection, LayerNorm, GELU, and dropout block. Masking excludes padded points and their placeholder coordinates from the descriptor.

We also pass this descriptor to a point-cloud count head, $\widehat{c}_{\mathrm{pc}} = \mathrm{softplus}(u_{\mathrm{pc}}(g_{\mathrm{pc}}))$. This head supplies an auxiliary training signal and is not an input to the per-point classifier.

Conditional Point-Cloud Correction. The PD branch first predicts the base logits in Eq. (4). We normalize the point-cloud descriptor with LayerNorm and broadcast it across the valid PD entries. For each entry, TUN forms $z_i = [F_i; \mathrm{LN}(g_{\mathrm{pc}})]$. Three heads operate on this conditional representation:

$$r_i = h_r(z_i), \quad q_i = \sigma(h_q(z_i)) \in (0,1), \quad t = h_t(\mathrm{LN}(g_{\mathrm{pc}})), \tag{7}$$

where h_r is the residual adapter, h_q is the gate head, and h_t is the sample-level threshold head. The first two heads produce one value per valid PD entry, whereas the threshold head produces one scalar shared by all entries in a sample. The positive-class logit is corrected by

$$s_{i,1} = s_{i,1}^{\mathrm{base}} + q_i r_i - t, \tag{8}$$

while the negative-class logit is unchanged. The residual and threshold heads are initialized to zero, and the gate bias is initialized to -2, so the point-cloud branch initially makes a small correction. For each seed, TUN is warm-started from the corresponding PD-Encoder only checkpoint, and the transferred parameters are frozen for the first 10 epochs.

Significance Classifier. The final probabilities are $p_i = \mathrm{softmax}(s_i)$, and the predicted label is $\arg\max_{k\in\{0,1\}} s_{i,k}$. All operations and metrics exclude padded entries. The shared point-cloud descriptor conditions each candidate-specific token F_i, yielding one two-class logit vector per retained PD point.

3.3 Loss Function

We adopt a weighted focal loss (Lin et al., 2017) because significant points are rare relative to noise points. For each PD point i, let $\mathbf{s}_i \in \mathbb{R}^2$ denote the two-class logits, let $\mathbf{p}_i = \mathrm{softmax}(\mathbf{s}_i)$ denote the class probabilities, and let $y_i \in \{0,1\}$ be the ground-truth label. We write $p_{t,i} = \mathbf{p}_i[y_i]$ for the probability assigned to the true class and define the weighted cross-entropy value $z_i = w_{y_i}(-\log p_{t,i})$. We then apply the focal modulator to this weighted value: $\ell_i = \alpha\left(1-\exp(-z_i)\right)^{\gamma} z_i$. Here $w_{y_i} \in \mathbb{R}_+$ is the class weight of label y_i (with $w_1 > w_0$), $\alpha > 0$ is a global scaling coefficient, and $\gamma \geq 0$ down-weights easy examples. Because PDs are padded to a fixed length, we compute the loss only over valid points using a mask $M_i \in \{0,1\}$. We normalize by the number of valid points:

$$\mathcal{L} = \frac{1}{\sum_i M_i} \sum_i M_i \, \ell_i.$$

We supplement the point loss with two count terms and a residual regularizer. Let $\hat{c}$ be the sum of predicted positive probabilities over valid entries and let c be the target positive count:

$$\widehat{c} = \sum_i M_i p_{i,1}, \qquad c = \sum_i M_i y_i. \tag{9}$$

The point-cloud count head gives a second estimate $\hat{c}_{\mathrm{pc}}$. Let $\mathcal{L}_{\mathrm{res}}$ denote the mean squared residual output over valid entries. We optimize the following objective:

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{focal}} + \lambda_c \mathcal{L}_{\mathrm{count}} + \lambda_{\mathrm{pc}} \mathcal{L}_{\mathrm{pc-count}} + \lambda_r \mathcal{L}_{\mathrm{res}}.$$

Both count terms use smooth-L_1 loss on log-counts. For the PD prediction, the loss compares $\log(1+\hat{c})$ with $\log(1+c)$; for the point-cloud head, it compares $\log(1+\hat{c}_{\mathrm{pc}})$ with $\log(1+c)$. The residual penalty is applied to r_i only on valid entries. We use $\lambda_c = 0.1$, $\lambda_{\mathrm{pc}} = 0.05$, and $\lambda_r = 10^{-4}$ for the full TUN model. The matched PD-Encoder only model uses the PD count term but has no point-cloud count or residual branch. All point-level terms exclude padded entries through the valid-entry mask.

3.4 Computational Complexity

Let N denote the retained number of PD entries, let M denote the retained number of point-cloud samples, and let H denote the hidden width. PD self-attention costs $O(N^2H)$ time and $O(N^2)$ attention memory. The point-cloud encoder uses shared point-wise layers and masked pooling, which cost $O(MH^2)$ time and $O(MH)$ activation memory for fixed-width layers. The conditional correction is point-wise and costs $O(NH^2)$. With the implemented caps $N = 100$ and $M = 4{,}096$, the point-cloud branch dominates model-side activation storage. Alpha-complex and persistence-diagram construction remain separate preprocessing costs.

4 RESULTS

We evaluate TUN and the supervised comparison methods over ten independent runs. All experiments were conducted on a server equipped with one NVIDIA GeForce RTX 4090 GPU (24GB VRAM). The system was powered by a 16-core Intel Xeon Platinum 8352V CPU @ 2.10 GHz and utilized 120GB of system RAM.

4.1 DATASETS

We evaluate four categories of point-cloud data: (i) planar shapes; (ii) CAD models drawn from the ABC dataset [1] (Koch et al., 2019) and the Thingi10k dataset [2] (Zhou & Jacobson, 2016); (iii) triply periodic minimal surface (TPMS) structures [3] (Wang et al., 2025); and (iv) zeolite crystal frameworks [4] (Baerlocher et al., 2007). The categories differ in geometric regularity and loop structure. Planar constructions provide controlled loop counts and visually similar diagrams; CAD models introduce heterogeneous surfaces and boundaries; TPMS structures contain repeated multi-scale tunnels; and zeolite frameworks provide porous crystalline geometries. Our dataset comprises 4,990 complete PD–point-cloud–label records: 3,000 for training, 500 for validation, and 1,490 for testing. The test set contains 250 planar, 500 CAD, 500 TPMS, and 240 zeolite records. Each record is a separately generated observation, and the three observation-level partitions are disjoint. Source shapes have first Betti number below 100, and source point clouds contain from hundreds to 100,000 points. Each PD is truncated or padded to 100 entries, and each point cloud to 4,096 points; valid-entry masks exclude padding from attention, pooling, loss, and evaluation.

For each instance, we use Poisson-disk sampling on the underlying geometric model and add a bounded point perturbation when specified. We then construct the alpha complex and compute its 1D persistence diagram with GUDHI. Examples appear in Figure 7 in Appendix D. The known geometric/topological count initializes the target: retained H_1 entries are ranked by native squared-radius persistence $\widetilde{d}_i - \widetilde{b}_i$, and the top β_1 entries receive the positive label. The geometric construction supplies β_1. For a connected orientable surface of genus g with q boundary components, $\beta_1 = 2g$ when $q = 0$ and $\beta_1 = 2g + q - 1$ when $q > 0$. One author then reviews each exported label file against its source geometry and corrects entries according to the early-birth criterion motivated by Proposition 3.1. Because sampling scales vary by instance, this step does not use a single global birth-time threshold. We use the corrected files as the final labels for training and testing. Appendix C reports class balance and the extent to which the final labels differ from the persistence-ranked initialization.

The model does not receive β_1 or the manual correction decisions as inputs. We evaluate prediction against the final labels.

4.2 EVALUATION METRICS

We use F1-score as the primary metric because positive entries are rare. Checkpoints minimize held-out validation loss. Each supervised experiment is repeated ten times, and we report the mean and sample standard deviation across runs. Appendix E reports the corresponding confusion counts.

Comparison Methods. The deterministic references are 2-means clustering and a confidence-set (CS) rule at confidence level 0.5. The SVM and MLP baselines are point-wise classifiers whose input consists of (b_i, d_i). DeepSets and Set Transformer are adapted to return one logit pair per PD entry and receive the same four PD descriptors as TUN. These models test whether permutation-aware diagram context is sufficient without access to the source geometry. Full implementation details are provided in Appendix B.

Table 1: Point-level F1-score across comparison methods. Deterministic rules are evaluated once; learning methods report mean ± sample standard deviation over ten runs. CS uses confidence level 0.5 and $B = 500$ bootstrap resamples. PD-Encoder only is the modality-matched ablation of TUN. The best result in each column is bold.

Method	Planar	CAD	TPMS	Zeolite
2-means	0.3071	0.1627	0.3224	0.5357
CS (0.5)	0.2525	0.1263	0.0000	0.0000
MLP	0.5143±0.1207	0.2693±0.0923	0.3243±0.1502	0.0743±0.1017
SVM	0.9296±0.0016	0.8701±0.0024	0.9653±0.0007	0.8880±0.0015
DeepSets	0.9874±0.0056	**0.9945±0.0031**	0.9926±0.0049	0.9987±0.0007
Set Transformer	0.9849±0.0055	0.9919±0.0102	0.9951±0.0031	0.9987±0.0010
PD-Encoder only	0.9878±0.0032	0.9883±0.0113	0.9965±0.0027	0.9981±0.0017
TUN	**0.9941±0.0019**	0.9935±0.0029	**0.9986±0.0014**	**0.9996±0.0007**

4.3 Results and Analysis

Point-level predictive performance. TUN achieves the highest F1-score on Planar, TPMS, and Zeolite; DeepSets is higher on CAD by 0.0010 (Table 1). Its unweighted four-category mean F1 is 0.9965, compared with 0.9933 for DeepSets, 0.9926 for Set Transformer, and 0.9927 for PD-Encoder only. The sample standard deviation of the seed-wise four-category average is also lowest for TUN: 0.0010, versus 0.0032, 0.0032, and 0.0036 for the three PD-only models.

Comparison with deterministic and point-wise baselines. The deterministic rules yield complementary error profiles. The 2-means rule attains 0.3320 unweighted mean F1: recall reaches 0.7622–0.9690 on Planar, CAD, and Zeolite, but assigning a complete cluster to the positive class produces many false positives. CS(0.5) yields precision above 0.97 on Planar and CAD but recall below 0.15; it predicts no positive entries on TPMS or Zeolite. A forced two-cluster partition and a single bootstrap threshold therefore do not transfer across these geometries.

Among the point-wise learned references, the RBF SVM is stronger than the MLP on every category and reaches 0.9133 unweighted mean F1. Absolute birth–death location carries substantial predictive signal, particularly on TPMS (0.9653), while TUN is higher by 0.0333–0.1234 across categories. The MLP varies more across runs, with category-wise standard deviations of 0.0923–0.1502.

Effect of the conditional full TUN extension. Relative to the same-seed PD-Encoder only model, TUN raises mean F1 by 0.0063, 0.0052, 0.0022, and 0.0015 on Planar, CAD, TPMS, and Zeolite, respectively. Averaged across the four categories, the gain is 0.0038 and is positive for all ten paired seeds. The improvement is primarily precision-driven. Among TUN and the three set-structured neural comparisons, TUN produces the fewest false positives on every category: 12.2, 8.1, 3.4, and 1.3, corresponding to reductions of 54.1%, 47.7%, 89.1%, and 86.0% relative to the lowest-false-positive alternative in each column, while retaining recall between 0.9915 and 0.9994. Tables 4 and 5 report the full error composition.

Fixed-budget CAD source-identity retrieval. We next evaluate selected H1 entries as descriptors using 500 separately curated manifold, watertight, single-component Thingi10K meshes (Zhou & Jacobson, 2016). A fixed split reserves 100 meshes for protocol validation and uses 400 clean gallery meshes for retrieval from independently sampled, synthetically corrupted queries.

TUN and top-persistence retain the same predicted number k_{TUN} of entries, which isolates subset identity at a fixed representation budget. Random-matched and Count-only control for arbitrary subsets and predicted cardinality. Table 2 averages nine corruptions over ten frozen TUN checkpoints. Appendix E.2 specifies the diagram distance, metrics, per-corruption results, and same-β_1 analysis.

[1] https://deep-geometry.github.io/abc-dataset

[2] https://github.com/thingi10k/thingi10k

[3] https://github.com/W-W-M/Morphology-Learning

[4] `http://america.iza-structure.org/IZA-SC/ftc_table.php`

Table 2: Full-gallery CAD retrieval over nine synthetic query corruptions. Mean ± sample standard deviation over ten frozen TUN checkpoints; Random-matched is averaged over five subset draws.

Selection	**R@1**	**R@5**	**mAP**
Count-only	0.0180 ± 0.0019	0.0763 ± 0.0056	0.0624 ± 0.0027
Random-matched	0.0424 ± 0.0024	0.1178 ± 0.0062	0.0885 ± 0.0040
Top-persistence@k_{TUN}	0.2618 ± 0.0173	0.3750 ± 0.0226	0.3215 ± 0.0197
TUN	$\mathbf{0.2934 \pm 0.0222}$	$\mathbf{0.4094 \pm 0.0267}$	$\mathbf{0.3540 \pm 0.0240}$

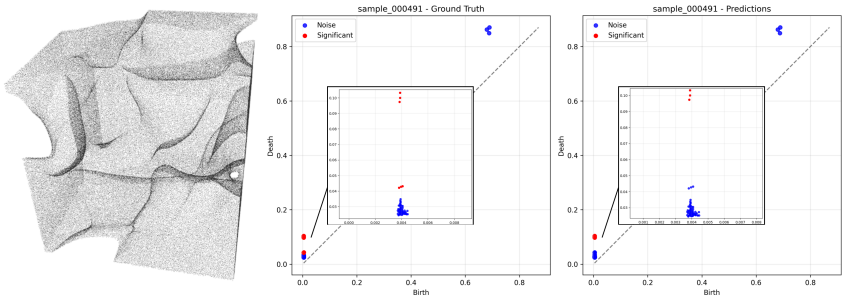

Figure 4: Illustrating a failure of TUN model. In this 1-PD of TPMS test sample, three points corresponding to small-scale loops in the original model have small persistence (marked in red in Ground Truth, while blue in Prediction), which makes them hard to distinguish from noise points.

Across all conditions, TUN exceeds the fixed-budget top-persistence selector by 0.0325 mAP, 0.0316 R@1, and 0.0344 R@5, with a positive checkpoint-wise aggregate difference in all ten runs. The mAP gain is largest under 1% and 5% outliers ($+0.2332$ and $+0.0958$). In the same-β_1 analysis, TUN reaches 0.5349 ± 0.0219 mAP versus 0.5058 ± 0.0190 for top-persistence; Appendix E.2 reports all corruption-specific results.

4.4 Limitations

Firstly, the experiments cover H_1 alpha-filtration diagrams with at most 100 entries and globally encoded point clouds with at most 4,096 samples. Moreover, in this study, we focus on 1-PDs because extending the label construction to 2-PDs (H_2 persistence diagrams) requires a dedicated dataset with H_2-specific geometric/topological metadata and instance-level labels; it is not obtained by changing the model input dimension alone. In addition, as shown in Figure. 4, small loops can overlap the persistence range of sampling artifacts; denser sampling and candidate-specific geometric alignment are directions for future work.

5 Conclusion

We introduced TUN for point-wise interpretation of one-dimensional persistence diagrams. TUN learns contextual representations of retained diagram entries and uses gated residual conditioning from the source point cloud to incorporate geometric information unavailable from the diagram alone. Together with the four-category benchmark and conditional early-birth analysis, this formulation provides a supervised framework for assigning operational labels to individual PD entries. Across the evaluated geometries, source conditioning consistently improves the matched PD-only model, while fixed-budget CAD retrieval shows that TUN-selected subdiagrams improve aggregate source-identity recovery over same-budget top-persistence under the stated synthetic corruptions.

Future work includes creating a dedicated 2-PD dataset with H_2-specific source geometries, metadata, and reviewed labels, and then adapting TUN to this setting. We will also investigate alternative source representations, including images and voxel grids, in place of point clouds. These extensions may broaden the range of supported data modalities and offer more interpretable links between source structure and individual PD predictions.

SOFTWARE AND DATA

The code is available in the supplementary materials.

IMPACT STATEMENT

This work studies per-point classification in persistence diagrams and uses public geometric and materials datasets. Its primary impacts concern scientific analysis and geometric data processing.

AI DISCLOSURE

In this work, Generative AI tools were used to support language editing, experimental planning, and software development. The authors reviewed the manuscript and the experiments, and any code produced by LLMs was checked and tested for correctness. The authors take responsibility for the final content.

REFERENCES

Michael Aupetit. Learning topology with the generative gaussian graph and the EM algorithm. In *Advances in Neural Information Processing Systems*, volume 18, 2005.

Ch Baerlocher, Lynne B McCusker, and David H Olson. *Atlas of Zeolite Framework Types*. Elsevier, 6 edition, 2007. ISBN 9780444530646. doi: 10.1016/B978-0-444-53064-6.X5186-X.

Danielle Barnes, Luis Polanco, and Jose A Perea. A comparative study of machine learning methods for persistence diagrams. *Frontiers in Artificial Intelligence*, 4:681174, 2021.

David Bramer and Guo-Wei Wei. Atom-specific persistent homology and its application to protein flexibility analysis. *Computational and Mathematical Biophysics*, 8(1):1–35, 2020. doi: 10.1515/cmb-2020-0001.

Daniel Brito-Pacheco, Panos Giannopoulos, and Constantino Carlos Reyes-Aldasoro. Persistent homology in medical image processing: A literature review. *ACM Transactions on Computing for Healthcare*, 7(4):1–31, 2026. doi: 10.1145/3820762.

Rickard Brüel-Gabrielsson, Vignesh Ganapathi-Subramanian, Primoz Skraba, and Leonidas J Guibas. Topology-aware surface reconstruction for point clouds. *Computer Graphics Forum*, 39(5):197–207, 2020. doi: 10.1111/cgf.14079.

Peter Bubenik, Michael Hull, Dhruv Patel, and Benjamin Whittle. Persistent homology detects curvature. *Inverse Problems*, 36(2):025008, 2020.

Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In *International Conference on Artificial Intelligence and Statistics*, pp. 2786–2796. PMLR, 2020.

Frédéric Chazal, David Cohen-Steiner, and André Lieutier. A sampling theory for compact sets in euclidean space. *Discrete & Computational Geometry*, 41(3):461–479, 2009. doi: 10.1007/s00454-009-9144-8.

Frédéric Chazal, Brittany Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Robust topological inference: Distance to a measure and kernel distance. *Journal of Machine Learning Research*, 18(159):1–40, 2018.

Yu Chen and Hongwei Lin. Robust model reconstruction based on the topological understanding of point clouds using persistent homology. *Computer-Aided Design*, pp. 103934, 2025.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pp. 263–271, 2005.

Zhetong Dong, Jinhao Chen, and Hongwei Lin. Topology-controllable implicit surface reconstruction based on persistent homology. *Computer-Aided Design*, 150:103308, 2022.

Herbert Edelsbrunner. The union of balls and its dual shape. *Discrete & Computational Geometry*, 13(1):415–440, 1995. doi: 10.1007/BF02574053.

Herbert Edelsbrunner and John Harer. Persistent homology—a survey. *Contemporary Mathematics*, 453:257–282, 2008. doi: 10.1090/conm/453/08802.

Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, and Aarti Singh. Confidence sets for persistence diagrams. *The Annals of Statistics*, 42(6):2301–2339, 2014. doi: 10.1214/14-AOS1252.

Jennifer Gamble and Giseon Heo. Exploring uses of persistent homology for statistical analysis of landmark-based shape data. *Journal of Multivariate Analysis*, 101(9):2184–2199, 2010.

Vincent Peter Grande and Michael T. Schaub. Point-level topological representation learning on point clouds. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20368–20398. PMLR, 2025. URL https://proceedings.mlr.press/v267/grande25a.html.

Zechao Guan, Shuai Du, and Qingshan Liu. Topolayer: A universal neural network layer for topological feature learning on point clouds using persistent homology. In *2025 IEEE International Conference on Multimedia and Expo*, pp. 1–6. IEEE, 2025. doi: 10.1109/ICME59968.2025.11209425.

Yaqi He, Jiacong Yan, and Hongwei Lin. Robust reconstruction of closed parametric curves by topological understanding with persistent homology. *Computer-Aided Design*, 165:103611, 2023.

Marti A. Hearst, Susan T. Dumais, Edgar Osuna, John Platt, and Bernhard Schölkopf. Support vector machines. *IEEE Intelligent Systems and Their Applications*, 13(4):18–28, 1998. doi: 10.1109/5254.708428.

Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G Escolar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, 2016.

Jiangbei Hu, Ben Fei, Baixin Xu, Fei Hou, Shengfa Wang, Na Lei, Weidong Yang, Chen Qian, and Ying He. Topogen: Topology-aware 3d generation with persistence points. *Computer Graphics Forum*, 44(7):e70257, 2025. doi: 10.1111/cgf.70257.

Anushrut Jignasu, Aditya Balu, Soumik Sarkar, Chinmay Hegde, Baskar Ganapathysubramanian, and Adarsh Krishnamurthy. Sdfconnect: Neural implicit surface reconstruction of a sparse point cloud with topological constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5271–5279, 2024.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9601–9611, 2019.

Prachi Kudeshia and Jiju Poovvancheri. Learning significant persistent homology features for 3D shape understanding. *arXiv preprint arXiv:2602.14228*, 2026. URL https://arxiv.org/abs/2602.14228.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International conference on machine learning*, pp. 3744–3753. PMLR, 2019.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The gudhi library: Simplicial complexes and persistent homology. In *International congress on mathematical software*, pp. 167–174. Springer, 2014.

Fionn Murtagh. Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5-6): 183–197, 1991.

Naoki Nishikawa, Yuichi Ike, and Kenji Yamanishi. Adaptive topological feature via persistent homology: Filtration learning for point clouds. In *Advances in Neural Information Processing Systems*, volume 36, pp. 9131–9143, 2023.

Ippei Obayashi. Volume-optimal cycle: Tightest representative cycle of a generator in persistent homology. *SIAM Journal on Applied Algebra and Geometry*, 2(4):508–534, 2018.

Ippei Obayashi, Takenobu Nakamura, and Yasuaki Hiraoka. Persistent homology analysis for materials research and persistent homology software: Homcloud. *journal of the physical society of japan*, 91(9):091013, 2022.

Chi Seng Pun, Si Xian Lee, and Kelin Xia. Persistent-homology-based machine learning: a survey and a comparative study. *Artificial Intelligence Review*, 55(7):5169–5213, 2022.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

Raphael Reinauer, Matteo Caorsi, and Nicolas Berkouk. Persformer: A transformer architecture for topological machine learning. *arXiv preprint arXiv:2112.15210*, 2021.

Yashbir Singh, Colleen M Farrelly, Quincy A Hathaway, Tim Leiner, Jaidip Jagtap, Gunnar E Carlsson, and Bradley J Erickson. Topological data analysis in medical imaging: current state of the art. *Insights into Imaging*, 14(1):58, 2023.

Weiming Wang, Yanhao Hou, Renbo Su, Weiguang Wang, and Charlie CL Wang. Simultaneous topology optimization of differentiable and non-differentiable objectives via morphology learning: Stiffness and cell growth on scaffold. *Advanced Intelligent Discovery*, 1(1):2400015, 2025.

Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5: 501–532, 2018. doi: 10.1146/annurev-statistics-031017-100045.

Chi-Chong Wong and Chi-Man Vong. Persistent homology based graph convolution network for fine-grained 3d shape segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7098–7107, 2021.

Kelin Xia and Guo-Wei Wei. Persistent homology analysis of protein structure, flexibility, and folding. *International journal for numerical methods in biomedical engineering*, 30(8):814–844, 2014.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Ji Zhang and Sanjiv Singh. LOAM: Lidar odometry and mapping in real-time. In *Proceedings of Robotics: Science and Systems*, July 2014.

Chi Zhou, Zhetong Dong, and Hongwei Lin. Learning persistent homology of 3D point clouds. *Computers & Graphics*, 102:269–279, 2022. doi: 10.1016/j.cag.2021.10.022.

Qingnan Zhou and Alec Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv preprint arXiv:1605.04797*, 2016.

Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. In *Proceedings of the twentieth annual symposium on Computational geometry*, pp. 347–356, 2004.

A MATHEMATICAL FOUNDATIONS OF TDA

This section reviews alpha filtrations, persistent homology, and persistence diagrams.

A.1 ALPHA FILTRATION

The convex hull of $n+1$ affinely independent points $\{u_0, u_1, \ldots, u_n\} \subset \mathbb{R}^N$ is an *n-simplex*, denoted $[u_0, u_1, \ldots, u_n]$. Basic geometric primitives such as vertices, edges, triangles, and tetrahedra correspond to 0-, 1-, 2-, and 3-simplices, respectively. A *simplicial complex* K is a finite set of simplices satisfying two conditions: every face of a simplex in K also belongs to K, and the intersection of any two simplices in K is either empty or a shared face (Edelsbrunner & Harer, 2010). The dimension of K is the largest dimension of any simplex it contains. A simplicial complex L is a *subcomplex* of K if $L \subseteq K$.

The *alpha complex* is a simplicial complex constructed from a point cloud through Voronoi and Delaunay complexes. Given a point cloud $P = \{x_i\}_{i=1}^m \subset \mathbb{R}^n$, the *Voronoi cell* for a point x_i is the region of space closer to x_i than to any other point in P:

$$V_{x_i} = \{x \in \mathbb{R}^n : \|x - x_i\| \leq \|x - x_j\|, \forall j \neq i\},$$

where $\|\cdot\|$ is the Euclidean norm. The dual graph of the Voronoi tessellation defines the *Delaunay complex*. It includes a simplex for every subset of points whose Voronoi cells have a non-empty common intersection:

$$\mathrm{Del}(P) = \left\{ [x_{i_1}, \cdots, x_{i_q}] : \bigcap_{j=1}^{q} V(x_{i_j}) \neq \emptyset \right\}.$$

The alpha complex is a subset of the Delaunay complex controlled by a scale parameter $r \geq 0$. For each point $x \in P$, consider the intersection of its Voronoi cell V_x with a closed ball of radius r centered at x, denoted $R_x(r) = B_x(r) \cap V_x$. The *alpha complex* $\mathrm{Alp}(P, r)$ consists of all simplices whose constituent points have intersecting restricted Voronoi regions:

$$\mathrm{Alp}(P, r) = \left\{ [x_{i_1}, \ldots, x_{i_q}] : \bigcap_{j=1}^{q} R_{x_{i_j}}(r) \neq \emptyset \right\}.$$

The condition $R_x(r) \subseteq V_x$ ensures that $\mathrm{Alp}(P, r)$ is always a subcomplex of $\mathrm{Del}(P)$.

Increasing the radius r from zero generates a nested sequence of alpha complexes. This ordered family of complexes,

$$\emptyset = \mathrm{Alp}(P, r_0) \subset \mathrm{Alp}(P, r_1) \subset \cdots \subset \mathrm{Alp}(P, r_k) \subset \cdots,$$

where $0 \leq r_0 < r_1 < \ldots$, is known as the *alpha filtration* of the point cloud P. Figure 5 illustrates this process.

A.2 PERSISTENT HOMOLOGY AND PERSISTENCE DIAGRAMS

Persistent homology tracks topological features, such as connected components, tunnels, and voids, across a sequence of nested spaces (Edelsbrunner & Harer, 2008; 2010). It records how long each feature persists over the filtration.

The formalism begins with the algebraic structure of a simplicial complex K. For a given dimension n, the set of all n-simplices in K forms a basis for the *n-chain group*, $C_n(K)$. An element of this group, an *n-chain*, is a formal sum $c = \sum a_i \sigma_i$, where each σ_i is an n-simplex and the coefficients a_i are taken from a field, typically $\mathbb{Z}_2$ in computational settings. The *boundary operator* $\partial_n : C_n(K) \to C_{n-1}(K)$ maps an n-simplex $\sigma = [x_0, x_1, \ldots, x_n]$ to the sum of its $(n-1)$-dimensional faces:

$$\partial_n \sigma = \sum_{j=0}^{n} [x_0, \ldots, \hat{x}_j, \ldots, x_n],$$

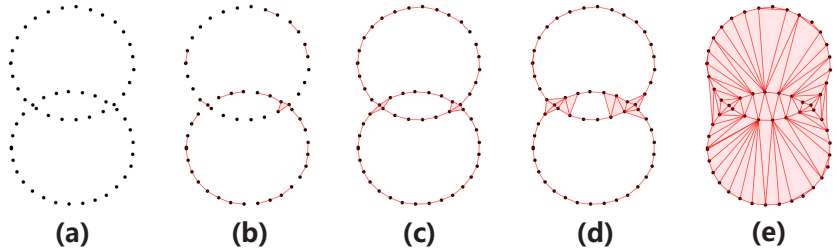

Figure 5: An alpha filtration. Additional simplices enter as the filtration parameter increases.

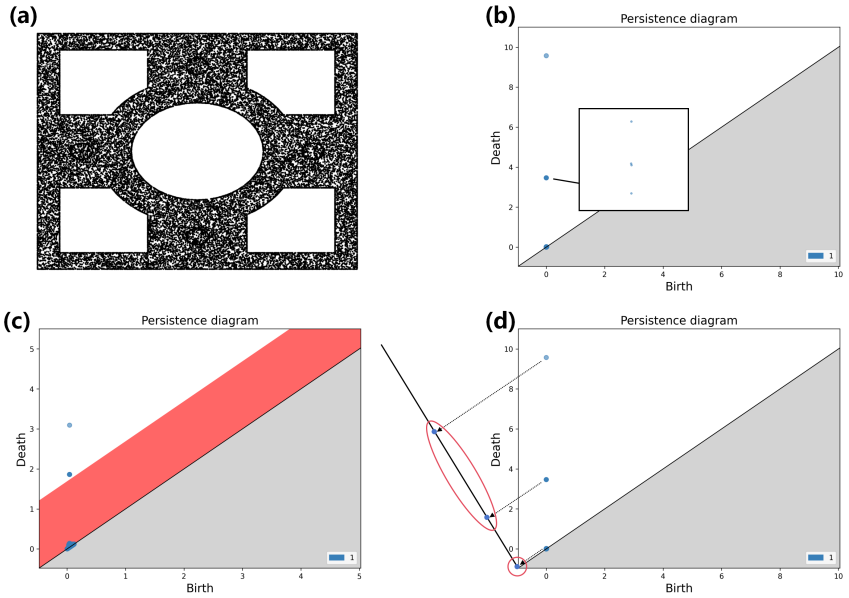

Figure 6: Illustration of persistence-based point labeling. (a) A planar point cloud with five loop features in the illustrative construction. (b) Its 1-PD. (c) The output of the confidence-set rule. (d) The output of the 2-means rule.

where $\hat{x}_j$ indicates that vertex x_j is omitted. This operator defines two important subgroups of $C_n(K)$: the group of *n-cycles*, $Z_n(K) = \mathrm{Ker}(\partial_n)$, consisting of chains with no boundary, and the group of *n-boundaries*, $B_n(K) = \mathrm{Im}(\partial_{n+1})$, consisting of chains that bound $(n+1)$-chains. The *n-th homology group* is the quotient group $H_n(K) = Z_n(K)/B_n(K)$, whose elements represent the n-dimensional holes of the complex.

For a filtration such as the alpha filtration described above,

$$K_0 \subset K_1 \subset \cdots \subset K_m,$$

the inclusion map $K_i \hookrightarrow K_j$ for $i \leq j$ induces a homomorphism on the corresponding homology groups, $f_n^{i,j} : H_n(K_i) \to H_n(K_j)$. The n-th **persistent homology group** from i to j is the image of this homomorphism, $H_n^{i,j} = \mathrm{Im}\, f_n^{i,j}$. Its rank, $\beta_n^{i,j} = \mathrm{rank}\, H_n^{i,j}$, is the n-th **persistent Betti number**. Persistent homology therefore tracks which homology classes (features) "persist" as the filtration progresses and records when they are born and die, that is, when they merge into an older class.

Persistent homology is commonly summarized by a *persistence diagram* (PD), a multiset of points in the extended plane $\mathbb{R}^2 \cup \{\infty\}$ (Edelsbrunner & Harer, 2010). For each dimension n, the n-th PD contains a point (b_i, d_i) for each n-dimensional topological feature that appears at filtration value b_i (its *birth time*) and disappears at value d_i (its *death time*). The vertical distance from a point to the diagonal, $d_i - b_i$, defines its *persistence*, which measures the feature's lifespan within the filtration.

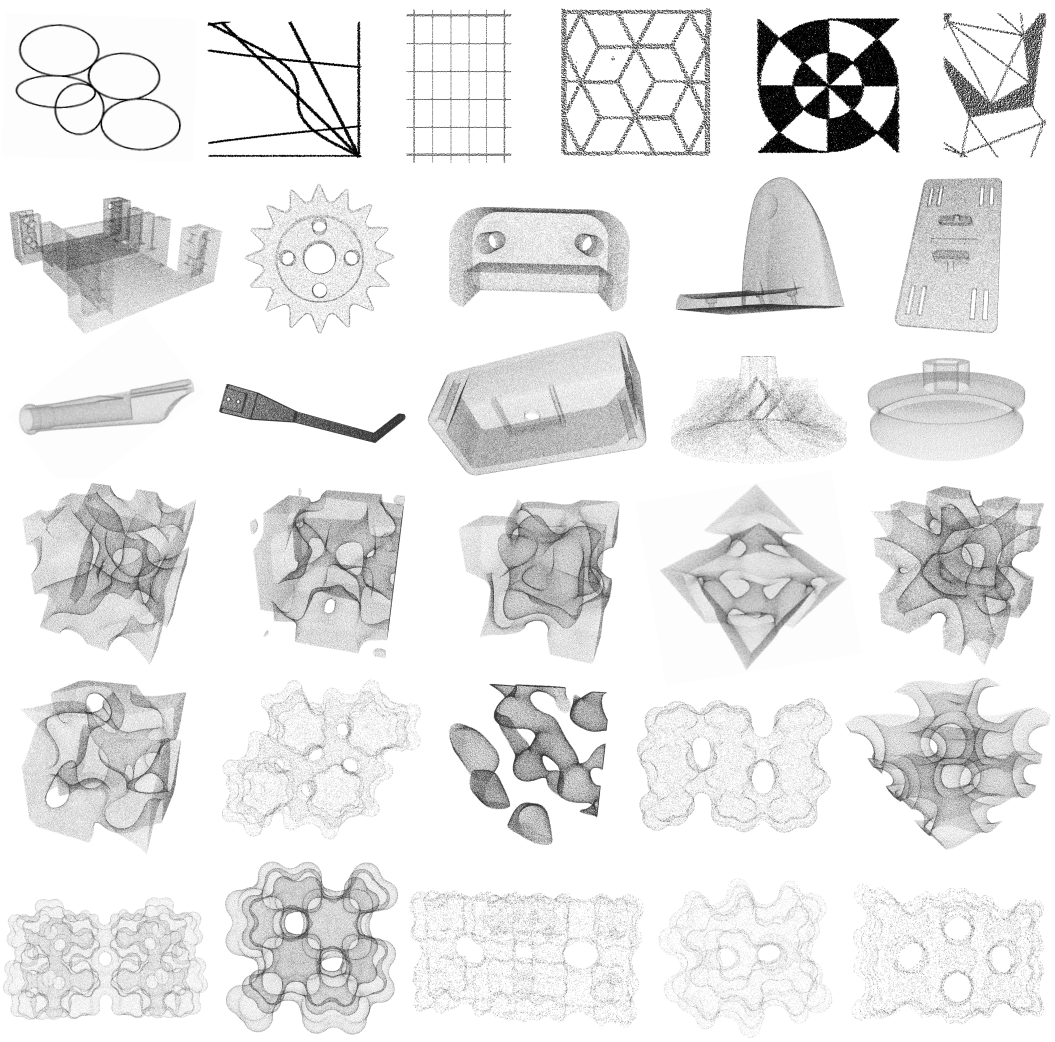

Figure 7: Examples of planar, CAD, TPMS, and zeolite point clouds used in the benchmark, grouped by category from top to bottom.

Because a homology class is an equivalence class of cycles, a PD entry does not by itself specify a unique geometric curve. A *representative cycle* selects one cycle from that class. Figure 1 uses volume-optimal representatives (Obayashi, 2018) to visualize selected entries, but these representatives are used for illustration rather than as labels for the complete benchmark.

A common heuristic treats points far from the diagonal, which have high persistence, as candidate *significant points* and points near the diagonal as candidate *noise points*. This heuristic provides a reference for interpreting persistence-based labels.

A.3 Persistence-Point Significance Rules

Figure 6 compares the confidence-set and 2-means rules. Panel (a) shows a planar point cloud whose illustrative construction contains five loops, and panel (b) shows its 1-PD, which contains five points with comparatively large persistence. In panel (c), the confidence-set rule places a statistically derived band around the diagonal at confidence level 0.5; entries inside the band are labeled as noise and entries outside it as significant (Fasy et al., 2014). In panel (d), 2-means partitions the retained (b_i, d_i) coordinates directly into two clusters. The cluster with larger mean persistence is labeled as significant.

B Implementation Details and Hyperparameters

Significant Points Labeling As described in Section 4.1, the benchmark uses the known geometric/topological count β_1 to initialize an operational target for each retained 1-PD. GUDHI stores alpha-filtration values as squared radii, so the exported coordinates are $\widetilde{p}_i = (\widetilde{b}_i, \widetilde{d}_i) = (b_i^2, d_i^2)$. We use the following procedure and omit tildes when discussing stored model inputs elsewhere. For each sample, we apply the following labeling procedure:

1. **Compute** β_1: For planar shapes, the geometric construction directly supplies the number of loops. For 3D models, it supplies the relevant topological metadata. For a connected orientable surface, we use $\beta_1 = 2g$ when the surface is closed and $\beta_1 = 2g + q - 1$ when it has $q > 0$ boundary components.
2. **Rank and assign**: We rank retained H1 entries by native squared-radius persistence $(\widetilde{d}_i - \widetilde{b}_i)$, select the top-$\beta_1$ entries as the initial positive set, and assign $y_i = 0$ to the remainder.
3. **Early-birth review**: One author inspects each exported label file with its source geometry. If needed, the review swaps an initially positive late-born entry with an earlier-born candidate while preserving exactly β_1 positive labels. Sampling scales differ across source geometries, so this decision is instance-specific. The corrected files are frozen before model training.

At inference, the model receives neither the geometric/topological count nor the manual correction history. It predicts the final label for each retained entry from the PD and available point-cloud context.

Input Numeric Handling. The stored PD input contains only the birth–death pair (b_i, d_i). Before feature expansion, we clip both coordinates to the implementation range $[0, 100]$. The encoder computes $\pi_i = d_i - b_i$ (clipped below at zero) and $\rho_i = \log(d_i / \max(b_i, 10^{-6}))$. We replace non-finite values with finite constants before the point-wise MLP. The valid-entry mask is constructed from the stored number of retained entries and is propagated through attention, pooling, loss computation, and evaluation. The point-cloud branch uses the stored number of valid points in the same way.

Persistence Diagram Encoder. We map the four-dimensional vector $v_i = [b_i, d_i, \pi_i, \rho_i]$ through $\mathrm{Linear}(4, H)$, LayerNorm, GELU, dropout, $\mathrm{Linear}(H, H)$, and a second LayerNorm. The resulting $N_{\mathrm{pd}} \times H$ token matrix then passes through PyTorch multi-head self-attention with eight heads. The key-padding mask prevents invalid entries from serving as keys or values. We follow the residual sum with LayerNorm and zero invalid rows again. We denote the point-wise contextual features by F_i. The global descriptor is the masked mean in Eq. (3), so its denominator is the number of valid entries rather than the padded length.

Point Cloud Encoder. The point-cloud branch consumes N_{pc} three-dimensional coordinates after padding or subsampling. It computes a validity-weighted centroid and root-mean-square radius using Eq. (5). The normalized coordinates pass through $\mathrm{Linear}(3, h)$, LayerNorm, GELU, $\mathrm{Linear}(h, H)$, LayerNorm, and GELU, where $h = \max(64, H/2)$. Masked mean and maximum pooling each produce an H-dimensional vector. We map their concatenation through $\mathrm{Linear}(2H, F)$, LayerNorm, GELU, and dropout to obtain $g_{\mathrm{pc}} \in \mathbb{R}^F$. A separate linear head followed by softplus predicts $\widehat{c}_{\mathrm{pc}}$ for the point-cloud count loss. This prediction is used only for supervision and is not concatenated to the per-point classifier input.

Conditional Point-Cloud Correction. The base classifier maps $[F_i; g_{\mathrm{pd}}] \in \mathbb{R}^{H+H}$ through $\mathrm{Linear}(2H, F)$, LayerNorm, GELU, dropout, and $\mathrm{Linear}(F, 2)$. We broadcast $\mathrm{LayerNorm}(g_{\mathrm{pc}})$ across the PD entries and concatenate it with F_i. The residual adapter maps this $(H+F)$-dimensional vector through $\mathrm{Linear}(H + F, F)$, LayerNorm, GELU, dropout, and $\mathrm{Linear}(F, 1)$. The gate head maps the same representation through $\mathrm{Linear}(H + F, F/2)$, GELU, and $\mathrm{Linear}(F/2, 1)$ followed by a sigmoid. The threshold head maps g_{pc} through $\mathrm{Linear}(F, F/2)$, GELU, and $\mathrm{Linear}(F/2, 1)$, producing one value per sample. These outputs are combined as in Eq. (8), changing only the positive logit. The added heads start with zero residual and threshold outputs and a small gate. TUN is warm-started from the same-seed PD-Encoder only checkpoint, with transferred parameters frozen for the first 10 epochs.

Significance Classifier. For each valid persistence point, we convert the corrected two-dimensional logit vector to class probabilities with softmax. The point-wise prediction is the class with the larger logit. The negative-class logit remains the PD-base value, while the positive-class logit receives the candidate-specific gated residual and the shared sample threshold. No β_1, manual review flag, or label-generation metadata is provided at inference.

Training Setup. We train with AdamW and cosine annealing and clip gradients to a global norm of 1.0. The batch size is 16 and the maximum number of epochs is 200. Each persistence diagram is retained at $N_{\text{pd}} = 100$ entries, and each point cloud is subsampled or padded to $N_{\text{pc}} = 4,096$ entries. Valid lengths exclude padding from the objective and metrics, and the point-cloud encoder applies per-cloud centering and RMS scaling. Experiments run on one NVIDIA GeForce RTX 4090 GPU (24GB VRAM), a 16-core Intel Xeon Platinum 8352V CPU at 2.10 GHz, and 120GB of system RAM.

The PD-Encoder only configuration uses a learning rate of 0.001. The final full TUN model uses a learning rate of 0.0005 for newly added context parameters and 0.0001 for transferred PD parameters, with weight decay 0.01 and a 10-epoch warm-start freeze. Training stops early when the validation loss fails to improve by 10^{-4} for 10 epochs. Each configuration is trained in ten independent runs, and the checkpoint with minimum validation loss is used for testing. One configuration is shared across the four categories; test-set metrics are not used for checkpoint or hyperparameter selection.

Configuration. Unless otherwise specified, we use $H = 256$, fusion dimension $F = 256$, intermediate point-cloud width $h = 128$, eight attention heads, and dropout 0.2. Data caps are $N_{\text{pd}} = 100$ and $N_{\text{pc}} = 4,096$ with batch size 16. The weighted focal loss uses $(\alpha, \gamma) = (1.0, 2.0)$ and class weights $w_0 = 1.0$, $w_1 = 2.0$. The full TUN objective uses expected-count loss weight 0.1, point-cloud count weight 0.05, and residual-L_2 weight 10^{-4}. In addition to the same point-wise focal term, the matched PD-Encoder only objective adds only the expected-count term and has no point-cloud branch.

Comparison Models. The two deterministic references are evaluated once on each test category. They operate in radius units: the native squared-radius coordinates $(\widetilde{b}_i, \widetilde{d}_i)$ are transformed to $(b_i, d_i) = (\sqrt{\widetilde{b}_i}, \sqrt{\widetilde{d}_i})$ while preserving entry order. For 2-means, we apply k-means with $k = 2$ and ten initializations; the cluster with larger mean radius persistence is positive. For CS(0.5), each point cloud of size n is resampled 500 times with replacement using subsamples of size $\lfloor n/\log n \rfloor$. In each repetition we measure the maximum nearest-neighbor distance from the held-out points to the subsample. If $q_{0.5}$ is the median of these distances, an entry is positive when $d_i - b_i > 4q_{0.5}$. The resampling seed starts at zero and is incremented by sample index.

All supervised comparison methods predict one binary label for each retained PD entry. For each of the ten initializations, the 3,500 development records are split by sample into 3,000 training and 500 validation records. SVM (Hearst et al., 1998) uses scikit-learn's RBF classifier with $C = 1$, $\gamma = \texttt{scale}$, and balanced class weights on the clipped but otherwise unstandardized coordinates. The two-layer MLP (Murtagh, 1991) uses 64 hidden units per layer, ReLU, BatchNorm, dropout 0.3, AdamW with learning rate 0.001, batch size 16, and inverse-frequency weighted cross-entropy. Its coordinates are standardized using statistics fitted only on that seed's training split, and training stops by validation loss with patience 20 or after 200 epochs. Both SVM and MLP receive only $(b_i, d_i) \in \mathbb{R}^2$.

DeepSets (Zaheer et al., 2017) and Set Transformer (Lee et al., 2019) are PD-only set baselines adapted with point-wise heads. Both use width 256, dropout 0.2, the same four derived PD descriptors as TUN, and an expected-count loss weighted by 0.1. DeepSets combines shared token layers with masked-mean aggregation. Set Transformer uses two eight-head self-attention blocks and multi-head-attention pooling. Both preserve point-level permutation equivariance while using permutation-invariant global context. These two set baselines use the same valid-entry masks, ten-run protocol, validation-loss checkpoint rule, and point-wise metrics as TUN; the point-wise MLP instead operates directly on the unpadded retained entries.

C Dataset and Label Audit

We audit the released labels against the count-and-persistence initialization described in Section 4.1. For each diagram, we reconstruct the initial positive set by taking the top k entries under native squared-radius persistence, where k is the positive count in the final label file. A diagram is marked as changed when this reconstructed set differs from the final positive set. The number of swaps is the size of the final positive set outside the reconstructed set; each swap therefore replaces one initially selected entry while preserving the positive cardinality.

Table 3: Dataset composition and label audit. Positive rate is computed over valid retained PD entries. Changed diagrams differ from the reconstructed top-β_1 persistence initialization; swaps count positive-set replacements. The development pool is split into 3,000 training and 500 validation records independently for each seed.

Split	**Records**	**Valid entries**	**Positive rate**	**Changed diagrams**	**Swaps**
Development pool	3,500	347,605	9.06%	851 (24.3%)	3,555
Planar	250	25,000	5.54%	50 (20.0%)	68
CAD	500	50,000	3.64%	28 (5.6%)	30
TPMS	500	50,000	10.77%	241 (48.2%)	1,083
Zeolite	240	24,000	19.63%	99 (41.3%)	548
Total	4,990	496,605	9.02%	1,269 (25.4%)	5,284

The audit makes two properties of the benchmark explicit. First, the positive class is sparse and its prevalence varies substantially across geometric categories. Second, the review changes the identity, but not the number, of positive entries in 1,269 of 4,990 diagrams. The largest changes occur for TPMS and zeolite structures, whose repeated multi-scale tunnels create more opportunities for an early-born entry to replace a higher-persistence late-born entry. Thus, the final operational target is not equivalent to selecting the top-β_1 entries by persistence alone.

D Illustrations of Some Point Clouds

Figure 7 shows representative planar shapes, CAD models, TPMS models, and zeolite structures from the benchmark.

E Additional Experimental Results

E.1 Point-Level Confusion Counts

Tables 4 and 5 report the error composition underlying Table 1. Among TUN and the three set-structured neural models, TUN yields the fewest false positives on all four categories, reducing their aggregate per-run count from 113.7 for the matched PD-Encoder only model to 25.0. DeepSets and Set Transformer attain slightly higher recall on CAD, whereas TUN reduces false positives from 15.5 for DeepSets to 8.1; this precision–recall trade-off accounts for the 0.0010 CAD F1 difference. On the other three categories, the same false-positive suppression accompanies the highest F1-score.

Table 4: Point-level confusion counts for Planar and CAD. Deterministic rules are evaluated once; values for learning methods are averaged over ten runs. Bold denotes the best value in each metric column: higher is better for TP/TN and lower is better for FP/FN.

	Planar				CAD			
Method	**TP**	**TN**	**FP**	**FN**	**TP**	**TN**	**FP**	**FN**
2-means	1182	18483	5131	204	1388	34323	13856	433
CS (0.5)	201	23609	5	1185	123	48176	**3**	1698
SVM	1254.4	23555.5	58.5	131.6	1634.1	47878.0	301.0	186.9
MLP	492.5	23609.2	**4.8**	893.5	306.8	48066.6	112.4	1514.2
DeepSets	**1386.0**	23578.5	35.5	**0.0**	1816.3	48163.5	15.5	4.7
Set Transformer	1385.7	23571.8	42.2	0.3	**1820.4**	48149.4	29.6	**0.6**
PD-Encoder only	1378.6	23587.4	26.6	7.4	1815.9	48140.6	38.4	5.1
TUN	1381.8	**23601.8**	12.2	4.2	1805.5	**48170.9**	8.1	15.5

Table 5: Point-level confusion counts for TPMS and Zeolite. Deterministic rules are evaluated once; values for learning methods are averaged over ten runs. TPMS corresponds to the porous-structure split in the result files. Bold denotes the best value in each metric column: higher is better for TP/TN and lower is better for FP/FN.

	TPMS				Zeolite			
Method	**TP**	**TN**	**FP**	**FN**	**TP**	**TN**	**FP**	**FN**
2-means	3449	32052	12561	1938	4564	11524	7766	146
CS (0.5)	0	44613	**0**	5387	0	19290	**0**	4710
SVM	5195.6	44431.2	181.8	191.4	4702.2	18111.6	1178.4	7.8
MLP	1107.0	44535.5	77.5	4280.0	194.5	19289.7	0.3	4515.5
DeepSets	5378.8	44541.2	71.8	8.2	4707.4	19280.7	9.3	2.6
Set Transformer	**5383.6**	44563.3	49.7	**3.4**	**4709.9**	19277.4	12.6	**0.1**
PD-Encoder only	5379.8	44581.9	31.1	7.2	4709.7	19272.4	17.6	0.3
TUN	5375.6	**44609.6**	3.4	11.4	4707.4	**19288.7**	1.3	2.6

E.2 CAD Retrieval Protocol Under Synthetic Corruptions

Dataset and fixed split. The retrieval collection is selected from Thingi10K by geometry criteria alone. Auditing merges coincident vertices and requires finite coordinates, a non-degenerate triangulation, manifold edges, watertightness, and one connected component. All 500 retained meshes pass these checks and have mesh-derived first Betti numbers below 100. A fixed source-identity manifest assigns 100 meshes to protocol validation and 400 to final evaluation. The full-gallery experiment uses the 400 final-evaluation identities and therefore measures retrieval among loop-bearing CAD models.

Sampling, corruptions, and persistence diagrams. For a clean gallery item, 8,192 surface points are generated by area-weighted oversampling followed by farthest-point sampling. Each query is independently resampled from its source mesh and receives exactly one of the following transformations: *density* at 50% or 25% of the nominal points; *jitter* with Gaussian coordinate noise whose standard deviation is 0.005 or 0.01 times the source bounding-box diagonal; *non-uniform* sampling with triangle probabilities proportional to squared triangle area; *outliers* at 1% or 5% of the sampled points, drawn from a box padded by 5% of the diagonal; or *partial* observation with 20% or 30% of the points removed by a random-direction half-space quantile. STL coordinate units are preserved before persistence computation.

For every clean or corrupted observation, GUDHI constructs the alpha filtration over its observation-specific dense point set using field $\mathbb{Z}_2$. Filtration values are squared radii. After removing infinite intervals, at most 100 H_1 entries are retained by persistence, matching the model input contract.

The TUN point-cloud input is a deterministic subsample of at most 4,096 points from the same independently sampled observation.

Selectors and common retrieval distance. For each of the ten frozen CAD checkpoints, TUN selects entries whose point-wise predicted class is positive. For the same diagram, Top-persistence@k_{TUN} retains the k_{TUN} longest intervals, where k_{TUN} is TUN's predicted positive count. This matched-budget diagnostic fixes cardinality to isolate subset identity. Random-matched draws five uniform subsets of the same size and averages their results. Count-only discards PD coordinates and compares two observations by $|k_q - k_g|$.

We use a finite-diagonal augmented assignment distance for all diagram-valued selectors. An off-diagonal match between $x, y \in \mathbb{R}^2$ incurs $\|x - y\|_2^2$, an unmatched interval (b, d) incurs $((d-b)/2)^2$, and ranking uses the square root of the minimum total assignment cost. The distance remains finite when one selected diagram is empty. The primary full-gallery protocol compares against all 400 clean final-evaluation models. The secondary protocol restricts the gallery to models with the same mesh-derived β_1 as the query, so its gallery size varies across queries. Both protocols use direct distance ranking without training a retrieval model.

Metrics and aggregation. Let r_q be the rank of the unique clean gallery model with the same source identity as query q. For Q queries,

$$\mathrm{R@}K = \frac{1}{Q}\sum_{q=1}^{Q} \mathbb{I}[r_q \le K], \qquad \mathrm{mAP} = \frac{1}{Q}\sum_{q=1}^{Q} \frac{1}{r_q}. \tag{10}$$

In this single-relevant-item task, mAP equals mean reciprocal rank. Each gallery protocol contributes $400 \times 9 = 3{,}600$ query cases per checkpoint. We first average the nine equally sized corruption conditions within each checkpoint and protocol, then report the mean and sample standard deviation across ten checkpoints. Random-matched is first averaged across its five subset draws. The per-corruption table below reports the same statistics for each condition in the full-gallery protocol.

In the full-gallery protocol, the outlier conditions produce the clearest separation: TUN is higher than top-persistence on all three metrics for both outlier ratios. Density and jitter results are close, whereas top-persistence is higher under non-uniform sampling and partial observations. The same-β_1 protocol preserves this ordering pattern while controlling the mesh-derived loop count. Because its gallery size varies, we use the full-gallery results as the primary retrieval comparison. The selection controls place both diagram-valued selectors well above Count-only and Random-matched, indicating that the geometry of the retained entries carries the retrieval signal.

Table 6: Full-gallery CAD retrieval results for each query corruption. Values are mean ± sample standard deviation over ten frozen TUN checkpoints; Random-matched is first averaged over five subset draws. Bold denotes the best value within each corruption condition and metric.

Corruption	Selection	R@1	R@5	mAP
Density (25%)	Count-only	0.0115 ± 0.0036	0.0530 ± 0.0075	0.0449 ± 0.0052
	Random-matched	0.0181 ± 0.0023	0.0402 ± 0.0042	0.0368 ± 0.0028
	Top-persistence@k_{TUN}	$\mathbf{0.4008 \pm 0.0503}$	$\mathbf{0.5677 \pm 0.0669}$	$\mathbf{0.4773 \pm 0.0574}$
	TUN	0.3990 ± 0.0473	0.5655 ± 0.0629	0.4753 ± 0.0537
Density (50%)	Count-only	0.0180 ± 0.0045	0.0853 ± 0.0154	0.0666 ± 0.0075
	Random-matched	0.0273 ± 0.0042	0.0616 ± 0.0108	0.0558 ± 0.0060
	Top-persistence@k_{TUN}	$\mathbf{0.5723 \pm 0.0413}$	$\mathbf{0.7080 \pm 0.0464}$	$\mathbf{0.6365 \pm 0.0429}$
	TUN	0.5685 ± 0.0417	0.7032 ± 0.0464	0.6333 ± 0.0430
Jitter (0.5% bbox)	Count-only	0.0158 ± 0.0035	0.0698 ± 0.0070	0.0565 ± 0.0042
	Random-matched	0.0252 ± 0.0039	0.1047 ± 0.0059	0.0782 ± 0.0052
	Top-persistence@k_{TUN}	0.0638 ± 0.0056	$\mathbf{0.2533 \pm 0.0089}$	0.1609 ± 0.0052
	TUN	$\mathbf{0.0683 \pm 0.0059}$	0.2513 ± 0.0079	$\mathbf{0.1632 \pm 0.0059}$
Jitter (1% bbox)	Count-only	0.0087 ± 0.0029	0.0457 ± 0.0079	0.0390 ± 0.0043
	Random-matched	0.0054 ± 0.0010	0.0328 ± 0.0021	0.0308 ± 0.0008
	Top-persistence@k_{TUN}	$\mathbf{0.0025 \pm 0.0000}$	$\mathbf{0.0273 \pm 0.0014}$	$\mathbf{0.0361 \pm 0.0005}$
	TUN	$\mathbf{0.0025 \pm 0.0000}$	$\mathbf{0.0273 \pm 0.0014}$	0.0358 ± 0.0005
Non-uniform ($w \propto A^2$)	Count-only	0.0145 ± 0.0044	0.0678 ± 0.0063	0.0551 ± 0.0033
	Random-matched	0.0452 ± 0.0050	0.1076 ± 0.0073	0.0857 ± 0.0063
	Top-persistence@k_{TUN}	0.4348 ± 0.0187	$\mathbf{0.5822 \pm 0.0279}$	$\mathbf{0.5074 \pm 0.0223}$
	TUN	$\mathbf{0.4353 \pm 0.0225}$	0.5657 ± 0.0263	0.4992 ± 0.0247
Outliers (1%)	Count-only	0.0457 ± 0.0086	0.1710 ± 0.0188	0.1232 ± 0.0128
	Random-matched	0.0873 ± 0.0066	0.2274 ± 0.0161	0.1629 ± 0.0111
	Top-persistence@k_{TUN}	0.0795 ± 0.0074	0.1585 ± 0.0081	0.1251 ± 0.0076
	TUN	$\mathbf{0.3058 \pm 0.0392}$	$\mathbf{0.4120 \pm 0.0408}$	$\mathbf{0.3583 \pm 0.0383}$
Outliers (5%)	Count-only	0.0265 ± 0.0046	0.1095 ± 0.0118	0.0847 ± 0.0053
	Random-matched	0.0073 ± 0.0018	0.0308 ± 0.0024	0.0285 ± 0.0021
	Top-persistence@k_{TUN}	0.0107 ± 0.0012	0.0325 ± 0.0035	0.0311 ± 0.0013
	TUN	$\mathbf{0.0863 \pm 0.0128}$	$\mathbf{0.1585 \pm 0.0203}$	$\mathbf{0.1269 \pm 0.0155}$
Partial (20% removed)	Count-only	0.0105 ± 0.0033	0.0452 ± 0.0070	0.0499 ± 0.0032
	Random-matched	0.0952 ± 0.0059	0.2542 ± 0.0133	0.1779 ± 0.0085
	Top-persistence@k_{TUN}	$\mathbf{0.4490 \pm 0.0245}$	$\mathbf{0.5760 \pm 0.0272}$	$\mathbf{0.5121 \pm 0.0253}$
	TUN	0.4360 ± 0.0207	0.5633 ± 0.0254	0.4996 ± 0.0226
Partial (30% removed)	Count-only	0.0107 ± 0.0026	0.0393 ± 0.0055	0.0417 ± 0.0038
	Random-matched	0.0707 ± 0.0045	0.2008 ± 0.0105	0.1403 ± 0.0064
	Top-persistence@k_{TUN}	$\mathbf{0.3430 \pm 0.0226}$	$\mathbf{0.4695 \pm 0.0268}$	$\mathbf{0.4067 \pm 0.0243}$
	TUN	0.3395 ± 0.0214	0.4383 ± 0.0203	0.3940 ± 0.0217

Table 7: Aggregate CAD retrieval results under the two gallery protocols. Values are mean ± sample standard deviation over ten frozen TUN checkpoints; Random-matched is first averaged over five subset draws. Bold denotes the best value within each gallery protocol and metric.

Gallery protocol	Selection	R@1	R@5	mAP
Full gallery	Count-only	0.0180 ± 0.0019	0.0763 ± 0.0056	0.0624 ± 0.0027
	Random-matched	0.0424 ± 0.0024	0.1178 ± 0.0062	0.0885 ± 0.0040
	Top-persistence@k_{TUN}	0.2618 ± 0.0173	0.3750 ± 0.0226	0.3215 ± 0.0197
	TUN	$\mathbf{0.2934 \pm 0.0222}$	$\mathbf{0.4094 \pm 0.0267}$	$\mathbf{0.3540 \pm 0.0240}$
Same-β_1 gallery	Count-only	0.1384 ± 0.0053	0.3882 ± 0.0059	0.2658 ± 0.0045
	Random-matched	0.1637 ± 0.0067	0.3861 ± 0.0077	0.2796 ± 0.0069
	Top-persistence@k_{TUN}	0.4023 ± 0.0193	0.6163 ± 0.0216	0.5058 ± 0.0190
	TUN	$\mathbf{0.4345 \pm 0.0225}$	$\mathbf{0.6438 \pm 0.0241}$	$\mathbf{0.5349 \pm 0.0219}$