HEAT FIELD SIGNATURES: FROM POINT CLOUDS TO SMOOTH GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Bringing multiscale geometric analysis directly to irregular point clouds remains difficult: quantities such as local dimension, anisotropy, density variation, and geometric transitions are typically estimated through explicit neighborhood, manifold, or graph constructions, or left for neural networks to infer from coordinates. We introduce *Heat Field Signatures* (HFS), which lift a point cloud to a multiscale family of smooth ambient heat fields, providing a direct interface from discrete samples to geometric analysis.

From this field, HFS computes closed-form global and local signatures directly from pairwise distances, capturing heat concentration, intrinsic dimension, anisotropy, and scale transitions. We further introduce the *Heat Dimension Spectrum* (HDS), a compact summary of multiscale geometric composition. HFS can be used as a closed-form descriptor, a lightweight learned representation, or a geometric feature channel for neural point-cloud models.

Across synthetic and real-world benchmarks spanning subcellular, neuronal, tree, and protein data, HFS outperforms strong point-cloud baselines, while standalone variants offer lower end-to-end cost. On SCOP protein-fold classification, HFS improves over the strongest standard deep baseline tested by 24.8 percentage points using coordinates alone, while standalone HFS representations are exactly rotation-invariant by construction. More broadly, HFS turns a classical heat field into a practical interface for multiscale geometric analysis in modern point-cloud learning.

1 INTRODUCTION

Modern learning has repeatedly advanced when a branch of mathematics acquires a practical interface to finite data. Kernels expose functional geometry through inner products, persistent homology turns topology into stable summaries, and graph Laplacians make diffusion and spectral structure computable from samples (Schölkopf & Smola, 2002; Edelsbrunner et al., 2002; Coifman & Lafon, 2006).

For irregular point clouds, however, there is no comparably standard interface to the rich toolbox of *differential geometry*. Quantities such as local dimension, anisotropy, curvature, density variation, and geometric transitions are often precisely what distinguishes classes, yet point-cloud networks typically learn such structure only implicitly from coordinates and neighborhoods (Qi et al., 2017a; Wang et al., 2019; Zhao et al., 2021).

We propose a simple interface: *lift the point cloud to a smooth, multiscale geometric object*. Formally, *Heat Field Signatures* (HFS) maps $X = \{x_1, \dots, x_m\} \subset \mathbb{R}^n$ to the multiscale ambient heat field

$$u_X(y,t) = (4\pi t)^{-n/2} \sum_{i=1}^{m} \exp\left(-\frac{\|y - x_i\|^2}{4t}\right),$$

whose graph is a smooth hypersurface in $\mathbb{R}^{n+1}$ for every diffusion scale t. A discrete, irregular sample is thereby replaced by a family of smooth geometric objects on which derivatives, Hessians, energies, and other geometric quantities are directly available. The point is not merely to smooth the data: the heat field provides an *interface through which geometric analysis can be brought systematically into point-cloud learning*.

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.

This interface is particularly natural for data whose identity is not encoded by a rigid global silhouette. Neurons, protein structures, spatial processes, and LiDAR trees are characterized by how geometry changes across space and scale: filaments meet at junctions, dense regions transition to sparse ones, and structures of different local dimension coexist. Diffusion time provides the scale axis needed to resolve this organization, while the smooth heat landscape makes it analytically accessible. Unlike spectral heat methods that diffuse *on* a graph or manifold, HFS analyzes the *ambient* field generated directly by the samples, requiring no mesh, graph Laplacian, normal estimation, or Laplacian eigendecomposition.

In this work, we instantiate this broader interface with a small set of complementary geometric measurements. Global heat signatures capture concentration, scale response, and roughness. Locally, the *heat dimension* estimates effective intrinsic dimension, the log-Hessian spectrum captures tangent–normal structure and anisotropy, and scale-transition responses identify junctions, boundaries, and mixed-dimensional regions. We further introduce the *Heat Dimension Spectrum* (HDS), which summarizes the multiscale distribution of point-, filament-, surface-, and volume-like regions and their interaction with sampling density. These are not intended to exhaust the geometry available from the heat field, but to demonstrate the range of information exposed by the interface.

The resulting representation can be used as a closed-form descriptor, equipped with a lightweight learned pooling head, or supplied as geometric features to a point-cloud network; closed-form HFS descriptors are already highly competitive, while learned pooling and backbone integration provide additional flexibility. Across synthetic density/topology benchmarks and real subcellular, neuronal, tree, and protein data, an HFS variant achieves the best performance on all seven benchmarks; the standalone variants provide substantially lower computational cost and exact rotation invariance by construction. On SCOP protein-fold classification, HFS improves over the strongest standard deep point-cloud baseline in Table 1 by 24.8 percentage points using coordinates alone. These results suggest that explicitly exposing geometric structure can be especially effective when intrinsic multiscale geometry itself carries the label.

Our contributions are:

- We introduce **HFS as a geometric interface for point clouds**: a discrete cloud is lifted to a multiscale family of smooth heat landscapes, making tools from differential geometry directly applicable to finite, irregular data.
- We instantiate this interface with **global and local heat-geometric signatures** capturing concentration, intrinsic dimension, anisotropy, and scale transitions, together with the **Heat Dimension Spectrum (HDS)** for shape-level aggregation.
- We develop both **closed-form and learned HFS representations**, retaining geometric invariance in the standalone variants while allowing task-specific learning and integration with standard point-cloud networks.
- Across synthetic and real-world benchmarks, HFS consistently outperforms strong point-cloud baselines, with the standalone variants providing exact rotation invariance and substantially lower computational cost.

2 Related Work

Point-cloud learning. PointNet (Qi et al., 2017a) applied shared MLPs and global max-pooling to individual points; PointNet++ (Qi et al., 2017b) added hierarchical local grouping via farthest-point sampling. DGCNN (Wang et al., 2019) built dynamic k-NN graphs with edge convolutions, and attention-based architectures such as Point Transformer (Zhao et al., 2021) and PCT (Guo et al., 2021) apply self-attention to local neighborhoods; masked autoencoders (Pang et al., 2022) further improve representations through self-supervised pretraining. Density-adaptive variants such as DRINet (Ye et al., 2021) and domain-adaptation methods such as PointDAN (Qin et al., 2019) address nonuniform sampling by reweighting neighborhood aggregation, and self-supervised approaches (Huang et al., 2021) learn density invariance through augmentation – but all of these methods learn geometric structure implicitly from coordinates and learned features. In contrast, our approach makes local geometry explicit: rather than learning density robustness from augmented data, logarithmic scale derivatives of the heat field are approximately invariant to multiplicative density variation by construction (Section 3), complementing learned representations with theoretically grounded geometric variables.

Handcrafted descriptors and dimension estimation. Before deep learning, point-cloud recognition relied on handcrafted local descriptors: FPFH (Rusu et al., 2009) and PFH (Rusu et al., 2008) encode surface geometry through histograms of angular relationships between point normals, while SHOT (Tombari et al., 2010) and spin images (Johnson & Hebert, 1999) combine shape and keypoint-centered histograms. These operate at a single fixed scale and assume well-oriented normals, limiting robustness to noisy or heterogeneous sampling. Separately, intrinsic-dimension estimators – correlation dimension (Grassberger & Procaccia, 1983), local PCA (Camastra & Staiano, 2016), and TWO-NN (Facco et al., 2017) – estimate dimension from local geometry or pairwise distances. TWO-NN also examines dimension across scales through subsampling. Our contribution is not scale-dependent dimension alone: heat dimension $d_{\mathrm{heat}}(x,t)$ is one component of a joint geometric-type descriptor, together with codimension and scale-transition signals, requiring no normal estimation. We aggregate its distribution into the Heat Dimension Spectrum, a shape-level summary of dimensional composition across scale.

Heat kernels and spectral methods. The Heat Kernel Signature (HKS) (Sun et al., 2009) uses the diagonal of the heat kernel on a mesh as a multiscale point descriptor, exploiting the connection between heat diffusion and Laplace-Beltrami eigenfunctions; the Wave Kernel Signature (WKS) (Aubry et al., 2011) replaces diffusion with wave propagation for improved frequency selectivity. Both require a mesh or graph Laplacian; HKS also allows boundaries with specified boundary conditions. Diffusion maps (Coifman & Lafon, 2006) use the heat kernel for geometry-preserving embeddings, but target global dimensionality reduction rather than local geometric type. We differ in three ways: we work directly on unstructured point clouds without mesh or graph construction; we use the differential geometry of the ambient heat field rather than a Laplacian eigendecomposition; and we extract local quantities – dimension, codimension, and scale transitions – rather than global embeddings or signatures.

Topological and geometric data analysis. Persistent homology (Edelsbrunner et al., 2002) encodes multiscale topological features of point clouds as persistence diagrams; extensions such as the multicover bifiltration (Edelsbrunner & Osang, 2021) incorporate density information into topological descriptors. The resulting density–distance bifiltration has also motivated synthetic benchmarks for evaluating joint sensitivity to scale and density (Carrière & Blumberg, 2020; Kerber & Russold, 2024), which we adopt as evaluation settings for our method. These methods capture connectivity and homological structure but do not directly measure local dimension, codimension, or anisotropy. Our approach is complementary: topological methods capture global connectivity across scale, while heat-geometric methods capture local geometric type at each point; the two frameworks could in principle be combined, which we leave for future work.

3 Heat Field Signatures

3.1 Overview

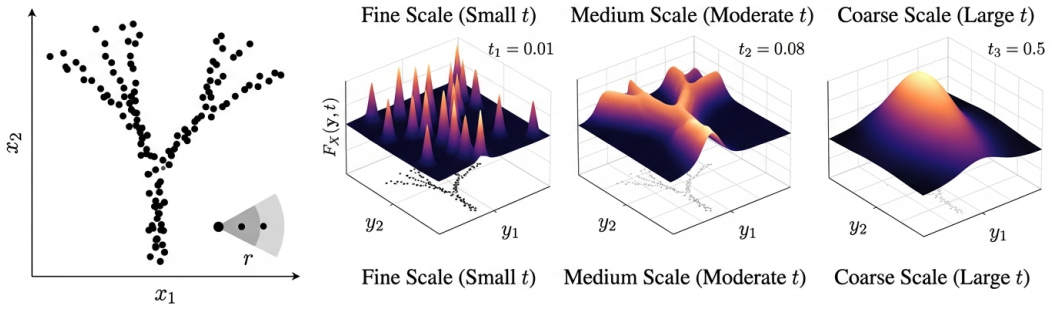
Given a point cloud $X = \{x_1, \ldots, x_m\} \subset \mathbb{R}^n$, we seek a representation that captures not only *where* the points lie, but how their geometric organization changes across scale. HFS first converts the discrete cloud into a smooth multiscale heat field and then reads this field at three complementary levels:

$$X \longrightarrow u_X(y,t) \longrightarrow [\Phi_{\mathrm{global}}(X),\ \Phi_{\mathrm{HDS}}(X),\ z_X] \longrightarrow \text{classifier}.$$

Global signatures describe how the entire heat landscape concentrates and dissipates; local signatures identify the geometric type surrounding each point; and the Heat Dimension Spectrum (HDS) summarizes how these local types are distributed across the shape. Together they capture both dominant morphology and localized structures such as thin branches, junctions, and mixed-dimensional interfaces. Figure 2 summarizes the pipeline; derivations and geometric interpretation are given in Appendix A. For general background on geometric heat-kernel methods, see Grigoryan (2009).

3.2 Heat Field and Global Signatures

We associate to X the ambient heat field $u_X(y,t) = (4\pi t)^{-n/2} \sum_{j=1}^{m} \exp\left(-\frac{\|y-x_j\|^2}{4t}\right), \quad t > 0,$ where $\sqrt{t}$ acts as a geometric length scale. Small t resolves fine local structure, while increasing t progressively exposes coarser organization. Define $D_{ij} = \|x_i - x_j\|, \quad w_{ij}(t) = \exp\left(-\frac{D_{ij}^2}{4t}\right).$



(A) Input: Point Cloud X in $\mathbb{R}^2$ (B) Evolving Heat Landscape $F_X(y,t)$ across Scale t

Figure 1: **Multiscale heat landscape.** For visualization, we show the unnormalized Gaussian field $F_X(y,t) = \sum_i e^{-\|y-x_i\|^2/4t}$. Fine scales resolve local neighborhoods, intermediate scales reveal branches and junctions, and coarse scales retain global structure. HFS uses the normalized field u_X, which has the same spatial Gaussian structure at each scale.

Because u_X is a Gaussian mixture, useful global functionals are available directly from pairwise distances:

$$E_2(t) = (8\pi t)^{-n/2} \sum_{i,j} e^{-D_{ij}^2/8t}, \quad C_2(t) = -t\,\partial_t \log E_2(t), \quad \mathcal{E}(t) = -\tfrac{1}{2}\partial_t E_2(t), \quad \widetilde{\mathcal{E}}(t) = \frac{\mathcal{E}(t)}{E_2(t)+\varepsilon}.$$

These measure, respectively, heat concentration, scale response, spatial roughness, and normalized roughness. Concatenating them across diffusion scales gives the global descriptor $\Phi_{\text{global}}(X)$. Their closed forms and geometric interpretation are detailed in Appendix A.

3.3 Local Heat-Geometric Signatures

Global quantities describe the heat landscape as a whole. To expose localized morphology, we instead examine logarithmic derivatives of u_X at each input point.

Heat dimension. We define $d_{\text{heat}}(x_i,t) = n + 2t\,\partial_t \log u_X(x_i,t) = \dfrac{\sum_j D_{ij}^2 w_{ij}(t)}{2t\sum_j w_{ij}(t)}$.

If the neighborhood of x behaves locally like a d-dimensional structure, then

$$u_X(x,t) \approx C(x)t^{-(n-d)/2} \qquad \Longrightarrow \qquad d_{\text{heat}}(x,t) \approx d.$$

Thus heat dimension provides a smooth, scale-dependent notion of local intrinsic dimension: in $\mathbb{R}^3$, values near 1, 2, and 3 correspond ideally to filament-, surface-, and volume-like regions. **Theorem** 1 in Appendix A.3 gives a finite-sample approximation guarantee and identifies the intermediate regime between sampling resolution and geometric variation in which this interpretation is valid.

Directional geometry and scale transitions. Heat dimension counts effective directions but does not distinguish them. We therefore use the log-Hessian $Q_X(x_i,t) = \nabla_y^2 \log u_X(y,t)\big|_{y=x_i}$, with eigenvalues $\lambda_1(x_i,t) \geq \cdots \geq \lambda_n(x_i,t)$. For an ideal locally flat d-dimensional structure,

$$2t\lambda_k \approx \begin{cases} 0, & \text{tangent directions,} \\ -1, & \text{normal directions,} \end{cases}$$

so the normalized spectrum captures tangent–normal structure, codimension, and anisotropy.

Finally, $\tau(x_i,t) = \partial_{\log t} d_{\text{heat}}(x_i,t) = t\,\partial_t d_{\text{heat}}(x_i,t)$ measures how rapidly the visible local geometry changes with scale. Large $|\tau|$ highlights boundaries, junctions, protrusions, and mixed-dimensional transitions.

Together, $\psi_X(x_i,t) = [d_{\text{heat}}(x_i,t),\ \tau(x_i,t),\ 2t\lambda_1(x_i,t),\ldots,2t\lambda_n(x_i,t)]$ provides a compact local geometric type. In $\mathbb{R}^3$, ψ_X has five components per scale. Closed-form derivatives and a geometric dictionary for these signatures are given in Appendix A.3.

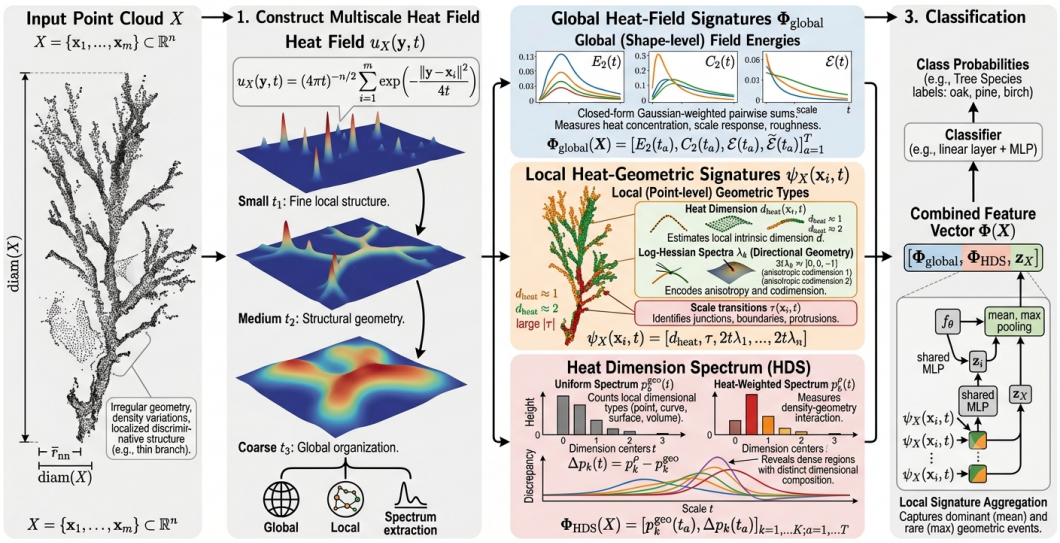


Figure 2: **Heat Field Signatures (HFS).** A point cloud is converted into a multiscale ambient heat field. HFS extracts complementary global heat responses, local geometric signatures, and the Heat Dimension Spectrum (HDS), then aggregates them into a fixed-dimensional representation for classification.

3.4 Heat Dimension Spectrum

A point cloud rarely has a single geometric type: a neuron may contain filamentary branches and a compact soma, while a spatial process may mix dense clusters with sparse regions. We therefore summarize the distribution of local heat dimensions with the *Heat Dimension Spectrum* (HDS).

Let $c_1, \ldots, c_K$ denote dimension centers; in $\mathbb{R}^3$ we use $c_k \in \{0, 1, 2, 3\}$. With $\sigma = 0.5$, define

$$\eta_k(d) = \frac{e^{-(d-c_k)^2/(2\sigma^2)}}{\sum_{\ell=1}^{K} e^{-(d-c_\ell)^2/(2\sigma^2)}}, \qquad p_k^{\text{geo}}(t) = \frac{1}{m} \sum_i \eta_k(d_{\text{heat}}(x_i, t)),$$

$$p_k^{\rho}(t) = \frac{\sum_i u_X(x_i, t) \eta_k(d_{\text{heat}}(x_i, t))}{\sum_i u_X(x_i, t)}, \qquad \Delta p_k(t) = p_k^{\rho}(t) - p_k^{\text{geo}}(t).$$

Here p^{geo} records the dimensional composition of the cloud, while Δp measures how that composition changes in heat-concentrated regions. Concatenating p^{geo} and Δp across scales gives $\Phi_{\text{HDS}}(X)$. Further motivation and comparison with low-order summaries are given in Appendix A.4.

3.5 Vectorization and HFS Variants

Across T scales, the per-point heat-geometric feature is

$$\phi(x_i) = [u_X(x_i, t_a),\ d_{\text{heat}}(x_i, t_a),\ \tau(x_i, t_a),\ 2t_a\lambda_1(x_i, t_a), \ldots, 2t_a\lambda_n(x_i, t_a)]_{a=1}^{T}.$$

In $\mathbb{R}^3$, $\phi(x_i)$ has $6T$ components. We evaluate three increasingly expressive HFS representations:

- **HFS-simple:** a fully closed-form descriptor using the global and HDS signatures with fixed statistical pooling of local heat features, *excluding* the log-Hessian spectrum.
- **HFS-desc:** the full closed-form descriptor, adding pooled log-Hessian features to HFS-simple.
- **HFS-full:** uses the same per-point heat features as HFS-desc, but *replaces fixed pooling with a shared MLP* followed by mean and max aggregation to obtain z_X.

Thus the general HFS representation has the form

$$\mathbf{\Phi}(X) = \left[\Phi_{\text{global}}(X),\ \Phi_{\text{HDS}}(X),\ z_X\right],$$

where z_X is obtained by fixed statistical pooling for HFS-simple/HFS-desc and learned pooling for HFS-full. Exact pooling statistics and architectures are given in Appendix C.1.

Unless otherwise stated, we use T logarithmically spaced diffusion times between

$$t_{\min} = c_{\min}\bar{r}_{\mathrm{nn}}^2, \qquad t_{\max} = c_{\max}\,\mathrm{diam}(X)^2.$$

The endpoints track sampling resolution and coarse shape scale, respectively (Appendix A.5). Dense HFS computation costs $O(m^2T)$ for fixed ambient dimension; scalable approximations are discussed in Appendix A.5.

4 Experiments

We evaluate HFS on tasks where class identity depends on multiscale geometry, density, branching, local dimension, or topological structure. We ask three primary questions: *(i)* does explicitly encoding this geometry improve classification, *(ii)* what is the end-to-end computational cost, and *(iii)* how does the representation behave under arbitrary rotations? We additionally compare with multiparameter-persistence methods and ablate the individual HFS components.

4.1 Experimental Setup

Datasets. We use three synthetic density/topology benchmarks, **Processes**, **Orbit5k**, and **DisksAnnuli** (Carrière & Blumberg, 2020; Kerber & Russold, 2024), and four real-world 3D morphology benchmarks: subcellular structures (**Allen** (Vasan et al., 2025; Viana et al., 2023)), neurons (**NeuroMorpho** (Ascoli et al., 2007; Ascoli, 2018)), LiDAR trees (**FOR-species20K** (Puliti et al., 2025)), and protein folds (**SCOP** (Kucera et al., 2023)); see Table 7. These datasets span density variation, branching and filamentary structure, multiscale organization, and local geometric type; SCOP additionally has no canonical orientation and provides a natural rotation stress test. Full construction, preprocessing, point budgets, and evaluation protocols are given in Appendix B. On two boundary tasks dominated by localized or rigid shape, IntrA and ATLAS-1, HFS remains competitive but does not outperform the strongest coordinate-based networks (Appendix C.2).

HFS variants. **HFS-simple** and **HFS-desc** use closed-form, training-free feature extractors (Section 3). **HFS-full** adds a learned pooling head over the per-point HFS features ($\sim$0.2M parameters), while **HFS+DGCNN** augments a DGCNN backbone with HFS features concatenated with coordinates. These variants separate the contributions of closed-form signatures, learned pooling, and backbone integration.

Baselines and evaluation. We compare with FPFH (Rusu et al., 2009), HKS (Sun et al., 2009), PointNet (Qi et al., 2017a), PointNet++ (Qi et al., 2017b), DGCNN (Wang et al., 2019), and Point Transformer (Zhao et al., 2021). All descriptor methods use the same two-layer MLP classifier, while deep baselines follow their published training recipes. Synthetic benchmarks report OA over 20 splits (80/20; Orbit5k 70/30), and real benchmarks report mACC over 5 folds. Rotation-specific baselines are introduced in Section 4.2.

Implementation. All HFS variants use $T = 8$ logarithmically spaced diffusion scales with $c_{\min} = 0.05$ and $c_{\max} = 0.25$, giving 216 dimensions for HFS-simple, 336 for HFS-desc, and $6T = 48$ per-point features in $\mathbb{R}^3$. Full hyperparameters, point budgets, augmentation details, and implementation settings are given in Appendix C.1; scale choices are analyzed in Appendix A.5. Code and configurations are available at
https://anonymous.4open.science/r/Heat-Field-Signatures.

4.2 Results

Point Cloud Classification. Table 1 shows a consistent pattern: an HFS variant achieves the best performance on all seven benchmarks. On the synthetic tasks, the closed-form **HFS-desc** extractor already surpasses all deep baselines on Processes and Orbit5k, while learned pooling raises HFS-full to 99.8% on DisksAnnuli. The gains persist on real morphology: HFS-full or HFS+DGCNN leads on subcellular structures, neurons, and LiDAR trees. The largest real-data margin occurs on SCOP, where HFS-full reaches 60.5% versus 35.7% for the strongest standard deep baseline, a 24.8-point gain using C_α coordinates alone.

Table 1: **Point-cloud classification.** Classical and deep baselines compared with HFS variants. **Bold** = best, underline = second; HFS rows shaded.

	Synthetic (OA over 20 splits)			*Real (mACC over 5 folds)*			
Method	Processes	Orbit5k	DisksAnnuli	Allen	NeuroMorpho	FOR-species	SCOP
FPFH (Rusu et al., 2009)	84.8 ±1.1	30.6 ±1.1	36.9 ±2.0	90.4 ±0.9	58.5 ±1.4	46.7 ±2.2	24.1 ±1.8
HKS (Sun et al., 2009)	85.2 ±1.2	32.8 ±1.1	50.4 ±1.8	79.5 ±0.8	30.7 ±1.3	31.2 ±0.8	20.0 ±1.1
PointNet (Qi et al., 2017a)	52.2 ±4.1	71.6 ±7.3	20.0 ±0.8	90.1 ±1.3	58.9 ±1.4	51.6 ±3.1	33.9 ±3.9
PointNet++ (Qi et al., 2017b)	59.0 ±3.0	89.1 ±1.2	91.2 ±2.4	92.8 ±0.9	64.5 ±2.1	62.8 ±1.5	19.4 ±1.6
DGCNN (Wang et al., 2019)	57.5 ±1.7	90.7 ±0.9	96.9 ±0.7	95.8 ±0.3	59.0 ±1.3	66.3 ±0.9	35.1 ±2.7
Point Tr (Zhao et al., 2021)	60.3 ±1.8	88.5 ±1.5	82.0 ±31.3	95.3 ±0.2	63.7 ±1.2	57.3 ±1.3	35.7 ±2.5
HFS-simple	97.3 ±0.7	89.4 ±0.7	91.4 ±0.9	96.8 ±0.9	65.5 ±1.2	62.8 ±2.3	49.5 ±2.2
HFS-desc	97.5 ±0.6	93.5 ±0.6	92.1 ±0.7	97.5 ±0.5	72.7 ±1.8	66.2 ±2.3	57.2 ±3.2
HFS-full	**97.6 ±0.4**	**97.6 ±0.4**	**99.8 ±0.1**	97.9 ±0.1	74.2 ±2.7	**66.6 ±1.3**	**60.5 ±2.2**
HFS+DGCNN	80.9 ±0.9	93.2 ±0.4	98.8 ±0.3	**98.2 ±0.2**	**77.6 ±1.0**	65.5 ±1.5	58.1 ±2.5

Runtimes. Table 2 reports end-to-end runtime, including feature extraction, training, and inference. Among the accuracy-competitive methods, **HFS-full** has a median per-dataset speed-up of 8.3× over DGCNN,

while **HFS-simple** is about 16.3× faster and requires no learned feature extractor. **HFS-desc** has median per-dataset speed-ups of 2.5–4.6× over PointNet++, DGCNN and Point Transformer, whereas **HFS+DGCNN** trades efficiency for its stronger accuracy on several datasets. Thus the standalone HFS variants achieve the accuracy gains of Table 1 at substantially lower computational cost.

Table 2: **End-to-end runtime (seconds).** Feature extraction, training, and inference are timed identically on one node (1 H200 + 16 CPU workers). $\mathcal{S}_{\mathrm{DGCNN}}$ denotes the median speed-up relative to DGCNN across datasets (> 1 = faster). Lower is better for runtime; **bold** = fastest, underline = second; HFS rows shaded.

Method	Allen	NMorpho	FOR-spec	SCOP	$\mathcal{S}_{\mathrm{DGCNN}}$
FPFH	**43.1**	**38.7**	35.3	28.8	**27.0**
HKS	127.9	42.0	**25.0**	3.8	23.9
PointNet	360.3	274.1	158.4	63.3	4.7
PointNet++	1882.4	1362.4	765.5	84.6	0.9
DGCNN	1773.9	1300.1	724.0	71.1	1.0
Point Tr	988.6	736.2	406.8	96.3	1.8
HFS-simple	105.4	82.3	47.5	3.3	16.3
HFS-desc	401.6	300.6	173.1	**3.1**	4.4
HFS-full	205.4	152.2	90.2	38.1	8.3
HFS+DGCNN	2305.0	1698.0	947.0	75.1	0.8

Rotation invariance. Table 3 evaluates each trained model on the same test clouds before and after an independent random $SO(3)$ rotation (Appendix C.4). The contrast is striking. Because HFS depends only on pairwise distances and eigenvalues of the heat-field Hessian, its underlying representation is invariant by construction.

HFS-simple, HFS-desc and HFS-full show no accuracy change under rotation in these paired checks, using the corrected extraction and feature precision documented in Appendix C.1. This does not imply bitwise invariance: standardization can amplify floating-point perturbations in nearly constant Hessian-eigenvalue channels, as observed in earlier float32 diagnostics. In contrast, standard coordinate-based networks lose 11–48 mACC points on Allen, NeuroMorpho, and FOR-species, often erasing much of their original performance. Moreover, the HFS guarantee is not limited to $SO(3)$: it extends to the full Euclidean group $O(n) \ltimes \mathbb{R}^n$, including rotations, reflections, and translations, in any ambient dimension.

Table 3: **Rotation invariance.** Accuracy change (Δ mACC) under random $SO(3)$ rotations. Standalone HFS is invariant by construction; HFS+DGCNN retains coordinate inputs and is evaluated empirically. DGCNN-SO(3) and VN-DGCNN obtain robustness through augmentation and equivariance, respectively.

Method	Allen	NMorpho	FOR-spec	SCOP	Invariance
FPFH	6.4	8.0	2.3	−1.0	approximate
HKS	28.0	7.1	12.5	2.3	approximate
PointNet	35.8	38.0	39.7	−2.0	none
PointNet++	11.1	30.9	45.1	−1.0	none
DGCNN	19.0	31.5	47.8	**0.0**	none
Point Tr	21.0	34.7	40.7	**0.0**	none
DGCNN-SO(3)	−0.2	0.6	0.6	**0.0**	augmented
VN-DGCNN	0.1	**0.0**	**0.0**	**0.0**	architectural
HFS-simple	**0.0**	**0.0**	**0.0**	**0.0**	exact
HFS-desc	**0.0**	**0.0**	**0.0**	**0.0**	exact
HFS-full	**0.0**	**0.0**	**0.0**	**0.0**	exact
HFS+DGCNN	0.1	0.2	0.4	−0.3	empirical

We also compare with two approaches designed specifically for rotation robustness. **DGCNN-SO(3)** (Esteves et al., 2018) retrains DGCNN with Haar $SO(3)$ augmentation, while **VN-DGCNN** (Deng et al., 2021) uses the rotation-equivariant Vector Neuron architecture. Both largely eliminate the rotation drop, but through augmentation or architectural constraints rather than an

invariant input representation. Interestingly, adding HFS features to the otherwise rotation-sensitive DGCNN almost completely removes its degradation: HFS+DGCNN has a maximum absolute change of 0.4 points across the four datasets, compared with losses of up to 47.8 points for DGCNN alone. This suggests that the invariant HFS channel can strongly stabilize a coordinate-based backbone, although this is an empirical effect rather than a formal guarantee—the HFS features may simply dominate the hybrid prediction. VN-DGCNN also remains well below the best HFS variant on the same split: clean mACC $94.1/51.7/55.4/55.3$ versus $98.3/76.2/69.1/62.7$ on Allen/NeuroMorpho/FOR-species/SCOP. SCOP provides an instructive contrast: conventional networks are already relatively insensitive to rotation because protein structures have no canonical reference frame. Thus robustness on SCOP largely reflects the data distribution, whereas HFS invariance is a representation-level property that holds independently of the dataset.

Comparison with multiparameter persistence. Processes and DisksAnnuli originate in the multiparameter-persistence literature, where density and geometric scale are modeled jointly, making them particularly relevant benchmarks for HFS. We compare against MP-Images, MP-Landscapes, GRIL, MP-HSM-C (Signed Barcodes), and GraphCode variants (GC, GC-NE), using the published results of Kerber & Russold (2024), which also average 20 80/20 splits. This is an indirect comparison: the realizations and generator implementations are not shared, particularly for our Processes variant. Further background on the density–distance bifiltration is given in Appendix D.

Table 4: **Multiparameter-persistence baselines.** Accuracy (%) on the synthetic benchmarks; MP baseline numbers are from Kerber & Russold (2024), using independent data implementations and realizations, so cross-study rankings are not controlled.

Method	DisksAnnuli	Processes
MP-I (Carrière & Blumberg, 2020)	64.1	66.0
MP-L (Vipond, 2020)	37.2	50.2
GRIL (Xin et al., 2023)	74.9	61.1
MP-HSM-C (Loiseaux et al., 2023)	57.0	70.7
GC (Kerber & Russold, 2024)	<u>86.9</u>	<u>83.4</u>
GC-NE (Kerber & Russold, 2024)	82.8	83.1
HFS-desc	92.1	97.5
HFS-full	**99.8**	**97.6**

Table 4 places our results alongside published MP scores for context; the differences do not establish superiority on matched data. HFS captures density–geometry interactions without constructing or vectorizing a persistence module. HFS and persistence remain conceptually complementary, and a controlled comparison would require rerunning both families on identical realizations.

Ablation Studies. Table 5 isolates which parts of the representation drive the gains in Table 1. All ablations use the *closed-form* HFS-desc descriptor rather than HFS-full: the latter adds a learned pooling head over the same six blocks, so ablating each block would require retraining the head for all thirteen configurations, confounding representation content with training noise. The closed-form descriptor isolates the former cleanly. Here $\diamond$ denotes the complete 336-dimensional descriptor—*not* HFS-full—formed by six blocks: global heat-field energies, the heat dimension spectrum (HDS), and pooled statistics of four local channels (density u_X, heat dimension d_{heat}, transition rate τ, and the log-Hessian spectrum). We ablate each block in both directions: keeping only that block (*X only*) or removing it ($\diamond, -$,X). Since these six blocks form exactly the descriptor in Table 1, $\diamond$ reproduces HFS-desc and $\diamond, -$,log-Hessian reproduces HFS-simple; the two tables agree within 0.7 points throughout, providing an internal consistency check. Three findings stand out.

First, **combining blocks generally helps**: on six of seven datasets the best individual block trails the full descriptor, by as much as 5.1 points (FOR-species); on Processes, the global block alone matches the full descriptor to the reported precision. Second, **which block matters is dataset-dependent, and in the direction the method predicts**: the density channel alone nearly closes the gap on DisksAnnuli (89.2 of 92.1), whose classes are defined by sampling density, yet collapses on the subcellular Allen data (27.6), where density carries little class information; conversely, the log-Hessian block, which encodes local anisotropy, is the strongest single block on every real dataset. A descriptor tuned to one geometric cue could not reproduce this pattern. Third, **the leave-one-out direction shows redundancy**: removing a single block other than the log-Hessian costs at most 1.9 points, because each channel is a different derivative of the same heat field and the others partially recover the lost signal. The exceptions are informative rather than incidental: removing the log-Hessian block costs 8.1 points on SCOP and 6.9 on NeuroMorpho – the two datasets whose labels are carried by filamentary structure – and removing all four local blocks together costs up to 9.6 points. The scale-schedule ablation (T, $c_{\min}$, $c_{\max}$) is reported in Appendix C.3.

Table 5: **Component ablation.** We report two types of ablations: the performance of each part alone, and of all parts but one. $\diamond$ = HFS-desc and $\diamond -$ log-Hessian = HFS-simple of Table 1. **Bold** = full descriptor; underline = best single block per column. Avg. is the unweighted mean across all seven benchmarks and is not part of the bold/underline scheme. Same classifier, splits, and folds as main table.

			Synthetic (OA)			*Real (mAcc)*				
Ablation type	**Configuration**	dim	Proc.	Orbit5k	DisksAnn.	Allen	NMorpho	FOR-s..	SCOP	**Avg.**
Full descriptor	HFS-desc ($\diamond$)	336	**97.5**	**93.7**	**92.1**	**97.9**	**72.1**	**66.7**	**57.5**	82.5
Single block only	global only	32	97.5	65.5	53.9	90.4	63.9	58.8	41.1	67.3
	HDS only	64	96.5	87.3	85.4	96.3	59.7	59.3	43.4	75.4
	density only	40	97.1	21.8	89.2	27.6	34.3	39.9	40.9	50.1
	heat-dim only	40	95.3	84.8	76.5	95.8	58.6	58.4	42.8	73.2
	transition only	40	94.9	85.7	74.3	96.5	59.6	57.6	44.2	73.2
	log-Hessian only	120	96.3	91.1	83.2	97.1	69.5	61.5	55.1	79.1
Full minus one block	$\diamond -$ global	304	97.5	93.6	92.1	97.6	71.8	64.7	59.0	82.3
	$\diamond -$ HDS	272	97.4	93.0	91.5	97.7	71.4	65.7	56.9	81.9
	$\diamond -$ density	296	97.4	93.5	90.5	97.7	72.7	66.1	57.7	82.2
	$\diamond -$ heat-dim	296	97.5	93.4	92.2	97.7	72.9	66.7	57.9	82.6
	$\diamond -$ transition	296	97.4	93.6	92.4	97.8	73.3	66.9	57.6	82.7
	$\diamond -$ log-Hessian	216	97.3	89.4	91.4	96.6	65.2	63.4	49.5	79.0
	$\diamond -$ all four local blocks	96	97.6	87.6	87.1	96.3	64.2	61.7	48.0	77.5

4.3 Discussion

Taken together, the experiments support a simple conclusion: when class identity is carried by intrinsic multiscale geometry, it can be advantageous to *compute* that geometry rather than ask a network to discover it from coordinates. Across seven benchmarks, HFS combines strong classification performance, with lower computational cost and exact geometric invariance in the standalone variants. The result is especially striking because much of this advantage is already present in the closed-form HFS descriptor; learned pooling (HFS-full) and backbone integration (HFS+DGCNN) improve it further, but they are not what creates the geometric signal.

More broadly, HFS illustrates a different design principle for point-cloud learning. Instead of treating geometry as an implicit intermediate representation to be learned, we first lift the discrete cloud to a smooth multiscale field and then extract quantities with direct geometric meaning: concentration, intrinsic dimension, anisotropy, and scale transitions. The ablations show that these cues are complementary but partly redundant, while the rotation experiments show a second benefit of explicit structure: properties that require augmentation or specialized architectures in coordinate networks can arise automatically from the representation itself. This suggests that classical geometric analysis and modern learning need not be competing approaches; analytic geometry can provide the representation, with learning reserved for the task-specific readout.

The claim is deliberately not universal. On tasks dominated by rigid global shape or localized appearance, coordinate-based networks remain competitive or stronger (Appendix C.2). HFS targets a complementary regime: irregular data where density, scale, local dimension, and morphology are entangled. The broader opportunity is therefore not simply a new point-cloud descriptor, but a family of representations obtained by interrogating multiscale fields with geometric invariants—of which the signatures studied here are only a first set.

5 Conclusion

We introduced *Heat Field Signatures* (HFS), a multiscale interface that lifts an irregular point cloud to a family of smooth ambient heat fields, making tools from geometric analysis directly accessible to point-cloud learning. The specific signatures developed here provide closed-form global and local geometric descriptors, require no graph, mesh, or persistence computation, and integrate naturally with lightweight or deep classifiers. Across diverse synthetic and real-world benchmarks, the results show that this heat-field interface provides an effective representation for irregular geometric data. Several directions remain open. Sparse kernels and landmarks could scale HFS to very large point clouds, while richer differential and integral invariants could capture geometry beyond the signatures studied here. HFS could also be combined with persistent homology to couple local geometric type with global connectivity. More broadly, HFS suggests a general strategy for representation learning: lift discrete data to a smooth multiscale geometric object, then apply the geometric tools best suited to the task.

AI USE STATEMENT

Generative AI tools were used to assist with debugging experimental code, improving the clarity, organization, and readability of the manuscript, and creating and refining the pipeline figure. The methods, theoretical framework, hypotheses, experimental design, dataset construction and preprocessing, and interpretation of results were developed by the authors. Generative AI was not used to formulate or prove mathematical claims. All AI-assisted code changes were reviewed, tested, and verified by the authors, and all AI-assisted text and visual material were reviewed and revised as needed. The authors take full responsibility for the final content of the paper.

ETHICS STATEMENT

This work develops geometric representations for point-cloud learning and does not involve human subjects, private or personally identifiable information, or deployment in high-risk settings. The datasets used in our experiments are publicly available research datasets spanning synthetic point clouds, subcellular structures, neuronal morphologies, LiDAR trees, and protein structures. Our experiments are intended for methodological evaluation of geometric representations rather than clinical, diagnostic, or other consequential decision-making. We are not aware of material ethical risks specific to the proposed method beyond those generally associated with downstream applications of machine learning.

REPRODUCIBILITY STATEMENT

Appendix B describes the datasets, preprocessing, and point budgets. Appendix C.1 gives the split protocol, HFS hyperparameters, and baseline training settings. Appendices A.5 and C.4 describe the scale schedule and rotation protocol, respectively. Section 3 and Appendix A specify the closed-form HFS descriptors and geometric quantities. Code and configurations are available at `https://anonymous.4open.science/r/Heat-Field-Signatures`.

REFERENCES

Giorgio A. Ascoli. An open repository for single-cell reconstructions of the brain forest. *Scientific Data*, 5:180006, 2018. doi: 10.1038/sdata.2018.6.

Giorgio A. Ascoli, Duncan E. Donohue, and Deepak L. Bhatt. NeuroMorpho.Org: A central resource for neuronal morphologies. *Journal of Neuroscience*, 27(35):9247–9251, 2007. doi: 10.1523/JNEUROSCI.2055-07.2007.

Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1626–1633, 2011.

Magnus Bakke Botnan and Michael Lesnick. An introduction to multiparameter persistence. *arXiv preprint arXiv:2203.14289*, 2022.

Francesco Camastra and Antonino Staiano. Intrinsic dimension estimation: Advances and open problems. *Information Sciences*, 328:26–41, 2016.

Gunnar Carlsson and Afra Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, 2009.

Mathieu Carrière and Andrew Blumberg. Multiparameter persistence image for topological machine learning. *NeurIPS*, 33, 2020.

Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006. doi: 10.1016/j.acha.2006.04.006.

Baris Coskunuzer and Cüneyt Gürcan Akçora. Topological methods in machine learning: A tutorial for practitioners. *arXiv preprint arXiv:2409.02901*, 2024.

Matt De Vries, Reed Naidoo, Olga Fourkioti, Lucas G. Dent, Nathan Curry, Chris Dunsby, and Chris Bakal. Interpretable point cloud classification using multiple instance learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22209–22220, 2025.

Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. Vector neurons: A general framework for SO(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12200–12209, 2021.

Manfredo Perdigao do Carmo. *Riemannian geometry*. 1992.

Herbert Edelsbrunner and Georg Osang. The multi-cover persistence of Euclidean balls. *Discrete & Computational Geometry*, 65(4):1296–1313, 2021. doi: 10.1007/s00454-021-00281-9.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002. doi: 10.1007/s00454-002-2885-2.

Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning SO(3) equivariant representations with spherical CNNs. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

Lawrence C. Evans. *Partial Differential Equations*, volume 19 of *Graduate Studies in Mathematics*. American Mathematical Society, 2nd edition, 2010.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7:12140, 2017. doi: 10.1038/s41598-017-11873-y.

Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1–2):189–208, 1983.

Alexander Grigoryan. *Heat kernel and analysis on manifolds*, volume 47. American Mathematical Society Providence, 2009.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R. Martin, and Shi-Min Hu. PCT: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. doi: 10.1007/s41095-021-0229-5.

Siyuan Huang, Yichen Xie, Song-Chun Zhu, and Yixin Zhu. Spatio-temporal self-supervised representation learning for 3D point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6535–6545, 2021.

Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

Michael Kerber and Florian Russold. Graphcode: Learning from multiparameter persistent homology using graph neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Tim Kucera, Carlos Oliver, Dexiong Chen, and Karsten Borgwardt. Proteinshake: Building datasets and benchmarks for deep learning on protein structures. *Advances in Neural Information Processing Systems*, 36:58277–58289, 2023.

David Loiseaux, Luis Scoccola, Mathieu Carrière, Magnus Bakke Botnan, and Steve Oudot. Stable vectorization of multiparameter persistent homology using signed barcodes as measures. *Advances in Neural Information Processing Systems*, 36:68316–68342, 2023.

Yatian Pang, Wenxiao Wang, Francis E.H. Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 604–621, 2022. doi: 10.1007/978-3-031-20086-1_35.

Stefano Puliti, Emily R. Lines, Jana Müllerová, Julian Frey, Zoe Schindler, Adrian Straker, Matthew J. Allen, Lukas Winiwarter, Nataliia Rehush, Hristina Hristova, Brent Murray, Kim Calders, Nicholas C. Coops, Bernhard Höfle, Liam Irwin, Samuli Junttila, Martin Krůček, Grzegorz Krok, Kamil Král, Shaun R. Levick, Linda Luck, Azim Missarov, Martin Mokroš, Harry J. F. Owen, Krzysztof Stereńczak, Timo P. Pitkänen, Nicola Puletti, Ninni Saarinen, Chris Hopkinson, Louise Terryn, Chiara Torresan, Enrico Tomelleri, Hannah Weiser, and Rasmus Astrup. Benchmarking tree species classification from proximally sensed laser scanning data: Introducing the FOR-species20k dataset. *Methods in Ecology and Evolution*, 16(4):801–818, 2025. doi: 10.1111/2041-210X.14503.

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 (CVPR)*, pp. 652–660, July 2017a.

Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017b.

Can Qin, Haoxuan You, Lichen Wang, C.-C. Jay Kuo, and Yun Fu. PointDAN: A multi-scale 3D domain adaption network for point cloud representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3384–3391, 2008.

Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (FPFH) for 3D registration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3212–3217, 2009.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Jian Sun, Maks Ovsjanikov, and Leonidas J. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009. doi: 10.1111/j.1467-8659.2009.01515.x.

Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 356–369, 2010.

Ritvik Vasan, Alexandra J. Ferrante, Antoine Borensztejn, Christopher L. Frick, Peyton Garrison, Nathalie Gaudreault, Saurabh S. Mogre, Fatwir S. Mohammed, Benjamin Morris, Guilherme G. Pires, Daniel Saelid, Susanne M. Rafelski, Julie A. Theriot, and Matheus P. Viana. Interpretable representation learning for 3D multi-piece intracellular structures using point clouds. *Nature Methods*, 22(7):1531–1544, 2025. doi: 10.1038/s41592-025-02729-9.

Matheus P. Viana, Jianxu Chen, Theo A. Knijnenburg, Ritvik Vasan, Calysta Yan, Joy E. Arakaki, Matte Bailey, Ben Berry, Antoine Borensztejn, Eva M. Brown, et al. Integrated intracellular organization and its variations in human iPS cells. *Nature*, 613(7943):345–354, 2023. doi: 10.1038/s41586-022-05563-7.

Oliver Vipond. Multiparameter persistence landscapes. *Journal of Machine Learning Research*, 21 (61):1–38, 2020.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics*, 38(5):146:1–146:12, 2019. doi: 10.1145/3326362.

Cheng Xin, Soham Mukherjee, Shreyas N Samaga, and Tamal K Dey. Gril: A 2-parameter persistence based vectorization for machine learning. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pp. 313–333. PMLR, 2023.

Xi Yang, Ding Xia, Taichi Kin, and Takeo Igarashi. Intra: 3d intracranial aneurysm dataset for deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2656–2666, 2020.

Maosheng Ye, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. DRINet: A dual-representation iterative learning network for point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7649–7658, 2021.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip H.S. Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16259–16268, 2021.

APPENDIX

A Heat Field Signatures: Details and Geometric Interpretation **14**

A.1 From Point Clouds to Heat Fields 14

A.2 Global Heat-Field Signatures 15

A.3 Local Heat Geometry 15

A.4 Heat Dimension Spectrum 18

A.5 Scale Choice and Computation 19

B Dataset Details **19**

B.1 Synthetic Benchmarks 19

B.2 Real-World Morphology Benchmarks 20

C Experimental Details **20**

C.1 Implementation Details 20

C.2 Boundary Datasets: Where HFS Is Neutral 22

C.3 Scale-Schedule Ablation 22

C.4 Rotation-Invariance Protocol 23

D Multiparameter Persistence for Point Clouds **24**

D.1 One-parameter persistence 24

D.2 Multiparameter persistence 24

D.3 The density–distance bifiltration 24

A HEAT FIELD SIGNATURES: DETAILS AND GEOMETRIC INTERPRETATION

This section collects the derivations and geometric interpretation of the heat-field quantities used in the main paper. The central idea is simple: Gaussian smoothing turns a discrete point cloud into a differentiable multiscale landscape. Global functionals describe the landscape as a whole, while logarithmic derivatives at the sample points describe local geometric type. For general background on heat-kernel methods and geometric analysis, see Grigoryan (2009); Evans (2010); do Carmo (1992).

A.1 FROM POINT CLOUDS TO HEAT FIELDS

A hard probe of local geometry is the ball count

$$N_X(y,r) = \#\{x_i \in X : \|y - x_i\| \le r\}.$$

If the data near y are approximately d-dimensional, then $N_X(y,r) \propto r^d$ over an appropriate scale range. HFS replaces this discontinuous count by the Gaussian-smoothed field

$$u_X(y,t) = (4\pi t)^{-n/2} \sum_{i=1}^{m} \exp\left(-\frac{\|y - x_i\|^2}{4t}\right), \qquad t > 0,$$

where $\sqrt{t}$ acts as a spatial scale. The normalization makes u_X satisfy the heat equation

$$\partial_t u_X = \Delta_y u_X.$$

Geometrically, small t resolves individual points and fine local structure; intermediate scales reveal branches, sheets, clusters, and junctions; and large t retains only coarse organization. HFS measures this evolution without explicitly constructing a mesh, graph Laplacian, or surface.

A.2 Global Heat-Field Signatures

Let $D_{ij} = \|x_i - x_j\|$. Since u_X is a Gaussian mixture, its principal global functionals have closed forms:

$$E_2(t) = \int_{\mathbb{R}^n} u_X(y,t)^2\, dy = (8\pi t)^{-n/2} \sum_{i,j} e^{-D_{ij}^2/8t}, \tag{1}$$

$$C_2(t) = -t\,\partial_t \log E_2(t) = \frac{n}{2} - \frac{\sum_{i,j} \frac{D_{ij}^2}{8t} e^{-D_{ij}^2/8t}}{\sum_{i,j} e^{-D_{ij}^2/8t}}, \tag{2}$$

$$\mathcal{E}(t) = \int_{\mathbb{R}^n} \|\nabla u_X(y,t)\|^2\, dy = -\frac{1}{2}\partial_t E_2(t), \tag{3}$$

$$\widetilde{\mathcal{E}}(t) = \frac{\mathcal{E}(t)}{E_2(t) + \varepsilon}. \tag{4}$$

These quantities answer complementary questions. E_2 measures *concentration*: nearby heat kernels overlap strongly and increase the energy. C_2 measures the rate at which this concentration changes with scale. In an ideal d-dimensional regime,

$$E_2(t) \sim t^{-(n-d)/2}, \qquad C_2(t) \approx \frac{n-d}{2},$$

so C_2 is globally sensitive to codimension. Finally, $\mathcal{E}$ measures spatial roughness of the heat landscape, while $\widetilde{\mathcal{E}}$ removes its overall energy scale.

Thus the global block records not simply the size of the point cloud, but how concentration and roughness evolve as neighboring structures begin to interact.

Heat bodies for visualization. For qualitative visualization only, we also consider superlevel sets

$$\Omega_{\alpha,t}(X) = \{y : u_X(y,t) \geq \alpha M_X(t)\}, \qquad M_X(t) = \max_i u_X(x_i,t).$$

At suitable scales these behave like smooth neighborhoods of the data: filaments produce tube-like regions, surfaces produce slab-like regions, and disconnected structures merge as t increases. Because their volume and boundary geometry require grid evaluation, heat bodies are not used in the main closed-form HFS representation.

A.3 Local Heat Geometry

Global signatures summarize the whole heat landscape, but many discriminative structures—thin branches, junctions, boundaries, and mixed-dimensional interfaces—occupy only a small subset of points. HFS therefore evaluates logarithmic derivatives of the heat field at each x_i.

Let

$$w_{ij}(t) = \exp\left(-\frac{D_{ij}^2}{4t}\right), \qquad W_i(t) = \sum_j w_{ij}(t).$$

Heat dimension. The heat dimension is

$$d_{\mathrm{heat}}(x_i,t) = n + 2t\,\partial_t \log u_X(x_i,t) = \frac{\sum_j D_{ij}^2 w_{ij}(t)}{2t\, W_i(t)}.$$

It is a smooth analogue of the local mass-growth exponent. Indeed, if a neighborhood behaves like a d-dimensional set,

$$u_X(x,t) \approx C(x) t^{-(n-d)/2},$$

then

$$d_{\mathrm{heat}}(x,t) \approx d.$$

Thus in $\mathbb{R}^3$ the values $1, 2, 3$ correspond ideally to filament-, surface-, and volume-like neighborhoods. At scales below the sampling resolution the self-point can dominate and $d_{\mathrm{heat}} \approx 0$, which motivates avoiding excessively small t.

Finite-sample guarantee: informal statement. At a smooth point of a d-dimensional manifold, $d_{\mathrm{heat}}(x,t)$ is close to d when the heat scale lies between the sampling scale and the geometric variation scale. Up to logarithmic factors, the error has the form

$$\underbrace{O(t)}_{\text{geometry/density bias}} \quad \underbrace{+\,O\left(\frac{1}{\sqrt{m\,p(x)\,t^{d/2}}}\right)}_{\text{finite-sample noise}},$$

together with a smaller correction caused by the self-point.

Theorem 1 (Finite-sample approximation of heat dimension)**.** *Let $M \subset \mathbb{R}^n$ be a compact C^3 embedded submanifold without boundary, of intrinsic dimension $1 \le d < n$, with $\mathrm{reach}(M) \ge \tau > 0$. Let p be a C^2 density on M satisfying*

$$0 < p_{\min} \le p \le p_{\max}, \qquad \|\nabla_M p\| \le L p_{\max}, \qquad \|\nabla_M^2 p\| \le L^2 p_{\max}.$$

Fix $m \ge 2$, $0 < \delta < 1$, $t > 0$ and $x \in M$, set $x_1 = x$, and draw $x_2, \dots, x_m$ independently from p. Define

$$\mu(x,t) = (m-1)p(x)(4\pi t)^{d/2}.$$

There exist constants $c, C_0, C_1, C_2, C_3 > 0$, depending only on d, n, and $p_{\max}/p_{\min}$, such that if

$$t(\tau^{-1} + L)^2 \le c, \qquad \mu(x,t) \ge C_0 \log\frac{2}{\delta},$$

then, with probability at least $1-\delta$,

$$\left|d_{\mathrm{heat}}(x,t) - d\right| \le C_1 t(\tau^{-1}+L)^2 + C_2\sqrt{\frac{\log(2/\delta)}{\mu(x,t)}} + \frac{C_3}{\mu(x,t)}.$$

The theorem identifies the scale window $\left(\dfrac{\log(2/\delta)}{m\,p(x)}\right)^{2/d} \lesssim t \ll (\tau^{-1}+L)^{-2}$, or, more intuitively, sampling scale $\lesssim \sqrt{t} \ll$ geometric variation scale. This motivates searching for an intermediate-scale regime. The practical choice $t_{\min} \propto \bar{r}_{\mathrm{nn}}^2$ is data-dependent; this fixed-time theorem does not guarantee that every selected scale satisfies the effective-sample-size condition.

Proof. Write

$$N_0 = \sum_{j=2}^{m} e^{-\|x-x_j\|^2/4t}, \qquad S = \sum_{j=2}^{m} \|x-x_j\|^2 e^{-\|x-x_j\|^2/4t}.$$

Since the self-point contributes one to the denominator and zero to the numerator,

$$d_{\mathrm{heat}}(x,t) = \frac{S}{2t(1+N_0)}.$$

For $\sqrt{t}$ small relative to the reach and density-variation scale, standard local-coordinate estimates approximate M by $T_x M \simeq \mathbb{R}^d$. Using

$$\int_{\mathbb{R}^d} e^{-\|u\|^2/4t}\,du = (4\pi t)^{d/2}, \qquad \int_{\mathbb{R}^d} \|u\|^2 e^{-\|u\|^2/4t}\,du = 2dt(4\pi t)^{d/2},$$

gives

$$\mathbb{E}[N_0] = \mu(x,t)\left[1 + O\big(t(\tau^{-1}+L)^2\big)\right],$$
$$\mathbb{E}[S] = 2dt\,\mu(x,t)\left[1 + O\big(t(\tau^{-1}+L)^2\big)\right].$$

Hence

$$\frac{\mathbb{E}[S]}{2t\,\mathbb{E}[N_0]} = d + O\big(t(\tau^{-1}+L)^2\big).$$

The summands of N_0 are bounded by 1, while $r^2 e^{-r^2/4t} \le 4t/e$. Bernstein's inequality therefore yields, simultaneously with probability at least $1-\delta$,

$$|N_0 - \mathbb{E}N_0| \le C\sqrt{\mu(x,t)\log(2/\delta)},$$

and

$$|S - \mathbb{E}S| \le Ct\sqrt{\mu(x,t)\log(2/\delta)},$$

after absorbing lower-order logarithmic terms under $\mu(x,t) \ge C_0 \log(2/\delta)$. Propagating these deviations through the ratio gives

$$\left|\frac{S}{2tN_0} - \frac{\mathbb{E}S}{2t\,\mathbb{E}N_0}\right| \le C\sqrt{\frac{\log(2/\delta)}{\mu(x,t)}}.$$

Finally,

$$\left|\frac{S}{2t(1+N_0)} - \frac{S}{2tN_0}\right| = \frac{S}{2tN_0(1+N_0)} \le \frac{C}{\mu(x,t)},$$

which gives the stated bound. □

The theorem concerns smooth manifold regions. Junctions, boundaries, and mixed-dimensional interfaces need not satisfy its assumptions; for HFS, such departures are useful signals rather than failure cases.

Log-Hessian spectrum. Heat dimension counts effective directions but does not describe their orientation. For this we use

$$Q_X(x_i,t) = \nabla_y^2 \log u_X(y,t)\big|_{y=x_i}.$$

If

$$\bar{r}_i = \frac{1}{W_i(t)} \sum_j w_{ij}(t)(x_i - x_j),$$

then the closed form is

$$Q_X(x_i,t) = \frac{1}{4t^2}\left[\frac{1}{W_i(t)}\sum_j w_{ij}(t)(x_i-x_j)(x_i-x_j)^\top - \bar{r}_i\bar{r}_i^\top\right] - \frac{I}{2t}.$$

Thus the log-Hessian is simply a scale-normalized local weighted covariance minus the ambient isotropic heat term.

Let

$$\lambda_1(x_i,t) \ge \cdots \ge \lambda_n(x_i,t)$$

be its eigenvalues. For an ideal locally flat d-dimensional structure,

$$2t\lambda_k \approx \begin{cases} 0, & \text{tangent directions,} \\ -1, & \text{normal directions.} \end{cases}$$

In $\mathbb{R}^3$ this gives

local type	$(2t\lambda_1, 2t\lambda_2, 2t\lambda_3)$
filament	$(0,-1,-1)$
surface	$(0,0,-1)$
volume	$(0,0,0)$.

Departures from these ideal patterns encode curvature, anisotropy, branching, boundaries, and other nonhomogeneous structure.

Scale-transition rate. Finally,

$$\tau(x_i, t) = t\,\partial_t d_{\mathrm{heat}}(x_i, t) = \partial_{\log t} d_{\mathrm{heat}}(x_i, t)$$

measures how quickly the visible local dimension changes with scale (computed by finite differences in $\log t$ across the T scales). Homogeneous interior regions have $\tau \approx 0$. A point on a thin branch attached to a surface, for example, can transition from $d_{\mathrm{heat}} \approx 1$ to $d_{\mathrm{heat}} \approx 2$ as the heat radius grows, producing large $|\tau|$. Junctions and boundaries similarly generate transient responses.

The three local quantities therefore have distinct roles:

d_{heat} :	how many geometric directions?
$2t\lambda_k$:	what tangent–normal structure?
τ :	how does the geometry change with scale?

Density-sensitive versus geometry-sensitive channels. Raw heat values depend directly on local sampling density. In contrast, if locally

$$u_X(x, t) \approx \rho(x) t^{-(n-d)/2} \times \mathrm{const},$$

then

$$\log u_X(x, t) \approx \log \rho(x) - \frac{n-d}{2} \log t + \mathrm{const}.$$

A slowly varying multiplicative density factor therefore largely cancels from derivatives of $\log u_X$. This motivates separating

$$\phi^{\rho}(x_i) = [u_X(x_i, t_a)]_{a=1}^{T}$$

from

$$\phi^{g}(x_i) = [d_{\mathrm{heat}}(x_i, t_a), \tau(x_i, t_a), 2t_a\lambda_1(x_i, t_a), \ldots, 2t_a\lambda_n(x_i, t_a)]_{a=1}^{T}.$$

The former deliberately retains density information; the latter emphasizes support geometry, although strong density gradients at the same scale as the kernel can still affect the geometric channels.

A.4 Heat Dimension Spectrum

The local descriptor is pointwise, whereas classification requires a shape-level summary. HDS records the distribution of local heat dimensions across the cloud and across scale.

Let $c_1, \ldots, c_K$ be dimension centers; in $\mathbb{R}^3$ we use $c_k \in \{0, 1, 2, 3\}$. Define

$$\eta_k(d) = \frac{\exp\!\big(-(d - c_k)^2/(2\sigma^2)\big)}{\sum_{\ell=1}^{K} \exp(-(d - c_\ell)^2/(2\sigma^2))}, \qquad \sigma = 0.5.$$

The uniform spectrum and heat-weighted spectrum are

$$p_k^{\mathrm{geo}}(t) = \frac{1}{m} \sum_i \eta_k(d_{\mathrm{heat}}(x_i, t)),$$

and

$$p_k^{\rho}(t) = \frac{\sum_i u_X(x_i, t)\eta_k(d_{\mathrm{heat}}(x_i, t))}{\sum_i u_X(x_i, t)}.$$

Their difference

$$\Delta p_k(t) = p_k^{\rho}(t) - p_k^{\mathrm{geo}}(t)$$

measures whether heat-concentrated regions have a different dimensional composition from the cloud as a whole. The final descriptor is

$$\Phi_{\mathrm{HDS}}(X) = [p_k^{\mathrm{geo}}(t_a), \Delta p_k(t_a)]_{k=1,\ldots,K;\ a=1,\ldots,T}\,.$$

HDS retains information that mean dimension alone loses: for example, a half-filament/half-volume cloud and a purely surface-like cloud can both have mean dimension 2, while their spectra are clearly different. For $K = 4$ and $T = 8$, HDS has only 64 entries and costs $O(mKT)$ once the heat quantities are available.

Table 6: **Geometric meaning of the HFS components.** All quantities arise from the same multiscale heat field but probe complementary aspects of point-cloud geometry.

Component	Level	Main geometric information
$E_2(t)$	Global	Heat concentration / pairwise overlap
$C_2(t)$	Global	Scale response and codimension-sensitive behavior
$\mathcal{E}(t)$	Global	Roughness of the heat landscape
$d_{\text{heat}}(x,t)$	Local	Effective intrinsic dimension
$2t\lambda_k(x,t)$	Local	Tangent–normal structure and anisotropy
$\tau(x,t)$	Local	Boundaries, junctions, and scale transitions
HDS	Shape	Distribution of dimensional types and density–geometry interaction

Table 7: **Benchmark summary.** Synthetic benchmarks report overall accuracy (OA) over repeated train/test splits; real-world benchmarks use stratified 5-fold CV and report mACC (mean class accuracy).

Type	Dataset	Domain	Classes	Clouds	Points	Metric	Evaluation
Synthetic	Processes	Spatial processes	4	4,000	105–321	OA	20× 80/20
	Orbit5k	Dynamical systems	5	5,000	1,000	OA	20× 70/30
	DisksAnnuli	Density + topology	5	5,000	874–2,717	OA	20× 80/20
Real-world	NeuroMorpho	Neurons	9	6,198	1,024	mACC	5-fold CV
	Allen	Subcellular structures	7	8,400	1,024	mACC	5-fold CV
	FOR-species20K	LiDAR trees	15	3,435	1,024	mACC	5-fold CV
	SCOP	Protein folds	15	1,455	128	mACC	5-fold CV

A.5 Scale Choice and Computation

We use T logarithmically spaced scales between

$$t_{\min} = c_{\min}\bar{r}_{\text{nn}}^2, \qquad t_{\max} = c_{\max}\,\text{diam}(X)^2.$$

The lower endpoint tracks sampling resolution, consistent with Theorem 1, while the upper endpoint captures coarse shape organization. For strongly nonuniform clouds we additionally consider the adaptive alternative

$$t_{i,a} = c_a r_i(k)^2,$$

where $r_i(k)$ is the k-th-neighbor distance.

Dense pairwise computation costs $O(m^2T)$ for fixed ambient dimension. Because Gaussian contributions beyond radius $C\sqrt{t}$ decay exponentially, a truncated implementation reduces this to $O(mkT)$ when only k local neighbors contribute appreciably.

B Dataset Details

Table 7 summarizes the seven main benchmarks. The synthetic datasets probe density, scale, and topology in $\mathbb{R}^2$, while the real-world datasets span neuronal, subcellular, tree, and protein morphology in $\mathbb{R}^3$. All two-dimensional synthetic clouds are embedded in $\mathbb{R}^3$ with a zero third coordinate for feature extraction.

B.1 Synthetic Benchmarks

Random Point Processes (Processes). Inspired by Kerber & Russold (2024), we classify Poisson, Matérn-like, Strauss-like, and Baddeley–Silverman processes, representing random, clustered, repulsive, and multiscale spatial interactions. Each class contains 1,000 clouds in $[0,1]^2$, generated by our re-implementation of the four processes. The clipped cluster and finite-step interaction implementations are not established as distribution-matched to the published generators. Because the processes are random, the point count varies between 105 and 321 (mean 194). We report mean accuracy over 20 random 80/20 splits.

Orbit5k. Orbit5k (Carrière & Blumberg, 2020) contains five classes generated by the standard dynamical system with $r \in \{2.5, 3.5, 4.0, 4.1, 4.3\}$, with 1,000 clouds of 1,000 points per class. The hardest classes have similar coarse geometry but differ in fine-scale density structure. We report mean accuracy over 20 random 70/30 splits following prior work.

Disks and Annuli (DisksAnnuli). Following the GraphCode benchmark of Kerber & Russold (2024), each point cloud contains dense disks or annuli embedded in uniform background noise, with object density 2.4–$4.8\times$ the background density. The benchmark jointly tests density-based localization and disk-versus-annulus topology. Point counts are not fixed: as in the original construction, they depend on the sampled areas and densities of the generated shapes and background, yielding 874–2,717 points per cloud. We report mean accuracy over 20 random 80/20 splits.

B.2 Real-World Morphology Benchmarks

NeuroMorpho. This is a balanced nine-class benchmark from complete 3D neuronal reconstructions in NeuroMorpho.Org (Ascoli et al., 2007; Ascoli, 2018): basket, bipolar, ganglion, granule, medium spiny, mitral, Purkinje, pyramidal, and stellate neurons. The resulting dataset contains 6,198 clouds. Each reconstruction is converted to a point cloud by sampling $m = 1{,}024$ points uniformly by arc length along its segments; coordinates are kept in their native micron units. We use stratified 5-fold cross-validation.

Allen. We use the `other_punctate` subset of the 3D subcellular point-cloud benchmark of Vasan et al. (2025), derived from the hiPSC imaging collection of Viana et al. (2023). The seven classes correspond to CETN2, HIST1H2BJ, NUP153, RAB5A, SLC25A17, SMC1A, and SON. Restricting to interphase cells and balancing the classes gives 8,400 clouds. Each is randomly subsampled to $m = 1{,}024$ points. The task primarily distinguishes multiscale spatial distributions and clustering of intracellular structures. We use stratified 5-fold cross-validation.

FOR-species20K. FOR-species20K (Puliti et al., 2025) contains terrestrial laser-scanning point clouds of individual trees. We use the 15 most populated species and balance them to 229 trees each, yielding 3,435 clouds. Each tree is randomly subsampled to $m = 1{,}024$ points. The strongly nonuniform LiDAR density caused by occlusion and viewing geometry provides a useful robustness test. We use stratified 5-fold cross-validation.

SCOP. We use the SCOP structural-classification task from ProteinShake (Kucera et al., 2023). Following standard residue-level point-cloud representations of protein structure, each cloud contains only the C_α backbone coordinates; no sequence, residue identity, side-chain, or chemical features are provided. We select the 15 most populated folds and balance them to 97 proteins each, yielding 1,455 clouds.

Protein structures have variable native resolution, with approximately one C_α point per residue. We initially used a point budget of $m = 1{,}024$, but upsampling shorter proteins created coincident points. These zero-distance duplicates caused the nearest-neighbor scale r_{nn}, and hence $t_{\min}$, to collapse toward zero, corrupting the finest-scale heat features. We therefore use $m = 128$, subsampling longer proteins as needed and reducing duplication; shorter proteins still require repeated sampling. Zero-distance neighbours are excluded from scale selection, while all sampled points remain in the heat sums. Clouds are centered but not orientation-normalized, making SCOP a natural test of rotation behavior. We use stratified 5-fold cross-validation. Because all methods receive coordinates only, these results should not be compared directly with protein models that additionally use sequence, residue identity, or other chemical information.

C Experimental Details

C.1 Implementation Details

Heat field signatures. HFS-simple, HFS-desc and HFS-full share one scale schedule: $T = 8$ diffusion times spaced logarithmically between $c_{\min}\, r_{\mathrm{nn}}^2$ and $c_{\max}\, \mathrm{diam}^2$ with $c_{\min} = 0.05$, $c_{\max} = 0.25$ (Appendix A.5); no per-dataset tuning is performed. Here r_{nn} is the mean distance from each

sample to its nearest *noncoincident* sample. Coincident samples are excluded as neighbours for scale selection but retain their multiplicities in all heat sums. This keeps the schedule well defined for clouds with at least two distinct coordinates. This yields $6T = 48$ features per point in $\mathbb{R}^3$. Each local channel (u_X, d_{heat}, τ, and the three eigenvalues $2t\lambda_k$) is pooled over points at every scale by its mean, standard deviation and 10th/50th/90th percentiles ($5T = 40$ values per channel); concatenating the global (32), HDS (64) and pooled-local (240) blocks gives the 336-dimensional HFS-desc descriptor. **HFS-simple** omits the log-Hessian block and is 216-dimensional. **HFS-full** keeps the same per-point features and replaces the fixed statistical pooling with a learned shared MLP of widths $(64, 128, 256)$ with batch normalization, mean and max pooling, and a 256-unit head with dropout 0.5 ($\sim$0.2M parameters), trained for 250 epochs with Adam at learning rate 10^{-3} under a cosine schedule, batch size 32; its three input blocks are standardized per feature using training-split statistics. **HFS+DGCNN** applies $\log(1 + u_X)$ to the density channels and standardizes all HFS channels using training-split statistics only. It concatenates the resulting 48 per-point HFS features with xyz and feeds the result to a DGCNN whose first k-NN graph is built on xyz only and which is otherwise unmodified.

Classical descriptors. HKS is computed on a symmetric 10-NN graph Laplacian (80 eigenpairs, 16 log-spaced times), and FPFH with Open3D on unit-ball-normalized clouds (normal radius 0.05, feature radius 0.10); both are pooled over points into a fixed-length cloud descriptor. On SCOP, where $m = 128$ makes these radii enclose no neighbours for roughly a third of the clouds, FPFH falls back to 30- and 100-nearest-neighbour neighbourhoods for those clouds.

Classifier for descriptor rows. Every closed-form descriptor row, FPFH, HKS, HFS-simple, and HFS-desc, is standardized and classified by the *same* two-layer MLP with hidden widths $(128, 64)$, trained to convergence (at most 3000 iterations; 1000 for the synthetic HFS rows and the scale ablation). Component ablations retain 3000 iterations and float32 features; main HFS-desc uses float64 features. Holding the classifier fixed across these rows is deliberate: it makes the comparison a comparison of *representations*, not of downstream models.

Deep baselines. PointNet, PointNet++, DGCNN, and Point Transformer use their authors' published optimizers, schedules, and epoch budgets (e.g. DGCNN: SGD with momentum 0.9, weight decay 10^{-4}, cosine schedule, label smoothing 0.2, 250 epochs; Point Transformer: SGD, multistep schedule, 200 epochs), with batch size 32. Training augmentation is the standard recipe of that literature: random rotation about the z axis, random scaling in $[0.8, 1.25]$, random translation, and per-point jitter. We deliberately do not add full $SO(3)$ augmentation to these four, since doing so would depart from the published protocols that produce their reported numbers; Section 4.2 analyzes the consequence and adds two baselines that address rotation directly. **DGCNN-SO(3)** is the same DGCNN architecture under the same recipe, differing *only* in that its rotation augmentation is drawn from the full Haar measure on $SO(3)$ rather than about the z axis alone, so any difference is attributable to the augmentation. **VN-DGCNN** replaces DGCNN's layers with Vector Neurons (Deng et al., 2021), which carry lists of 3-vectors instead of scalars and are therefore rotation-equivariant by construction, with an invariant readout; it uses the same optimizer, schedule and epoch budget as DGCNN. We verified its invariance numerically rather than assuming it: rotating an input changes its logits by 6×10^{-8} relative, i.e. float32 round-off.

Point budget. On the four real benchmarks every method receives identical clouds and point budgets, but preprocessing differs. Deep baselines center each cloud and scale it to the unit ball inside their data loaders, following their published recipes; the HFS features are computed on the raw coordinates, since the HFS scale schedule adapts to each cloud. Raw HFS density and energy channels can retain absolute-size information; this is a comparison of complete pipelines, not a normalization-matched ablation. On the synthetic benchmarks the descriptor methods, which accept variable-size input, use the clouds at native resolution while the deep baselines resample to a fixed budget ($m = 256$ for Processes, $m = 1{,}024$ for Orbit5k and DisksAnnuli), so the two families see the same clouds at different resolutions. Clouds are resampled to a fixed budget m per dataset: $m = 1{,}024$ for the subcellular, neuron and tree benchmarks, whose raw acquisitions contain far more points than that (Allen images, for example, provide 20,480), so the budget is a genuine *sub*sample. Proteins are the exception: a protein has a native resolution of one point per residue (median 189 in our subset), so a budget of 1,024 would not subsample but *duplicate* real residues

roughly five-fold, inventing structure the molecule does not have and injecting spurious density variation into a density-sensitive descriptor. We therefore use $m = 128$ for SCOP. Proteins with more than 128 residues, including the median 189-residue protein, are subsampled; shorter proteins require repeated sampling. This budget reduces duplication relative to $m = 1{,}024$. We note the consequence honestly: PointNet++ is the one baseline whose accuracy falls at the lower budget, because it alone builds its hierarchy by repeatedly subsampling the input ($m \to m/2 \to m/8$ centroids), whereas the k-NN-based baselines and the descriptor rows improve at the lower point budget.

Protocol and hardware. Synthetic benchmarks use 20 random stratified splits (80/20; Orbit5k 70/30), with the split seeds shared by every method; real benchmarks use stratified 5-fold cross-validation, with the fold assignment fixed once and shared by every method, so all methods are trained and evaluated on exactly the same partitions. Means and population standard deviations are computed across splits or folds; differences are computed before rounding. The corrected extractor handles repeated coordinates and collinear clouds without imputation. Invalid inputs or nonfinite features raise explicit errors; no clouds or failed channels are silently dropped or zero-filled. Neural reruns use one dedicated NVIDIA GPU per job, with CPU extraction for the closed-form descriptors. All runtimes in Table 2 use one H200 and 16 CPU workers with the same software environment, and all methods for a dataset are measured on the same allocated node. Timings include fresh feature extraction, normalization and model setup, training, and test-feature extraction and inference; cached features are not timed as fresh computation.

C.2 Boundary Datasets: Where HFS Is Neutral

HFS is designed for labels carried by intrinsic multiscale geometry. It is worth stating plainly where that assumption does *not* hold, both because it sharpens the claim and because the boundary is informative.

We report two such datasets. **IntrA** (Yang et al., 2020) asks whether a reconstructed vessel segment contains an intracranial aneurysm, a single localized bulge on an otherwise smooth tube, so the label is a property of one region of the surface rather than of the multiscale organization of the whole cloud. **ATLAS-1** (De Vries et al., 2025) classifies drug treatment from the reconstructed surface of a single cell, where the discriminative signal is largely overall cell shape and size.

Table 8 shows the expected pattern: coordinate-based networks are at least as strong as HFS on both, and the margins that HFS enjoys on the morphology benchmarks disappear. HFS remains competitive rather than uncompetitive. On IntrA the best HFS variant is second by mACC and 2.8 points behind DGCNN; on ATLAS-1 it is 2.0 points behind Point Transformer. But neither is a task where HFS is the right tool, and we do not claim it is.

We read this as support for the paper's thesis rather than against it. A representation that computes density, intrinsic dimension and scale transitions should help exactly when those quantities carry the label, and should be neutral when they do not; a method that won everywhere would be evidence that the benchmarks, not the representation, were doing the work. The practical guidance follows directly: use HFS where the class is intrinsic morphology, and a coordinate encoder where it is rigid shape or localized appearance.

Table 8: **Boundary datasets** (mACC / macro-F1, %). Both labels are carried by rigid or localized shape rather than by intrinsic multiscale geometry, and HFS is neutral there. IntrA uses the official split; ATLAS-1 follows the protocol of De Vries et al. (2025); DGCNN and HFS+DGCNN are averaged over three seeds per fold. **Bold** = best, underline = second.

Method	IntrA	ATLAS-1
FPFH	81.6 / 82.1	74.6 / 74.5
HKS	62.6 / 64.1	50.3 / 50.3
PointNet	71.5 / 73.6	74.9 / 74.3
PointNet++	89.0 / 90.9	76.1 / 76.1
DGCNN	**92.6** / **91.9**	76.3 / 76.2
Point Tr	79.0 / 80.4	**78.1** / **78.0**
HFS-simple	79.3 / 80.6	72.9 / 72.8
HFS-desc	85.2 / 86.0	74.9 / 74.9
HFS-full	87.6 / 88.6	74.6 / 74.6
HFS+DGCNN	89.8 / 91.0	76.1 / 76.1

C.3 Scale-Schedule Ablation

The scale schedule has three constants: the number of diffusion times T and the range multipliers $c_{\min}, c_{\max}$. Table 9 varies them on one synthetic and one real dataset, holding everything else fixed.

Accuracy increases monotonically with T:
on NeuroMorpho $59.8 \to 69.7 \to 72.1 \to 74.5$ for $T = 2, 4, 8, 16$, with the step from 8 to 16 worth 2.4 points for twice the descriptor length. Two scales underperform the larger tested grids; the later gains are much smaller than the gain from $T = 2$ to $T = 4$. We use $T = 8$ throughout as an accuracy/size compromise. In the two datasets tested here, increasing to $T = 16$ improves HFS-desc by 0.1 points on Processes and 2.4 points on NeuroMorpho.

Sensitivity to the range is small. Across the five $(c_{\min}, c_{\max})$ settings accuracy spans 0.5 points on Processes (97.4–97.8) and 1.1 on NeuroMorpho (72.1–73.2), comparable to the split- or fold-to-fold standard deviation. The defaults $(0.05, 0.25)$ are therefore not a tuned choice, and no per-dataset tuning is performed anywhere in the paper.

Table 9: **Scale-schedule ablation.** Top: number of scales T (range fixed at $c_{\min} = 0.05$, $c_{\max} = 0.25$). Bottom: range multipliers at $T = 8$. Processes: OA over 20 splits; NeuroMorpho: mACC over 5 folds. The defaults used throughout are $T = 8$, $(0.05, 0.25)$ (shaded).

Setting	dim	Processes	NeuroMorpho
$T = 2$	84	94.8 ±0.8	59.8 ±1.7
$T = 4$	168	97.3 ±0.5	69.7 ±1.5
$T = 8$	336	97.5 ±0.7	72.1 ±1.6
$T = 16$	672	97.6 ±0.6	74.5 ±1.2
$(c_{\min}, c_{\max})$			
$(0.01, 0.25)$	336	97.8 ±0.6	72.6 ±1.0
$(0.05, 0.10)$	336	97.4 ±0.6	73.2 ±1.5
$(0.05, 0.25)$	336	97.5 ±0.7	72.1 ±1.6
$(0.05, 0.50)$	336	97.6 ±0.6	72.5 ±1.2
$(0.10, 0.25)$	336	97.6 ±0.6	73.0 ±1.0

C.4 Rotation-Invariance Protocol

Protocol. Each method is trained *once*, on the unmodified training split, and then evaluated twice: on the original test clouds, and on the *same* test clouds after each has been rotated by an independently drawn random rotation. No method is trained on rotated test clouds; the only rotations seen in training are the augmentation of each recipe (rotation about z for the coordinate networks except DGCNN-SO(3), which uses full $SO(3)$; HFS-full uses fixed invariant features), and no model is retrained between the two evaluations, so the reported quantity isolates the sensitivity of the learned or computed representation to a change of frame. The reported Δ is the balanced accuracy on the original test set minus the balanced accuracy on the rotated test set, so 0 means the rotation had no effect and positive values are accuracy lost. These are paired differences for a fixed trained model and test split. Negative values indicate measured gains after rotation and are retained as observed. All three standalone HFS variants have zero measured accuracy change in the corrected runs; numerical conditioning remains a separate issue from the underlying geometric invariance.

Sampling rotations. Rotations are drawn uniformly from $SO(3)$ under the Haar measure. We sample a matrix with i.i.d. standard normal entries, take its QR decomposition, correct the column signs by the signs of the diagonal of R, and negate one column if the determinant is -1; this yields a uniformly distributed element of $SO(3)$ rather than the non-uniform distribution obtained from uniformly sampled Euler angles. Each test cloud receives its own independent rotation.

Feature-level check. Accuracy alone cannot distinguish a representation that is genuinely invariant from one that merely happens to be robust on a particular dataset, so we also measure invariance directly on the features. For each test cloud we compute the relative deviation $\|\Phi(X) - \Phi(RX)\|_2 \,/\, \|\Phi(X)\|_2$ between the descriptor of the original cloud and of its rotated copy, and average over the test set. These mean deviations range from about 5×10^{-8} to 6×10^{-6} for the complete standalone inputs and the hybrid's HFS channels. For HFS-full, all three input blocks are also checked separately; the largest blockwise mean is 9.4×10^{-6}. For HFS+DGCNN the feature check excludes xyz, which is not invariant. The small feature perturbations are consistent with the standalone invariance guarantee, but do not certify bitwise equality or hybrid invariance.

Measuring canonical orientation. For each cloud we take the principal axis $v(X)$, the unit eigenvector of the covariance matrix with the largest eigenvalue, resolving its sign ambiguity by fixing a common hemisphere. We measure alignment by $\|\frac{1}{N}\sum_i v(X_i)\|_2$. Axes concentrated around a common direction give values close to 1. For uniformly distributed unoriented axes in three dimensions, hemisphere fixing gives a limiting mean norm of $1/2$, rather than zero. The measured scores are 0.94 for FOR-species and 0.44 for SCOP, indicating substantially weaker principal-axis alignment in SCOP. This statistic alone does not establish that SCOP orientations are uniformly distributed.

D MULTIPARAMETER PERSISTENCE FOR POINT CLOUDS

This appendix gives a brief, self-contained account of multiparameter persistence and of the density–distance bifiltration used by the topological baselines in Section 4.2. We assume familiarity with homology but not with persistence.

D.1 ONE-PARAMETER PERSISTENCE

Persistent homology summarizes how the topology of a point cloud evolves across a single scale. Given X, one builds a growing family of simplicial complexes (a *filtration*) indexed by a scale $r \geq 0$, most commonly the Vietoris–Rips complex

$$\mathrm{Rips}_r(X) = \big\{\, \sigma \subseteq X : \|x_i - x_j\| \leq r \text{ for all } x_i, x_j \in \sigma \,\big\}.$$

As r increases, simplices are only added, so $\mathrm{Rips}_r(X) \subseteq \mathrm{Rips}_{r'}(X)$ for $r \leq r'$. Applying k-th homology yields a sequence of vector spaces linked by inclusion-induced maps, a *persistence module*, whose structure is captured completely by a *barcode*: a multiset of intervals recording the scale at which each topological feature (component, loop, void) is born and dies. Barcodes are stable to perturbations of X (Edelsbrunner et al., 2002) and can be vectorized for machine learning.

Why one parameter is not enough. A single scale conflates density with geometry. In a cloud with nonuniform sampling, the scale needed to connect a sparse filament already collapses a dense cluster, and outliers create short-lived features indistinguishable from genuine sparse structure. Distinguishing a dense annulus from background noise, or a repulsive point process from a clustered one, requires reasoning about topology *and* density jointly, which a one-parameter filtration cannot express.

D.2 MULTIPARAMETER PERSISTENCE

Multiparameter persistence addresses this by filtering along two (or more) parameters at once. One indexes complexes by a pair $(s, r) \in \mathbb{R}^2$ that is monotone in each coordinate, obtaining a *bifiltration* $\{K_{s,r}\}$ with $K_{s,r} \subseteq K_{s',r'}$ whenever $s \leq s'$ and $r \leq r'$. Homology now produces a *two-parameter persistence module*: a grid of vector spaces $M_{s,r} = H_k(K_{s,r})$ with commuting maps induced by the inclusions (Botnan & Lesnick, 2022; Coskunuzer & Akçora, 2024).

The essential difficulty is that, unlike the one-parameter case, two-parameter modules admit *no* complete discrete invariant analogous to the barcode (Carlsson & Zomorodian, 2009). There is no finite list of intervals that recovers the module up to isomorphism. Practical pipelines therefore replace the (intractable) full module with stable *feature maps* that send the module to a vector or graph usable by a classifier. The baselines in Section 4.2 are instances of this: multiparameter persistence images and landscapes, GRIL, and Hilbert signed-measure convolutions each vectorize the module for a downstream classifier, while graphcodes embed it as a graph for a GNN (Carrière & Blumberg, 2020; Vipond, 2020; Xin et al., 2023; Loiseaux et al., 2023; Kerber & Russold, 2024).

D.3 THE DENSITY–DISTANCE BIFILTRATION

For point clouds, the canonical choice of two parameters is *distance* (geometric scale) and *density*. Fix a density estimate; a common choice is a Gaussian kernel density estimator with bandwidth $\delta > 0$,

$$\rho_\delta(x) = \frac{1}{m} \sum_{j=1}^{m} \exp\!\left(-\frac{\|x - x_j\|^2}{2\delta^2} \right),$$

and let $\gamma(x) = -\rho_\delta(x)$ be the corresponding *codensity*, so that low codensity marks dense regions. The density–distance bifiltration is then built in two nested steps. For a density threshold s we first restrict to the sufficiently dense subsample $X_s = \{\, x \in X : \gamma(x) \leq s \,\}$, and then build the Rips complex at scale r on that subsample, $K_{s,r} = \mathrm{Rips}_r(X_s)$. Both operations are monotone, raising s admits sparser points, raising r adds simplices, so $\{K_{s,r}\}_{(s,r)}$ is a genuine bifiltration and its k-th homology is a two-parameter module. Intuitively, the density axis peels the cloud from its densest core outward, while the distance axis grows connectivity at each density level; a topological feature is characterized by the *region* of the (s, r)-plane over which it persists, not by a single birth–death pair.

Related density-aware constructions replace the KDE sublevel filtration with a distance-to-measure or multicover filtration (Edelsbrunner & Osang, 2021), but the two-parameter density/scale structure is the same.

Why this matches the benchmarks. This construction is exactly what the Processes and DisksAnnuli benchmarks are designed to probe. On DisksAnnuli, an annulus contributes an H_1 loop that is present only within the dense-object region (a band on the density axis) and only across a range of scales (a band on the distance axis); a disk contributes none, and uniform background noise contributes neither, so the classes separate jointly in (s, r) though not in either parameter alone. On Processes, attractive, repulsive, and multiscale interactions imprint distinct joint density–scale signatures. HFS targets this same density–geometry entanglement, but through closed-form differential channels of the heat field (Section 3) rather than by computing and vectorizing a persistence module.