

# **Topological Feature Embedding and Representation for Machine Learning on 3D Point Clouds**

**Yuhe Qin, April 23**

- Machine Learning on 3D Point Clouds
- Topological Feature Representation Mechanism
- Integration with Point Cloud Learning
- Experiments

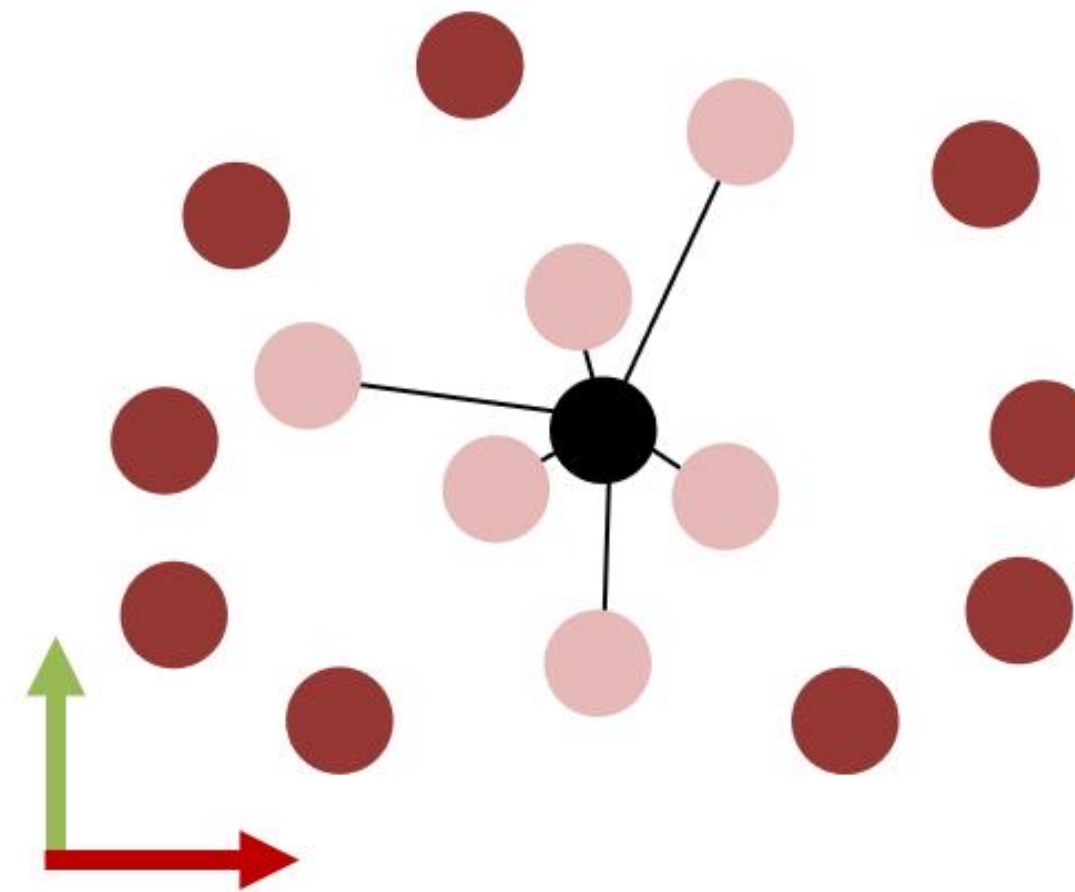
# Machine Learning on 3D Clouds

# Three Key Features of Point Clouds

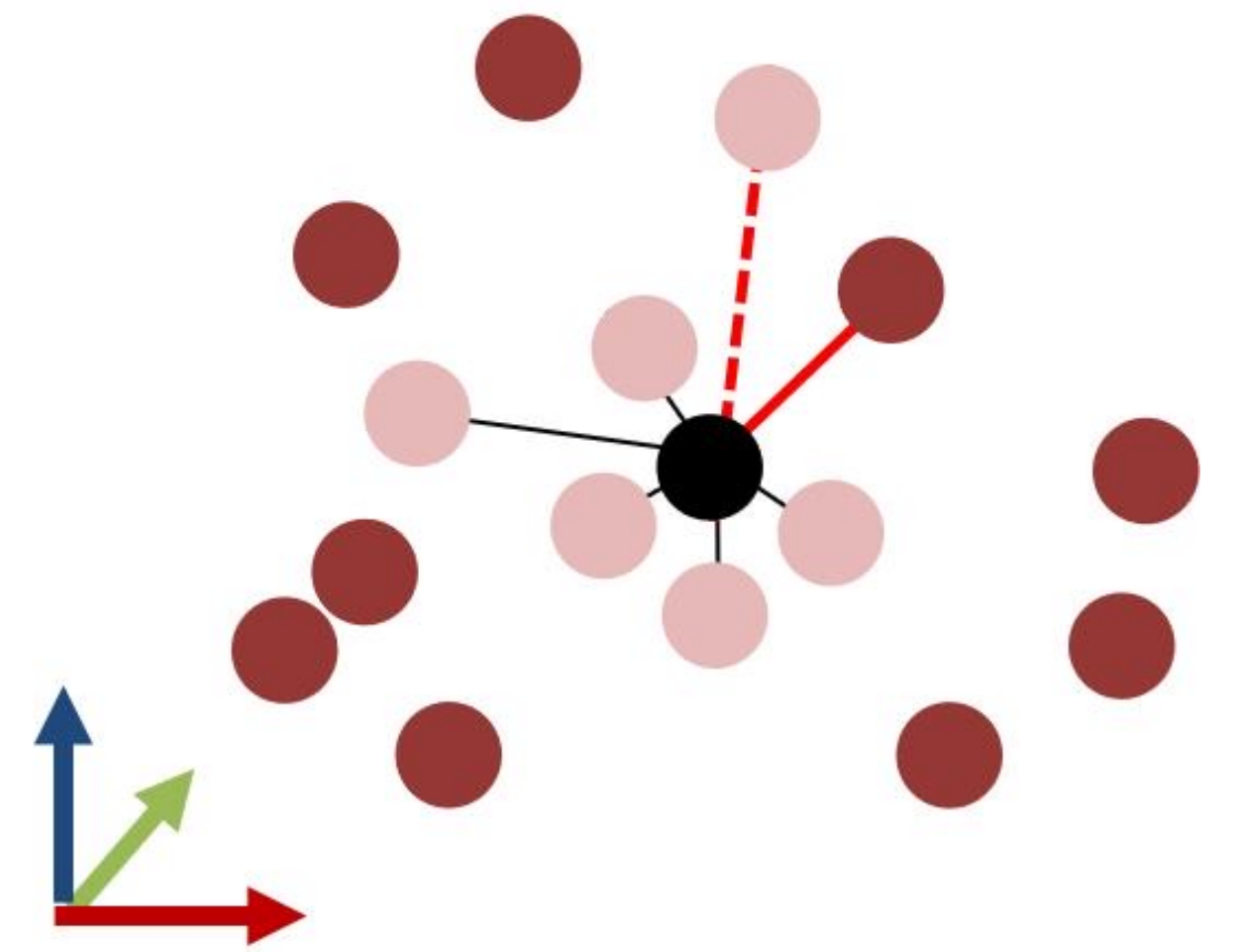
- Unstable Connectivity Modeling

Point neighborhood as kNN  
(k=6)

2D kNN  
with view from the top

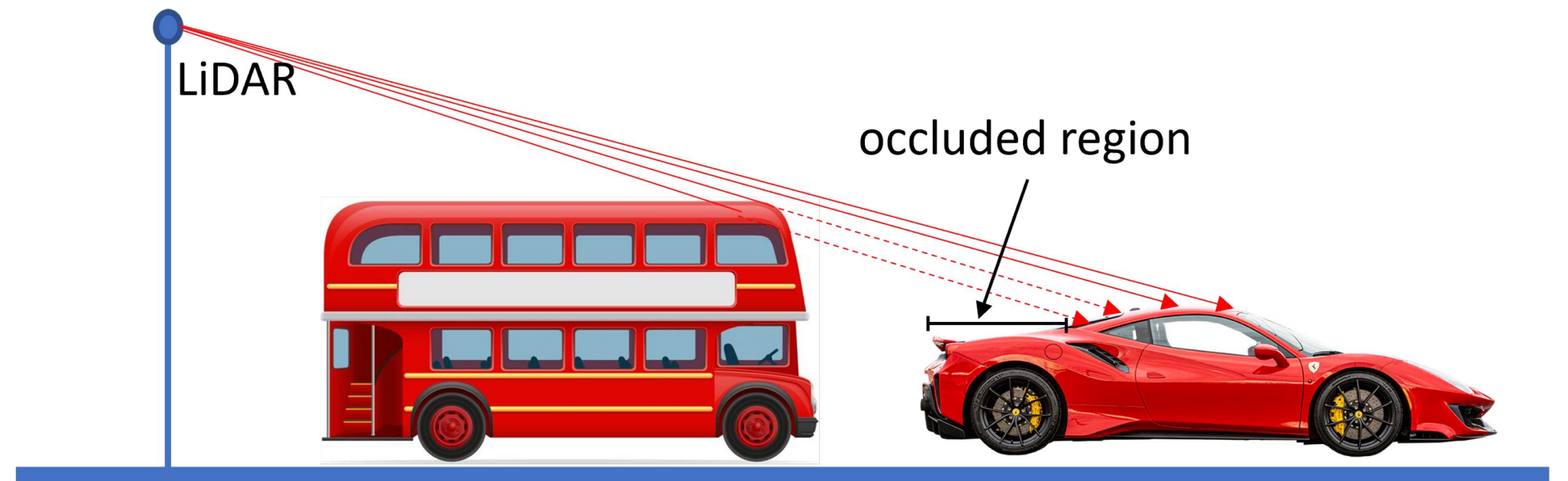


3D kNN  
with view from the side



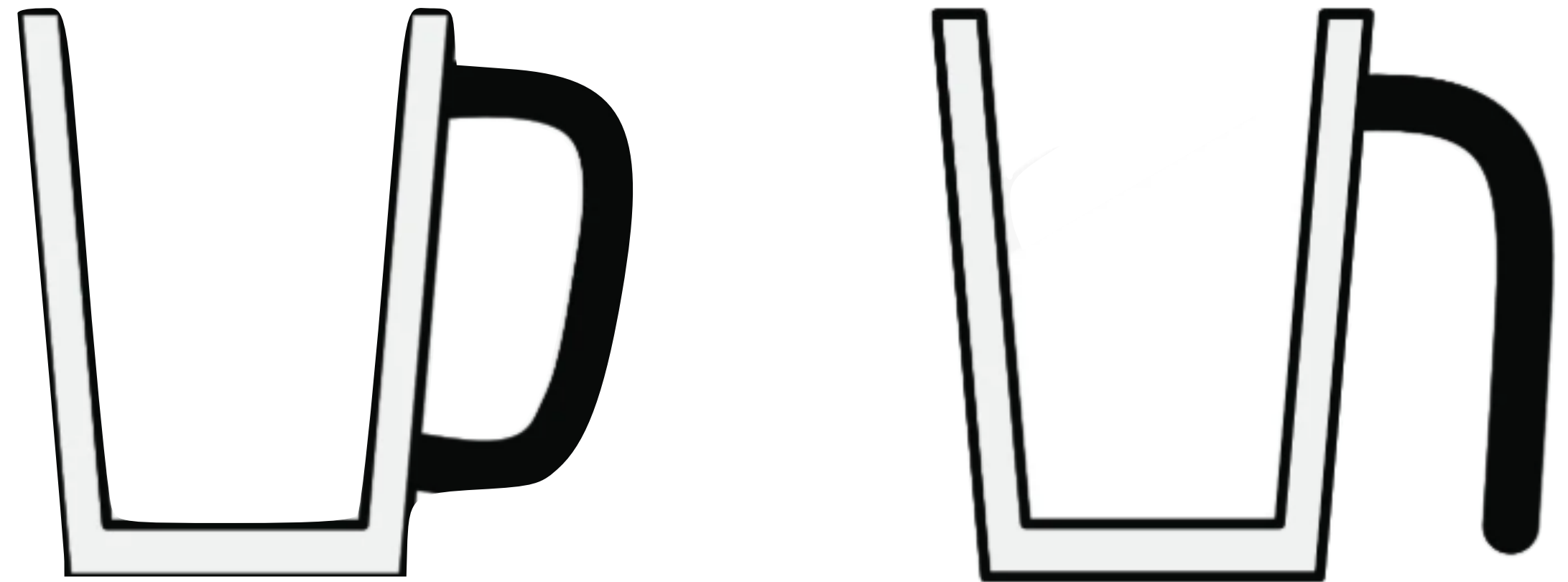
# Three Key Features of Point Clouds

- Unstable Connectivity Modeling
- Irregularly and Incompletely Sampled



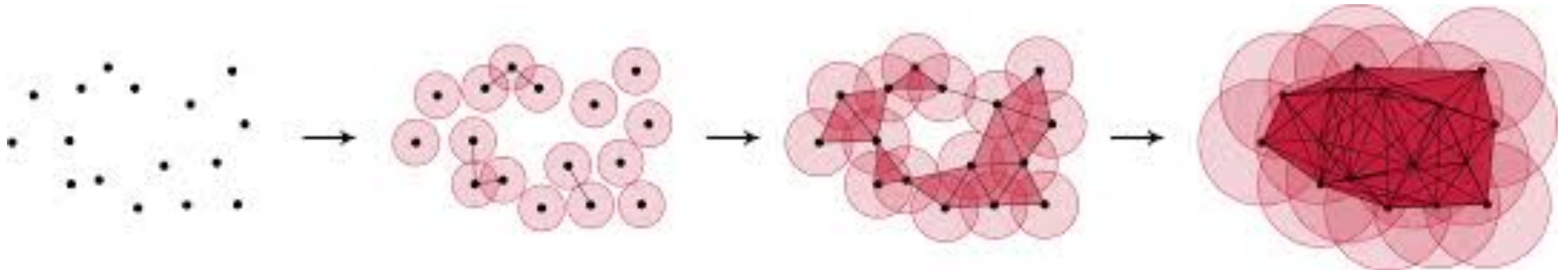
# Three Key Features of Point Clouds

- Unstable Connectivity Modeling
- Irregularly and Incompletely Sampled
- Limited Perception of Global Structure



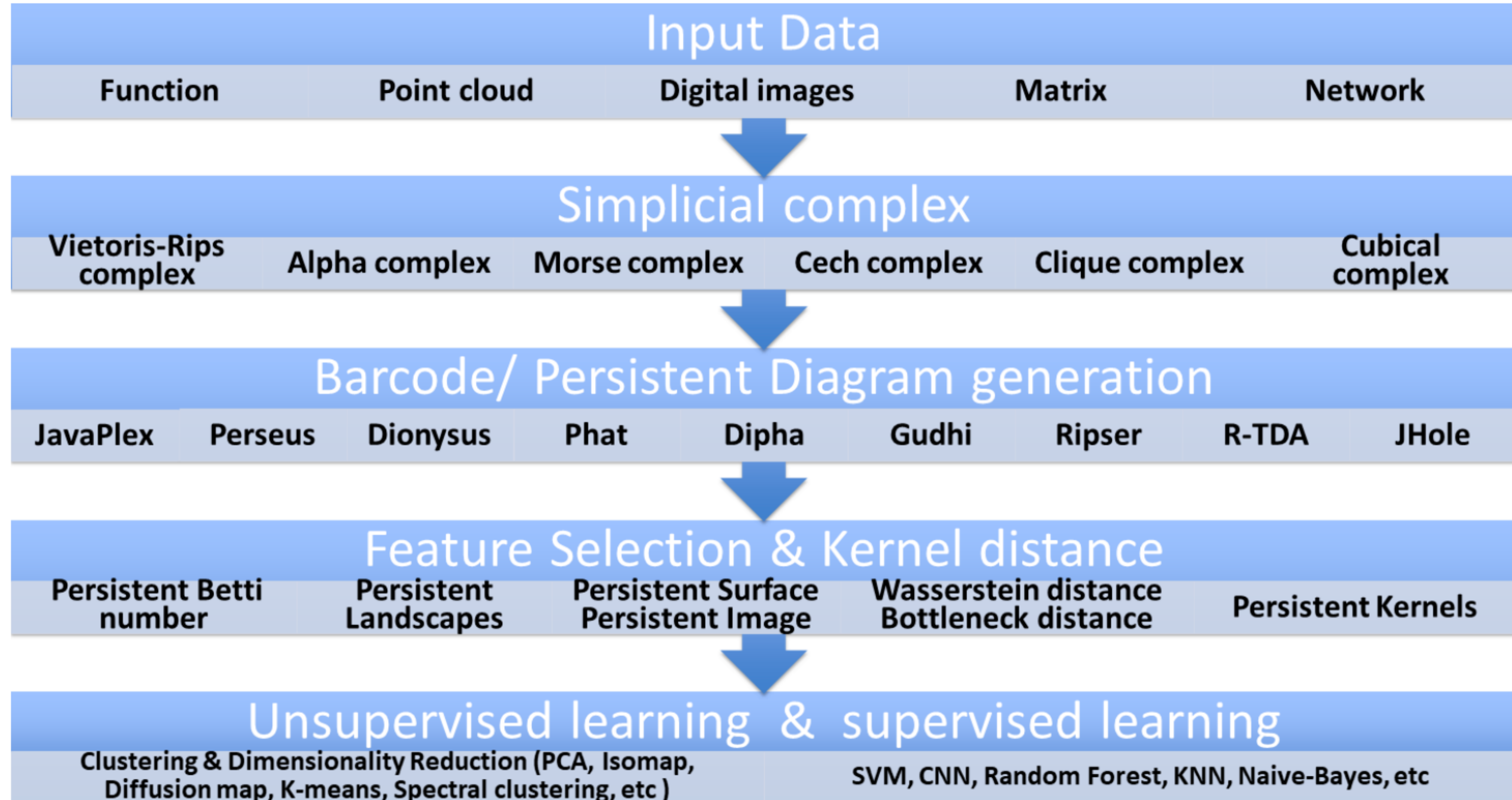
# How Topology Helps?

- Unstable Connectivity Modeling → **PH does not require predefined connectivity**
- Irregularly and Incompletely Sampled → **PH provides a multi-scale view**
- Limited Perception of Global Structure → **Topology describes global structure**



# PHML

$$X \rightarrow VR_{\epsilon}(X) \rightarrow \{VR_{\epsilon_j}(X)\}_{j=1}^m \rightarrow (D_0, D_1) \rightarrow z_{\text{topo}}$$



**TopoRec, TopoSeg, TopologyNet,  
TopoDiT-3D, RipsNet, STITCH.....**

# Topological Feature Representation Mechanism

# Global Topological Representations

Persistent homology is computed from the whole point cloud to obtain a global topological descriptor.

Pipeline:

$$X \rightarrow \text{Vietoris-Rips filtration} \rightarrow \mathcal{D}(X) \rightarrow z_{\text{topo}}$$

Vectorization:

$$z_{\text{topo}} = \psi(\mathcal{D}(X))$$

Common choices:

- Persistence image
- Persistence landscape
- Betti curve

# Local / Point-wise Topological Representations

Instead of computing topology from the whole shape, compute topology around each point or local region.

$$h_i^{\text{topo}} = \psi(\mathcal{D}(\mathcal{N}(x_i)))$$

where  $\mathcal{N}(x_i)$  is the neighborhood of point  $x_i$ .

# Learnable Topological Representations

Topology is no longer only a fixed precomputed descriptor. It is integrated into the learning process.

$$z_{\text{latent}} = f_{\theta}(X, z_{\text{topo}})$$

# Integration with Point Cloud Learning

# Feature-level Augmentation

Topology is used as an additional input feature.

$$z = [z_{\text{geo}}, z_{\text{topo}}]$$

# Representation-level Integration

Topology is inserted into the intermediate feature representation of the model.

$$z_{\text{latent}} = f_{\theta}(z_{\text{geo}}, z_{\text{topo}})$$

# Loss-level Structural Constraints

Topology is used as a constraint in the loss function.

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{topo}}$$

# TopoRec

Modification	Recall @1	Recall 1%
Original	100	100
5%-jitter ( $\mu : 0, \sigma : 10^{-3}$ )	97.11	100
+5%-scaling	75	99.04
-5%-scaling	56.73	99.04
5%-downsampling	79.81	99.04
5%-upsampling	39.42	75.96
[0.2, 0.2, 0.1]-translation	100	100
45°/90°/135°-rotation	100	100

Table 3. Recall results for the Shapenet dataset.  $b = 10$

---

## Algorithm 1: TopoRec

---

**Data:**  $Q, \mathcal{D} = \{P_1, P_2, \dots, P_n\}, b$

**Result:**  $P^* \in \mathcal{D}$

- 1 Onetime pre-processing: generating a suitable persistence**  $\forall P \in \mathcal{D}$ . Generate persistence **pers** $_P$  (a list of birth-death pairs) for every  $P \in \mathcal{D}$  in homology dimension 2. If the **pers** $_P$  has less than 2 birth-death pairs, generate the persistence in homology dimension 1 and use it as **pers** $_P$ . Use the persistence in dimension 0 as **pers** $_P$ , if dimension 1 fails to generate at least 2 birth-death pairs;
  - 2 Onetime pre-processing: generating a vector**  $\forall P \in \mathcal{D}$ . Generate a  $b$ -length ATOL descriptor vector  $v_P$  using **pers** $_P$ , for every  $P \in \mathcal{D}$ , where  $v_P = [a_{P1}, a_{P2}, \dots, a_{Pb}]$ ;
  - 3 Generate the query  $b$ -length ATOL vector**  $v_Q = [c_1, c_2, \dots, c_b]$  for  $Q$  using its persistence, obtained using the same approach used for the point clouds in  $\mathcal{D}$ ;
  - 4 Return the point cloud**  $P^* \in \mathcal{D}$  such that  $P^* = \arg \min_{P \in \mathcal{D}} \|v_P - v_Q\|$ , where  $\|\cdot\|$  denotes the Euclidean distance in  $b$ -space;
-

# TopoSeg

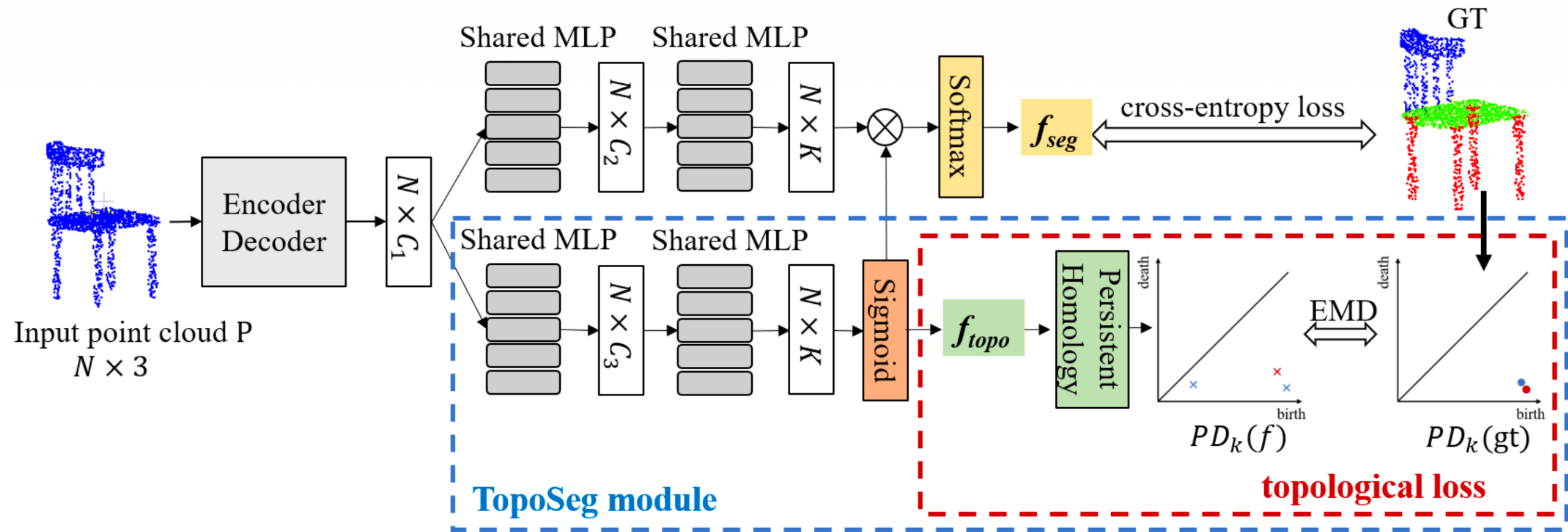
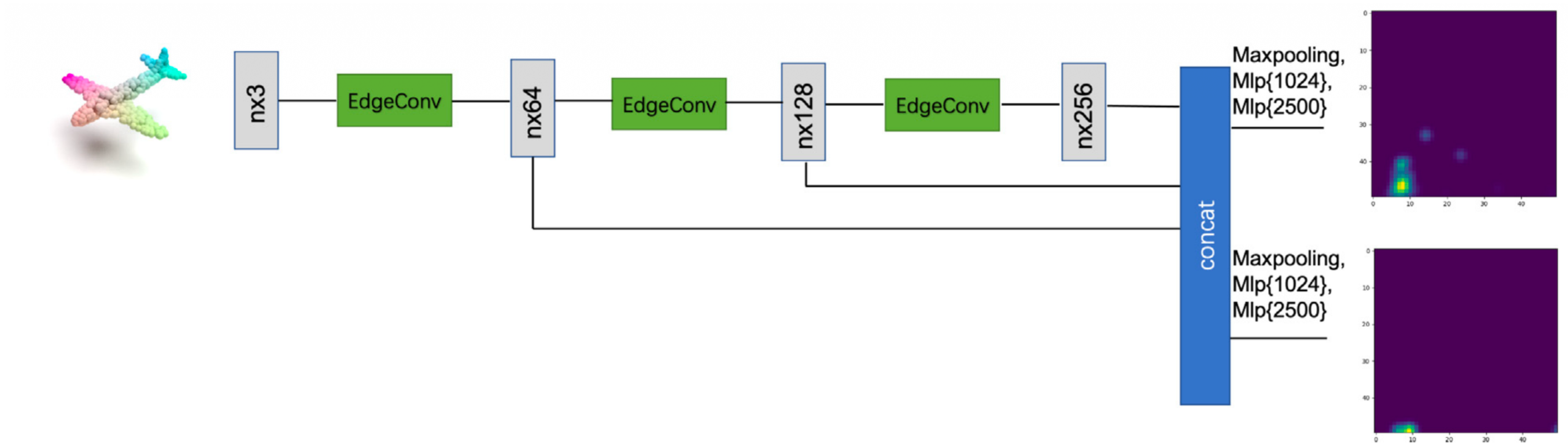


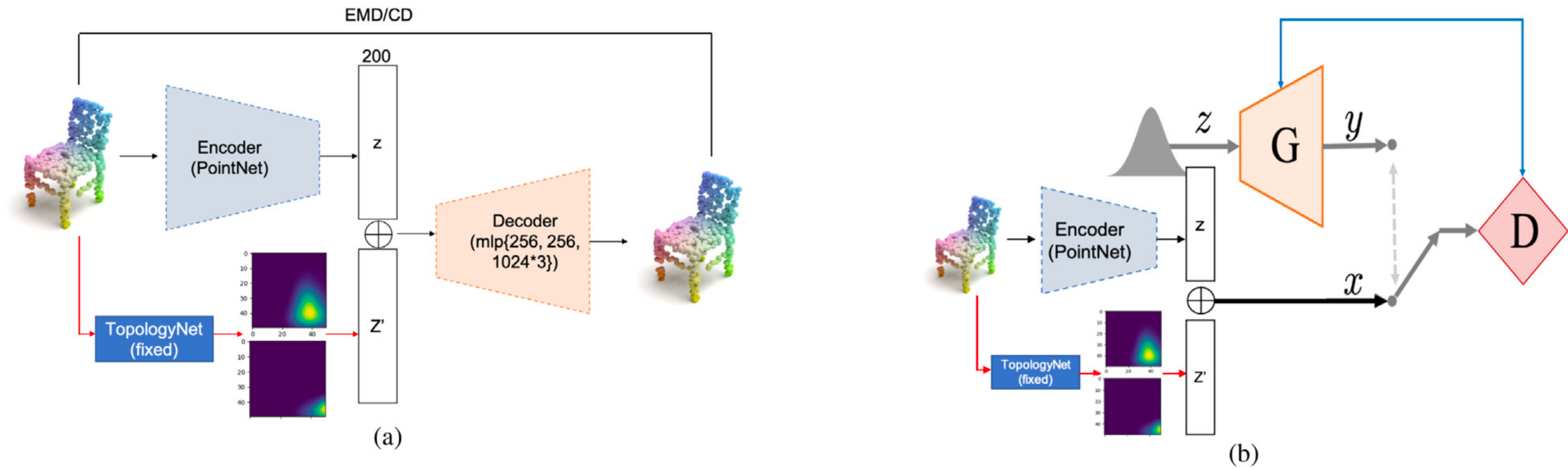
Figure 2: Point cloud segmentation network architecture embedded with the proposed TopoSeg module and topological loss. The upper branch is the Point cloud segmentation network, and the bottom branch is the TopoSeg module.

# TopologyNet



**Fig. 2.** The architecture of the proposed TopologyNet.

# TopologyNet



**Fig. 3.** (a) The architecture of the proposed topological autoencoder; (b) the architecture of  $l$ -GAN with topological autoencoder.

<b>Representation type ↓ / Integration type →</b>	<b>Feature-level</b>	<b>Representation-level</b>	<b>Loss-level</b>
<b>Global PH descriptor</b>	<b>TopoRec</b> ; PH features + vectorization for point-cloud recognition/retrieval	<b>TopoDiT-3D</b> ; PH information injected into diffusion transformer feature learning	
<b>Local / point-level PH feature</b>	<b>Beksi &amp; Papanikolopoulos</b> ; topological persistence for point-cloud segmentation	<b>Node-Level Topological Representation Learning / TOPF</b> ; point-level PH-related representation learning	<b>TopoSeg</b> ; PH-based topological loss for point-cloud segmentation
<b>Learnable / differentiable topology</b>	<b>RipsNet</b> ; learned estimator of vectorized persistence diagrams	<b>TopologyNet</b> ; end-to-end learning of topological representations from 3D point clouds	

# Comparative Insight

**Table 1 Comparison of Topology Integration Strategies**

<b>Criterion</b>	<b>Feature-level</b>	<b>Representation-level</b>	<b>Loss-level</b>
Topology role	External descriptor	Latent representation	Structural constraint
Coupling strength	Weak	Moderate	Strong
Topology gradient	Not propagated	Partial / approximated	Via topology loss
Computational cost	Low	High	Medium–High
Training stability	High	Medium	Low–Medium
Main advantage	Simple and stable	Expressive and flexible	Structure-aware
Main limitation	Representation isolation	High complexity	Optimization sensitivity
Typical use case	Classification / retrieval	Feature learning	Segmentation / reconstruction

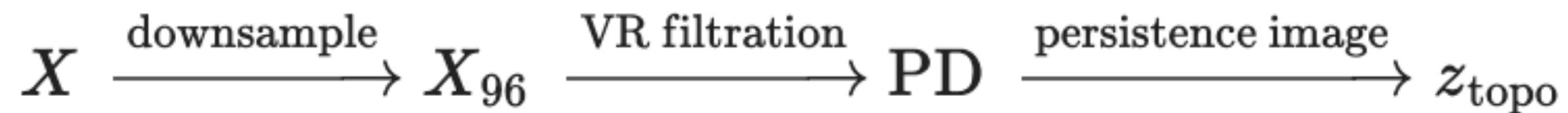
# Experiments

# Experiment Setup

- Dataset: ModelNet40.
- Backbone: PointNet.
- Topology: persistent homology on downsampled point clouds.
- Vectorization: persistence images.
- Compared variants: Baseline, Topo-Only, Topo-Concat All, H0 only, H1 only.

$$X \xrightarrow{\text{downsample}} X_{96} \xrightarrow{\text{VR filtration}} \text{PD} \xrightarrow{\text{persistence image}} z_{\text{topo}}$$

# Topological Feature Representation & Integration



```
# Topological feature generation

# Use only xyz coordinates of the input point cloud
xyz = np.asarray(points, dtype=np.float32)[: , :3]

# Downsample the point cloud to reduce PH computation cost
# Example: 1024 points -> 96 points
if xyz.shape[0] > subsample:
    sample_idx = rng.choice(xyz.shape[0], size=subsample, replace=False)
    xyz = xyz[sample_idx]

# Compute the Vietoris–Rips persistence diagram
# H0 captures connected components, H1 captures loop structures
diagram = _diagram_from_points(xyz, max_edge=max_edge)

# Extract H1 features from the persistence diagram
# lifetime = death - birth
b1_births, b1_life = _finite_pairs_by_dim(diagram, dim=1)

# Normalize birth and persistence values to [0, 1]
b1_births_n = np.clip(b1_births / max_edge, 0.0, 1.0)
b1_persistence_n = np.clip(b1_life / max_edge, 0.0, 1.0)

# Convert the H1 persistence diagram into a fixed-length persistence image
# Output: 64-dimensional topological feature vector
topo_vec = _vectorize_persistence_image(
    b1_births_n,
    b1_persistence_n,
    topo_dim=64,
    pi_sigma=0.08
)
```

# Results

- Topo-Only performs much worse than PointNet.
- Adding all topology does not necessarily help.
- $H_1$ -based features give a small improvement over baseline.
- This suggests topology may give a slightly small improvement, but only as selected auxiliary information, not as a replacement for geometry.

**Table 2 Classification Performance on ModelNet40**

Method	OA	mAcc
Baseline (PointNet)	0.8954	0.8569
Topo-Only (PI)	0.2241	0.1833
Topo-Concat (All)	0.8926	0.8533
Topo-Concat ( $H_0$ )	0.8970	0.8581
Topo-Concat ( $H_1$ )	0.8992	0.8597

OA: overall accuracy over all test samples.  
mAcc: average accuracy across all object classes.

# References

- [1] Ghosh, A., Kulbaka, I., Dahlin, I., & Dutta, A. “TopoRec: Point Cloud Recognition Using Topological Data Analysis.” WACV, 2026.
- [2] Liu, W., et al. “TopoSeg: Topology-aware Segmentation for Point Clouds.” IJCAI, 2022.
- [3] Zhou, C., Dong, Z., & Lin, H. “Learning Persistent Homology of 3D Point Clouds.” Computers & Graphics, 2022.
- [4] Guan, Z., Yan, F., Du, S., Ma, L., & Liu, Q. “TopoDiT-3D: Topology-Aware Diffusion Transformer with Bottleneck Structure for 3D Point Cloud Generation.” BMVC, 2025.
- [5] de Surrel, T., Hensel, F., Carrière, M., Lacombe, T., Ike, Y., Kurihara, H., Glisse, M., & Chazal, F. “RipsNet: A General Architecture for Fast and Robust Estimation of the Persistent Homology of Point Clouds.” TMLR, 2022.
- [6] Jignasu, A., Herron, E., Jiang, Z., Sarkar, S., Hegde, C., Ganapathysubramanian, B., Balu, A., & Krishnamurthy, A. “STITCH: Surface Reconstruction Using Implicit Neural Representations with Topology Constraints and Persistent Homology.” arXiv, 2024.
- [7] Kudeshia, P., Poovvancheri, J., Ghoneim, A., & Chen, D. “A Persistent Homology Design Space for 3D Point Cloud Deep Learning.” arXiv, 2026.
- [8] Pun, C. S., Lee, S. X., & Xia, K. “Persistent-Homology-Based Machine Learning: A Survey and a Comparative Study.” Artificial Intelligence Review, 55, 5169–5213, 2022.

[1] Ghosh, A., Kulbaka, I., Dahlin, I., & Dutta, A.

“TopoRec: Point Cloud Recognition Using Topological Data Analysis.”  
WACV, 2026.

[2] Liu, W., et al.

“TopoSeg: Topology-aware Segmentation for Point Clouds.”  
IJCAI, 2022.

[3] Zhou, C., Dong, Z., & Lin, H.

“Learning Persistent Homology of 3D Point Clouds.”  
Computers & Graphics, 2022.

[4] Guan, Z., Yan, F., Du, S., Ma, L., & Liu, Q.

“TopoDiT-3D: Topology-Aware Diffusion Transformer with Bottleneck Structure for 3D Point Cloud Generation.”  
BMVC, 2025.