Topology- and Gradient-Guided Knowledge Distillation for Point Cloud Semantic Segmentation

Anonymous authors
Paper under double-blind review

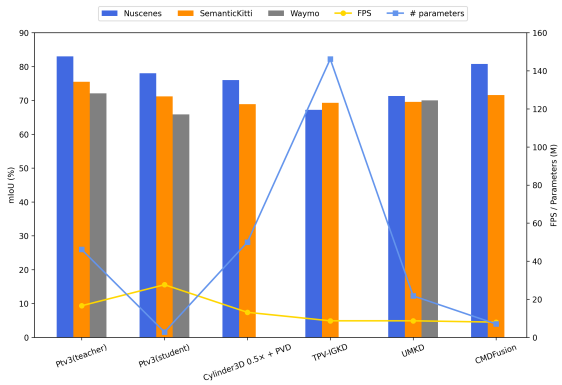
Abstract

Point cloud processing has gained significant attention due to its critical role in applications such as autonomous driving and 3D object recognition. However, deploying high-performance models like Point Transformer V3 in resource-constrained environments remains challenging due to their high computational and memory demands. This work introduces a novel distillation framework that leverages topology-aware representations and gradient-guided knowledge distillation to effectively transfer knowledge from a high-capacity teacher to a lightweight student model. Our approach captures the underlying geometric structures of point clouds while selectively guiding the student model's learning process through gradient-based feature alignment. Experimental results in the Nuscenes, SemanticKITTI, and Waymo datasets demonstrate that the proposed method achieves competitive performance, with an approximately 16× reduction in model size and up to 1.9× decrease in inference time compared to its teacher model. Notably, on NuScenes, our method achieves competitive performance among knowledge distillation techniques trained solely on LiDAR data, surpassing prior knowledge distillation baselines in segmentation performance. Our implementation is available anonymously at `https://anonymous.4open.science/r/PTv3-distill-4E9E`.

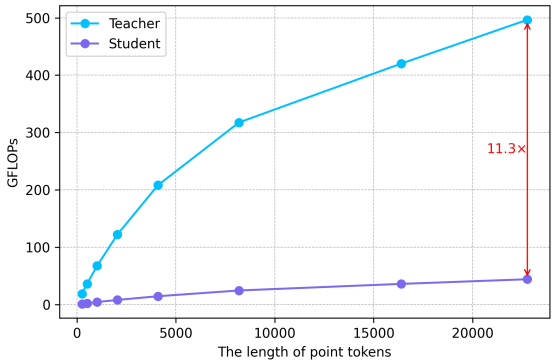
1 Introduction

Point cloud data are a critical representation of 3D geometry and have become essential in a wide range of applications, from autonomous driving and robotic navigation to urban mapping Zhou & Tuzel (2018); Geiger et al. (2012); Gomez-Ojeda et al. (2016); Oh & Watanabe (2002). Recent advances in deep learning have enabled significant progress in point cloud processing, with models such as Point Transformer V3 Wu et al. (2024) setting new benchmarks in accuracy and robustness. Despite the success of models like Point Transformer V3, their high computational demands and memory requirements Golla & Klein (2015); Cao et al. (2019) pose challenges for deployment in resource-constrained environments, such as edge devices or real-time systems. To address this issue, various model compression strategies have been introduced, including methods such as network pruning Han et al. (2016); Liu et al. (2019b); Louizos et al. (2018), quantization Choi et al. (2019); Dong et al. (2022); Nagel et al. (2019), lightweight model architectures Howard et al. (2019); Ma et al. (2018), and knowledge distillation Zhang et al. (2023); Hou et al. (2022a); Zhang et al. (2024).

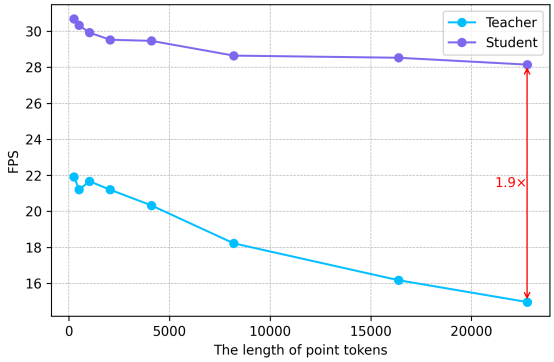
Knowledge distillation is a machine learning technique that aims to transfer knowledge from a large and high capacity model to a smaller and more efficient model Hinton et al. (2014); Romero et al. (2015); Sanh et al. (2019). This approach allows the student model to approximate the performance of the teacher while being computationally less demanding, making it suitable for deployment in resource-constrained environments such as edge devices or mobile platforms. Over the years, knowledge distillation has been effectively applied in various domains, including image recognition Romero et al. (2015); Liu et al. (2019a) and natural language processing Hahn & Choi (2019); He et al. (2021); Rashid et al. (2021), demonstrating its versatility and impact. Recently, several approaches have been introduced to incorporate knowledge distillation into 3D detection tasks using point cloud data Guo et al. (2021b); Zhang et al. (2024). However, these methods



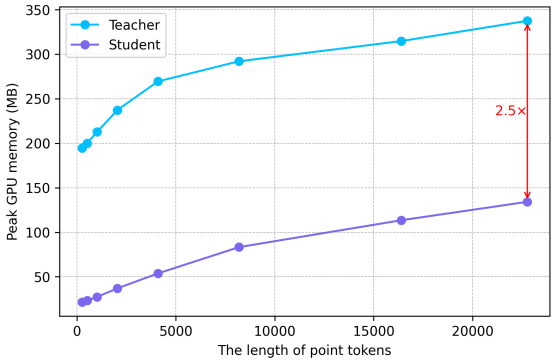
(a) Performance comparison



(b) GFLOPs comparison



(c) FPS comparison



(d) Peak Memory Comparison

Figure 1: Comprehensive comparisons between our proposed method and state-of-the-art knowledge distillation baselines across multiple evaluation metrics Hou et al. (2022c); Li et al. (2024b); Sun et al. (2024); Cen et al. (2024). (a) The radar chart demonstrates that our method achieves consistently better mIoU on three key datasets (NuScenes, SemanticKITTI, and Waymo), along with favorable FPS and memory efficiency. (b)-(d) As the input point token length increases, our approach maintains lower GPU memory usage and FLOPs, while sustaining significantly faster inference speed. (d) Peak GPU memory usage during inference, measured using max memory allocated function in Torch. This metric reflects the highest amount of memory used by the PyTorch tensors by the caching allocator during the inference phase. Notably, this value may differ significantly from the memory reported by the PyTorch Profiler, as it includes temporary allocations used by CUDA kernels.

focus primarily on multimodal student-teacher selection in a multimodal context, often overlooking the unique geometric characteristics of point clouds.

To overcome these deployment hurdles, we propose a framework combining topology-aware representations with gradient-guided distillation. The framework leverages the inherent geometric and structural properties of point clouds to preserve critical topological information during the distillation process. By integrating gradient-based guidance, the proposed approach selectively emphasizes salient geometric features that contribute most significantly to the model's performance, enabling efficient knowledge transfer from a high-capacity teacher model to a lightweight student model. This strategy ensures that the student model retains competitive accuracy while significantly reducing computational and memory requirements, making it suitable for real-time and edge-based applications.

Extensive experiments on the proposed method have been conducted to demonstrate the effectiveness of our approach over previous knowledge distillation methods. Our main contributions can be summarized as follows.

- We propose a novel distillation framework that integrates topology-aware knowledge representation and gradient-guided distillation techniques, addressing the challenges of deploying high-performance point cloud models in resource-constrained environments.
- The framework leverages the unique geometric and structural properties of point clouds, embedding topological information into the distillation process to ensure the preservation of critical features necessary for accurate predictions.
- By incorporating gradient-guided distillation, our method selectively emphasizes salient features, enabling efficient and effective knowledge transfer from the teacher model to the student model.
- Extensive experimental results on popular benchmark datasets, such as Nuscenes reveal that our approach achieves up to a 16× reduction in the number of parameters and a 77.75% reduction in CUDA memory consumption in linear operations and a 2.5× lower in peak CUDA memory usage during inference while maintaining accuracy within 5% of state-of-the-art of non-distilled methods.

While traditional knowledge distillation approaches, such as matching logits or intermediate attention features Wu et al. (2023); Han et al. (2024); Wang et al. (2020) have proven effective for compressing standard transformer architectures, they are often insufficient for 3D point cloud semantic segmentation. Under aggressive compression, lightweight 3D student networks exhibit severe geometric degradation, as (i) the fragmentation of large continuous regions (loss of connectedness) and (ii) the over-smoothing of boundaries for thin or small structures. Standard KD objectives do not explicitly preserve the global geometric continuity and critical structural regions necessary to prevent these issues. Therefore, our framework is designed to be complementary to traditional KD: rather than replacing standard distillation, we explicitly transfer structural topology (inspired by topology-preserving dense prediction methods Hu et al. (2019)) and gradient-guided spatial saliency to resolve these 3D-specific failure modes.

2 Related Works

3D Point Cloud Processing. Representing 3D data via point clouds is central to autonomous driving and robotics. Traditional deep learning approaches fall into three main categories Halperin & Eisl (2025): projection-based methods Chen et al. (2017); Lang et al. (2019); Li et al. (2016) that map points to 2D grids (often sacrificing geometric detail); voxel-based methods Choy et al. (2019); Song et al. (2017); Wang et al. (2017) using 3D sparse convolutions (limited by grid resolution and sparsity); and point-based methods Ma et al. (2022); Qi et al. (2017); Thomas et al. (2019); Zhao et al. (2019) that directly process raw points. While early point-based approaches struggled with local structures, modern transformer architectures Guo et al. (2021a); Robert et al. (2023); Wu et al. (2022b); Yang et al. (2023) have significantly improved long-range dependency modeling.

Furthermore, specialized feature extraction frameworks like MASS Li et al. (2022b) and PillarSegNet Manivasagam et al. (2021) have been developed to optimize dense top-view (BEV) and sparse pillar representations. While these designs excel at their specific representations, our work takes an orthogonal approach focused on knowledge distillation (KD). Specifically, our topology-aware KD operates in the latent feature space, transferring structural geometry from a high-capacity point-based teacher to a lightweight student. Distilling across fundamentally different representations (e.g., point-to-voxel) requires complex spatial alignment modules. By adopting a homogeneous point-to-point setup, we isolate our topological loss from cross-representation confounding variables, leaving cross-modal distillation for future work.

Point Transformer Architecture. Transformers improve point-based methods by leveraging self-attention to capture multi-scale dependencies Guo et al. (2021a); Wu et al. (2022b). Evolving from Point Transformer V1 Zhao et al. (2021) and V2 Wu et al. (2022a), which suffered from high computational costs and kNN bottlenecks—Point Transformer V3 (PTv3) Wu et al. (2024) shifted toward simplicity by serializing point clouds via space-filling curves and serialized patch attention. Despite achieving state-of-the-art results with massive reductions in memory and latency, PTv3's preprocessing overhead and reliance on high-end hardware still limit its deployment in resource-constrained, real-time scenarios.

Knowledge distillation (KD) improves a lightweight student model by transferring knowledge from a pre-trained teacher, offering a pathway to compress heavyweight models like PTv3. Moving beyond early softmax matching Hinton et al. (2015), modern KD aligns intermediate feature layers Liu et al. (2019a); Romero et al. (2015); Tung & Mori (2019) to capture the rich geometric contexts essential for 3D point-cloud data.

Multi-Step Knowledge Distillation. When compressing a high-capacity teacher into a significantly smaller student, the large capacity gap can lead to optimization difficulties or training collapse. In such extreme scenarios, introducing an intermediate Teacher Assistant (TA) model (e.g., TAKD Mirzadeh et al. (2020), AMD Han et al. (2024)) is a standard multi-step solution to smooth knowledge transfer. However, our proposed framework bypasses the need for a TA model for two reasons. First, our student is a scaled-down variant of the same PTv3 backbone family, achieving a strong baseline even without distillation; thus, optimization stability is not the primary bottleneck in this regime. Second, rather than mitigating the capacity gap with intermediate networks, we explicitly target the qualitative information lost during compression. By transferring global topological structures and gradient-guided spatial saliency, our method provides the structural signals necessary to reduce reliance on an intermediate model. Nevertheless, combining our topology-aware distillation with a TA framework represents a promising direction for future work in even more aggressive compression settings.

Topological Distillation leverages topological data analysis (TDA) to transfer global structural features. Methods like TGD Jeon et al. (2024) and TopKD Kim et al. (2024) distill knowledge via persistence images (PI). However, they face scalability challenges due to TDA's computational cost and approximation errors during PD-to-PI conversion, limiting their effectiveness on diverse, noisy point clouds. Unlike these existing topological KD methods designed primarily for 2D images, our approach targets large-scale LiDAR segmentation by employing exact diagram-level matching and custom gradient routing through the Vietoris-Rips complex, avoiding the scaling bottlenecks of vectorization. Furthermore, our framework complements traditional feature distillation (e.g., AD-KD Wu et al. (2023), AMD Han et al. (2024), MiniLM Wang et al. (2020)) by explicitly resolving 3D-specific failure modes, such as structural fragmentation and over-smoothed boundaries that standard KD objectives fail to preserve under strong compression.

Differentiable Topological Layers. Integrating persistent homology end-to-end requires overcoming the non-differentiability of the discrete persistence map. Early approaches decoupled topology from optimization via post-hoc descriptors. More recent methods like Topological Autoencoders Moor et al. (2020), PersLay Carrière et al. (2020), and Differentiable Topology Layers Brüel-Gabrielsson et al. (2020) treat persistence diagrams as differentiable layers. However, they typically rely on soft approximations or kernel-based smoothing. In contrast, our approach leverages the *inverse map theorem* to route gradients directly through the exact critical edges of the Vietoris-Rips complex Poulenard et al. (2018), enabling precise structural alignment without approximation artifacts.

3 Methodology

3.1 Overview of the Framework

Our framework introduces a topology-guided distillation approach designed to transfer both structural complexity and semantic precision from a high-capacity teacher to a lightweight student. As illustrated in Figure 2, the student mimics the teacher through a multi-objective optimization strategy. Beyond standard output replication, we introduce a Topological Distillation mechanism that aligns the underlying geometric manifolds of the feature spaces.

The student model optimizes the following objectives:

- **Targeted Topological Distillation**: We employ a differentiable Vietoris-Rips filtration to extract exact persistence diagrams (H_0 and H_1) from both models. A persistence-weighted Chamfer loss forces the student to replicate significant topological features (clusters and loops) while ignoring noise.

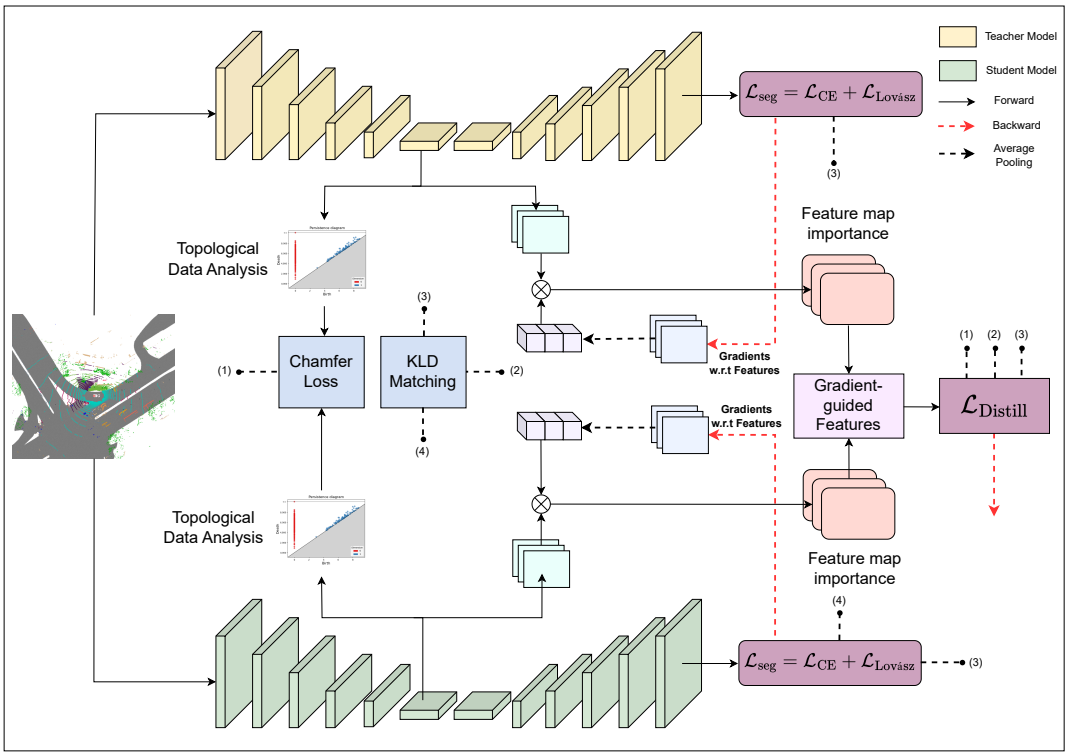


Figure 2: Overview of the proposed framework. The distillation process is driven by (1) **Topological Data Analysis (TDA)**, which aligns the persistence diagrams of feature manifolds using a weighted Chamfer distance; (2) **Logits Distillation** via KLD; and (3) **Gradient-Guided Feature Alignment**. The combination enables the lightweight student to capture global geometry (H_1 loops, H_0 clusters) and local semantics simultaneously.

- **Gradient-Guided Feature Alignment**: We utilize task-specific gradients to identify and align spatially salient feature regions, ensuring the student focuses on semantically critical points.
- **Distribution Matching**: We employ Kullback-Leibler Divergence (KLD) on the soft logits to transfer the teacher's class-wise confidence distributions.

3.2 Manifold Alignment via Differentiable Topology

Standard losses like MSE focus on point-to-point accuracy but ignore the global shape of the data. This often causes lightweight models to learn fractured manifolds—getting the points roughly correct, but breaking the underlying geometric structure.

3.2.1 Differentiable Vietoris-Rips Filtration

The core challenge in integrating Topological Data Analysis (TDA) into deep learning is the discrete nature of the persistence map. Standard algorithms for computing persistent homology rely on combinatorial operations (matrix reduction over finite fields) which are non-differentiable. Integrating Topological Data Analysis (TDA) into deep learning involves computing persistent homology on the latent embeddings. Although the latent space $\mathbb{R}^C$ is continuous, for each input the network produces a finite set of feature vectors (a point cloud in $\mathbb{R}^C$). Persistent homology is explicitly designed to infer topological structure from such finite samples in a metric space, a principle widely utilized in topology-aware deep learning Moor et al. (2020); Carrière et al. (2020); Brüel-Gabrielsson et al. (2020). Furthermore, because persistent homology depends exclusively

on the pairwise distance matrix, it is mathematically well-defined regardless of the high dimensionality. To ensure the topological signals remain informative and are not overwhelmed by high-dimensional distance noise, our framework relies on three design choices: (1) we apply persistent homology to learned embeddings whose geometry is inherently shaped by the segmentation task; (2) we utilize Topology-Preserving Sampling with a stability argument (Appendix A.2) Cohen-Steiner et al. (2007); and (3) we suppress near-diagonal artifacts using persistence weighting (Section 3.2.2), which reduces the impact of low-persistence noise.

The core technical challenge in this integration lies in optimization and differentiation. The standard computation of persistent homology involves discrete algorithmic steps, such as sorting simplices and matrix reduction over finite fields, which are inherently non-differentiable. To overcome this, we exploit the fact that for Vietoris–Rips filtrations the birth/death times are piecewise-smooth functions of the pairwise distances: each birth/death value equals the filtration value of a paired creator/destroyer simplex, whose filtration value is the maximum edge length within that simplex.

Problem. Let $f_\theta : \mathcal{X} \to \mathcal{H}$ be the encoder parameterized by θ. For an input X, we obtain feature embeddings $H = f_\theta(X) = \{h_1, \dots, h_N\} \subset \mathbb{R}^C$. The persistence map $\Psi : H \to \mathcal{D}$ involves sorting edge lengths and reducing a boundary matrix over a finite field $\mathbb{Z}_2$, operations which usually have zero gradients almost everywhere.

Solution. Although persistent homology is computed by discrete operations (sorting simplices and reducing a boundary matrix over $\mathbb{Z}_2$), the resulting diagram coordinates are piecewise-smooth functions of the underlying edge lengths: for a fixed pairing and a fixed set of maximizers defining each simplex filtration value, the birth/death times vary smoothly with the distances. Non-differentiabilities occur only at degenerate configurations, when (i) the pairing changes, or (ii) the filtration value of a critical simplex has multiple maximizing edges. Following Solomon et al. (2022); Gameiro et al. (2016); Poulenard et al. (2018), we therefore treat the Ripser Bauer (2021) as a forward-only black box and implement a custom autograd rule, which is in the forward pass we compute persistence pairs and record the corresponding birth/death critical simplices, and in the backward pass we route gradients from each pair (b_k, d_k) to the critical edge(s) that realize these filtration values in the distance matrix. At degenerate configurations we use an explicit subgradient selection rule, which yields a valid gradient for stochastic optimization. In addition to the non-differentiabilities of the persistence map (pairing changes and max-edge ties), our loss uses nearest-neighbor Chamfer minima and a diagonal fallback via nested $\min(\cdot)$ operators. These are piecewise-smooth and admit valid Clarke subgradients; in implementation we backpropagate through the selected nearest neighbor (and through the diagonal term when it is active), and at ties we use a deterministic uniform averaging. This yields a valid subgradient method in stochastic optimization. Solution is described in Algorithm 1.

Theorem 1 (Gradient Backpropagation via Critical-Edge Routing)**.** *Let $\mathcal{H} = \{h_1, \dots, h_N\} \subset \mathbb{R}^C$ be the set of feature embeddings for N points and let $\mathbf{D} \in \mathbb{R}^{N \times N}$ is pairwise distance matrix where each entry is the Euclidean distance $\mathbf{D}_{ij} = \|h_i - h_j\|_2$. We construct a Vietoris-Rips filtration on these points. In this process, any simplex σ (a geometric shape like a triangle or tetrahedron) appears at a specific "time" called its filtration value $v(\sigma)$. This value is determined by the longest edge in the simplex:*

$$v(\sigma) \;=\; \max_{\{i,j\} \subset \sigma} \mathbf{D}_{ij}.$$

*Let a persistent homology (PH) backend (Risper Bauer (2021)) processes this filtration and returns the multiset of topological features $\mathcal{P} = \{(b_k, d_k, \dim(k))\}_{k=1}^{K}$. Each feature k is described by: **(i)** A persistence pair (b_k, d_k) representing its Birth and Death times; **(ii)** a creator simplex $\sigma_b(k)$ is the specific shape that created the feature; **(iii)** a destroyer simplex $\sigma_d(k)$ is the specific shape that destroyed (filled in) the feature.*

To make this differentiable, we identify exactly which edges determined the birth and death times. We define the Maximizer Edge Sets as:

$$E_b(k) \;:=\; \arg\max_{\{i,j\} \subset \sigma_b(k)} \mathbf{D}_{ij}, \qquad E_d(k) \;:=\; \arg\max_{\{u,v\} \subset \sigma_d(k)} \mathbf{D}_{uv}.$$

(For H_0 births, $b_k = 0$ and we take $E_b(k) = \emptyset$.)

When the filtration value of a simplex is defined by a max over its edge lengths, the gradient routes to the (edge) argmax. If the argmax is unique, the routing is unique. If several edges tie for the maximum,

the function is non-smooth at that point and any Clarke subgradient is valid. Concretely, let $E_b(k)$ and $E_d(k)$ be the sets of maximizer edges for the birth and death simplices of pair k. We choose coefficients $\{\alpha_{ij}^{(k)}\}_{\{i,j\}\in E_b(k)}$ and $\{\beta_{uv}^{(k)}\}_{\{u,v\}\in E_d(k)}$ with $\alpha_{ij}^{(k)} \geq 0$ and $\sum_{\{i,j\}\in E_b(k)} \alpha_{ij}^{(k)} = 1$ (and similarly for β), and distribute the gradient across the tied edges accordingly. Common choice is uniform averaging over tied maximizers ($\alpha_{ij}^{(k)} = 1/|E_b(k)|$).

Let $\mathcal{L}$ be any loss differentiable with respect to the diagram coordinates $\{(b_k, d_k)\}$. Assume we are away from pairing-change degeneracies (i.e., the creator/destroyer simplices $\sigma_b(k), \sigma_d(k)$ are locally constant); at tie points among maximizer edges, interpret derivatives in the Clarke subgradient sense. Then a valid gradient of $\mathcal{L}$ with respect to $\mathbf{D}_{ij}$ (for $i < j$) is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{D}_{ij}} = \sum_{k=1}^{K} \left(\frac{\partial \mathcal{L}}{\partial b_k} \alpha_{ij}^{(k)} \mathbb{1}_{\{\{i,j\}\in E_b(k)\}} + \frac{\partial \mathcal{L}}{\partial d_k} \beta_{ij}^{(k)} \mathbb{1}_{\{\{i,j\}\in E_d(k)\}} \right), \qquad i < j, \tag{1}$$

Where:

- $\frac{\partial \mathcal{L}}{\partial b_k}$ and $\frac{\partial \mathcal{L}}{\partial d_k}$ are the gradients from the Persistence-Weighted Chamfer Loss (PWCD) (Algorithm 3).
- $\mathbf{1}_{\{i,j\}\in E_b(k)}$ is an indicator function. It is 1 if edge $\{i, j\}$ is the Critical Creator Edge for feature k, and 0 otherwise.
- Symmetrically $\frac{\partial \mathcal{L}}{\partial \mathbf{D}_{ji}} = \frac{\partial \mathcal{L}}{\partial \mathbf{D}_{ij}}$.

Moreover, let $D_{ij} = \|h_i - h_j\|_2$ and let $g_{ij} := \partial L / \partial D_{ij}$ be accumulated for undirected edges $\{i, j\}$ with $i < j$ (as in Equation 1). For any stabilizer $\varepsilon_{\text{dist}} > 0$, a valid gradient with respect to embeddings is obtained by summing over undirected edges: for each $i < j$,

$$\frac{\partial L}{\partial h_i} \mathrel{+}= g_{ij} \frac{h_i - h_j}{D_{ij} + \varepsilon_{\text{dist}}}, \qquad \frac{\partial L}{\partial h_j} \mathrel{-}= g_{ij} \frac{h_i - h_j}{D_{ij} + \varepsilon_{\text{dist}}}.$$

In practice, Algorithm 4 computes $(\partial L/\partial b_k, \partial L/\partial d_k)$ for PWCD, and Algorithm 1 implements Equation 1 by routing these diagram gradients to the critical edges of the paired simplices, thereby obtaining $\partial L/\partial D$ and finally $\partial L/\partial F$.

3.2.2 Persistence-Weighted Chamfer Distance

While the Differentiable Rips Filtration successfully extracts persistence diagrams, comparing them during training presents a unique optimization challenge. Standard topological metrics, such as the Wasserstein distance, rely on optimal transport (OT). Although exact OT has cubic computational complexity, entropic regularization (e.g., Sinkhorn) Cuturi (2013) and other advanced approximations Wang et al. (2023); Zhou et al. (2022); Izquierdo & Civera (2024) make transport-based metrics practically efficient.

However, our choice of a Chamfer-style objective over Sinkhorn OT is driven primarily by gradient behavior and robustness to topological noise. In distillation, the teacher and student persistence diagrams (PD) possess dynamic, unequal cardinalities. While Unbalanced or Partial OT (UOT/POT) Chizat et al. (2018); Fatras et al. (2021) can mathematically handle unequal sets, they introduce hyperparameters (e.g., entropic regularization ϵ and mass-divergence penalties) that are notoriously difficult to tune when PD noise fluctuates across training batches. Crucially, Sinkhorn OT produces a dense, global soft matching. When the student PD contains heavy near-diagonal noise early in training, this soft matching distributes gradients broadly across many low-value matches, pulling features in sub-optimal directions.

In contrast, the Chamfer distance provides a computationally efficient $O(n^2)$-time-complexity alternative that resolves these gradient stability issues. A Chamfer formulation with a diagonal fallback yields localized, direct gradients. It naturally tolerates unequal set sizes and allows unmatched or noisy student points to be safely pushed toward the diagonal (vanishing persistence) without forcing global mass conservation (as empirically validated in Appendix D.5).

Despite these advantages, the standard Chamfer distance has a fatal flaw in knowledge distillation: it treats all target features equally. A teacher's PD often contains hundreds of low-persistence features (points near the diagonal) caused by minor fluctuations or sampling noise in the latent space. In a standard Chamfer calculation, the student would be forced to allocate representational capacity to replicate this stochastic noise, leading to overfitting of the teacher's artifacts rather than its semantic geometry.

To solve this, we introduce a *Persistence-Weighted Chamfer Distance*. This metric effectively functions as a soft attention mechanism. It forces the student to prioritize robust topological features (such as prominent loops and clusters) while safely permitting it to ignore the teacher's stochastic noise.

Definition. Let $P^T = \{p_i\}_{i=1}^M$ and $P^S = \{q_j\}_{j=1}^N$ be the multisets of finite persistence pairs in a fixed homology dimension, where $p = (b, d) \in \mathbb{R}^2$. Define persistence $\text{pers}(p) = d - b$, teacher weights $w(p) = \log(1 + \text{pers}(p))$, and the stabilized normalizer $\bar{w}_T = \sum_{p \in D^T} w(p) + \epsilon$. Let the squared distance to the diagonal be $\text{dist}_\Delta^2(b, d) = (d - b)^2/2$.

The Persistence-Weighted Chamfer Distance $\mathcal{L}_{CD}^w$ is defined as:

$$\mathcal{L}_{\text{topo}}(P^T, P^S) = \frac{1}{\bar{w}_T} \sum_{p \in P^T} w(p) \min\Big(\min_{q \in P_S} \|p - q\|_2^2,\ \text{dist}_\Delta^2(p) \Big) \tag{2}$$

$$+ \frac{1}{|P^S|} \sum_{q \in P^S} \min\Big(\min_{p \in D_T} \|q - p\|_2^2,\ \text{dist}_\Delta^2(q) \Big), \tag{3}$$

with the standard conventions that if $D^S = \emptyset$ then $\min_{q \in D_S}(\cdot) = +\infty$ and the precision term is 0, and if $D^T = \emptyset$ then the recall term is 0. This yields a well-defined objective for all (D^T, D^S) and ensures that unmatched student features are penalized by moving them toward the diagonal (vanishing persistence).

Theoretical Justification (See Appendix A.3).

3.3 Saliency Aware Feature Alignment

Standard feature distillation treats all spatial locations equally, often causing the student to waste representational capacity on task irrelevant background regions rather than semantic boundaries. To address this, we introduce a Saliency Aware Feature Alignment mechanism. We interpret the gradient of the task loss $\mathcal{L}_{\text{seg}}$ with respect to feature maps as a proxy for semantic importance, prioritizing regions where model predictions are most sensitive to feature perturbations.

3.3.1 Discriminative Channel Weighting

Let $F^l \in \mathbb{R}^{N \times C}$ denote the feature activations at layer l. We define the task sensitivity of the k-th channel as the global magnitude of its gradient with respect to the segmentation loss.

Although the teacher model's parameters are frozen during distillation, we explicitly compute the gradients of the task loss $\mathcal{L}_{\text{seg}}$ with respect to the *feature activations* (hidden states) to quantify spatial sensitivity. The channel-wise importance weight ω_l^k is computed as:

$$\omega_k^l = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{\partial \mathcal{L}_{\text{seg}}(Y, Y_{GT})}{\partial F_{i,k}^l} \right| \tag{4}$$

where Y_{GT} denotes the ground truth labels. This step identifies which feature channels are causally responsible for the correct classification, independent of the model's weight updates.

3.3.2 Spatial Attention Transfer

We construct a spatial saliency map $M^l \in \mathbb{R}^N$ by aggregating the weighted feature responses across the channel dimension. This aggregation collapses the feature descriptors into a scalar attention score for each

point, rendering the representation dimension agnostic:

$$M_i^l = \sum_{k=1}^{C} \left| \omega_k^l F_{i,k}^l \right| . \tag{5}$$

Since the teacher and student possess different channel dimensions ($C_T \neq C_S$), the raw magnitudes of their attention maps are not directly comparable. To ensure scale invariance, we apply min max normalization to generate the final attention descriptor $\hat{M}^l$:

$$\hat{M}^l = \frac{M^l - \min(M^l)}{\max(M^l) - \min(M^l) + \epsilon}. \tag{6}$$

The alignment objective minimizes the L_1 distance between the normalized attention maps of the teacher ($\hat{M}_T$) and student ($\hat{M}_S$):

$$\mathcal{L}_{\text{grad}} = \frac{1}{N} \sum_{l=1}^{L} \left\| \hat{M}_T^l - \hat{M}_S^l \right\|_1 . \tag{7}$$

By optimizing $\mathcal{L}_{\text{grad}}$, we force the student to replicate the teacher's *spatial focus*, ensuring that limited capacity is allocated to the most discriminative points in the cloud regardless of the underlying feature dimension.

3.4 Optimization Objective

To complement the geometric and spatial alignment provided by the topological and saliency modules, we incorporate a Semantic Distillation term using Kullback-Leibler Divergence (KLD) Kullback (1951). While $\mathcal{L}_{\text{topo}}$ and $\mathcal{L}_{\text{grad}}$ align the encoder's latent manifold, the KLD loss is applied to the temperature-scaled output logits of the classifier. This guides the student to mimic not only the teacher's hard predictions but also the underlying class probability distribution, capturing inter-class correlations.

The total training objective integrates the standard segmentation task with the three distillation constraints:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seg}} + \lambda_{\text{topo}} \mathcal{L}_{\text{topo}} + \lambda_{\text{grad}} \mathcal{L}_{\text{grad}} + \lambda_{\text{KLD}} \mathcal{L}_{\text{KLD}}, \tag{8}$$

where:

- $\mathcal{L}_{\text{seg}}$ is the standard Cross-Entropy loss for point-wise classification using ground truth labels.
- $\mathcal{L}_{\text{topo}}$ enforces the preservation of high-level topological structures (clusters and loops) in the latent space.
- $\mathcal{L}_{\text{grad}}$ ensures the student focuses on the same discriminative spatial regions as the teacher.
- $\mathcal{L}_{\text{KLD}}$ minimizes the KL divergence between the teacher's and student's softened logits, transferring semantic confidence.

The hyperparameters λ_{topo}, λ_{grad}, and λ_{KLD} balance the contribution of each knowledge stream.

4 Experiments and Results

To assess our topology-aware distillation framework, we performed experiments on three prominent autonomous driving datasets: SemanticKITTI Behley et al. (2019), Waymo Open Dataset Sun et al. (2020), and NuScenes Caesar et al. (2020). These datasets offer large-scale, real-world point-cloud sequences, ideal for benchmarking point-cloud processing techniques. We provide detailed descriptions of datasets, training protocols, and evaluation procedures in the Appendix C.

4.1 Experimental Results

Performance on nuScenes Dataset. Table 1 compares our method against state-of-the-art frameworks on the nuScenes test set. While recent cross-modal methods integrating camera data (LC), such as TPV-IGKD Li et al. (2024a) (81.3% mIoU) and U2MKD Liu & et al. (2024) (81.2% mIoU), establish the current upper bound, our proposed LiDAR-only student achieves a highly competitive 78.2% mIoU. This significantly outperforms established baselines including Cylinder3D Zhu et al. (2021) (76.1%), AMVNet Liong et al. (2020) (76.1%), and 2DPASS Yan et al. (2022) (76.2%). The advantages of our approach are further highlighted by its class-wise performance. Notably, our student achieves the highest accuracy among all compared models, including multi-modal fusion ones in the car (94.3%) category and demonstrates superior performance in the terrain (78.8%) category. This performance directly validates our methodological design against the common failure modes of lightweight networks: structural fragmentation and boundary over-smoothing. For large geometric manifolds like terrain, our topological supervision ($\mathcal{L}_{\text{topo}}$) optimizes H_0 (connected components) to preserve the teacher's dominant connectedness patterns. As shown by the qualitative visualizations in Appendix D.2 (Figure 5), this effectively mitigates the feature over-smoothing that causes localized fragmentation, restoring scattered predictions into continuous semantic regions. Furthermore, the gradient-guided alignment ($\mathcal{L}_{\text{grad}}$) aligns task-loss gradient saliency with respect to hidden activations to highlight sensitive boundary regions.

Methods	**mIoU**	barrier	bicycle	bus	car	construction	motorcycle	pedestrian	traffic-cone	trailer	truck	driveable	other	sidewalk	terrain	manmade	vegetation
RangeNet++ Milioto et al. (2019)	65.5	66.0	21.3	77.2	80.9	30.2	66.8	69.6	52.1	54.2	72.3	94.1	66.6	63.5	70.1	83.1	79.8
PolarNet Zhang et al. (2020b)	71.0	74.7	28.2	85.3	90.9	35.1	77.5	71.3	58.8	57.4	76.1	96.5	71.1	74.7	74.0	87.3	85.7
SalsaNext Cortinhal et al. (2020b)	72.2	74.8	34.1	85.9	88.4	42.2	72.4	72.2	63.1	61.3	76.5	96.0	70.8	71.2	71.5	86.7	84.4
Cylinder3D Zhu et al. (2021)	76.1	76.4	40.3	91.2	93.8	51.3	78.0	78.9	64.9	62.1	84.4	96.8	71.6	76.4	75.4	90.5	87.4
C3D_0.5× + KA Hou et al. (2022b)	73.9	74.2	36.3	88.5	87.6	47.1	76.9	78.3	63.5	57.6	83.4	94.9	70.3	73.8	73.2	88.4	86.3
AMVNet Liong et al. (2020)	76.1	79.8	32.4	87.4	90.4	62.5	81.9	75.3	72.3	83.5	65.1	97.4	67.0	78.8	74.6	90.8	87.9
2DPASS Yan et al. (2022)	76.2	75.3	43.5	95.3	91.2	54.5	78.9	78.2	62.1	70.0	84.2	96.3	73.2	74.2	74.9	89.8	85.9
SDSeg3D Li et al. (2022a)	77.7	77.5	49.4	93.9	92.5	54.9	86.7	80.1	67.8	65.7	**86.0**	96.4	74.0	74.9	74.5	86.0	82.8
RPVNet Xu et al. (2021)	77.6	78.2	43.4	92.7	93.2	49.0	85.7	80.6	66.9	69.4	80.5	96.9	73.5	75.9	76.0	90.6	88.9
GFNet Qiu et al. (2022)	76.1	81.1	31.6	76.0	90.5	60.2	80.7	75.3	71.8	82.5	65.1	**97.8**	**80.4**	80.4	76.2	**91.8**	88.9
SVASeg Zhao et al. (2022)	74.7	74.1	44.5	88.4	86.6	48.2	72.4	72.3	61.3	57.5	75.7	96.3	70.7	74.7	74.6	87.3	86.9
U2MKD Liu & et al. (2024) (LC)	81.2	**85.9**	43.4	93.1	90.1	72.5	86.3	81.6	**78.5**	85.7	76.5	97.6	69.8	80.3	77.0	**91.8**	89.5
KT-Weakly Wang et al. (2025)(LC)	80.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MV3Dseg Sun et al. (2025)	81.1	79.7	59.2	**96.0**	92.6	60.3	88.4	83.6	70.8	80.2	81.8	96.7	74.4	79.6	76.1	90.0	88.2
TPV-IGKD Li et al. (2024a) (LC)	81.3	85.4	43.2	92.9	93.2	**75.7**	77.4	83.4	77.2	**86.8**	77.4	97.7	71.4	**81.3**	77.4	91.7	89.0
Teacher (PTv3)	**83.0**	83.5	**59.9**	**96.0**	**95.0**	58.3	**91.0**	**86.4**	77.7	79.2	85.1	97.2	77.0	80.9	**80.0**	91.2	**90.1**
Student PTv3 (w/o KD)	76.1	76.1	46.7	89.9	92.2	40.4	83.9	78.3	63.2	68.2	81.8	96.3	72.8	73.6	75.4	89.7	88.8
Student PTv3 (Ours)	78.2	79.1	48.3	92.9	94.3	41.3	85.7	82.9	62.4	70.2	80.3	96.7	76.3	74.2	78.8	90.0	89.3

Table 1: Comparison with state-of-the-art semantic segmentation methods on the nuScenes test set. Methods denoted with (LC) utilize LiDAR-camera fusion, whereas all others, including our proposed framework, are strictly LiDAR-only. In our setup, both the teacher and student models are based on the Point Transformer V3 (PTv3) architecture. The baseline student (w/o KD) is trained using standard Cross-Entropy (CE) loss, while our method (Ours) employs the full distillation objective detailed in Section 3.4. With competitive performance, our model requires fewer parameters, achieves higher inference speed (FPS), and consumes less memory (see Tables 3, 8, and 9 for detailed efficiency analyses).

Performance on SemanticKITTI. Table 2 details the semantic segmentation performance on the SemanticKITTI test set. Our distilled student model achieves a highly competitive 73.1% mIoU, effectively approaching the state-of-the-art MV3Dseg Sun et al. (2025) (73.2%) while using a significantly lighter architecture. Crucially, our method outperforms recent complex baselines such as KT-Weakly Wang et al. (2025) (71.4%) and C3D+SCPNet Xia et al. (2023) (71.5%). The efficacy of our distillation strategy is evidenced by a substantial 1.6% improvement over the baseline student (71.5%). Theoretical advantages of

our Topology-Aware Distillation become evident in the class-wise breakdown. While multi-view methods like MV3Dseg excel in texture-heavy classes (e.g., vegetation), our approach dominates in geometrically complex and structural categories. Under aggressive compression, fine structures are typically washed out. Our persistence-based loss explicitly counters this: optimizing H_1 (cycles and loops) forces the student to preserve non-trivial structural patterns and thin, disconnected topological features that Scene Completion methods Song et al. (2017) tend to over-smooth to fill occlusions. Importantly, our persistence weighting mechanism isolates critical simplices, ensuring the distillation focuses strictly on these dominant, stable structures rather than chasing near-diagonal topological noise or stochastic artifacts.

Methods	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic
SalsaNext Cortinhal et al. (2020b)	59.5	91.9	48.3	38.6	38.9	31.9	60.2	59.0	19.4	91.7	63.7	75.8	29.1	90.2	64.2	81.8	63.6	66.5	54.3	47.4
KPConv Thomas et al. (2019)	58.8	96.0	32.0	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	31.6	**95.0**	64.2	84.8	69.2	69.1	56.4	47.4
FusionNet Zhang et al. (2020a)	61.3	95.3	47.5	37.7	41.8	34.5	59.5	56.8	11.9	91.8	68.7	77.1	30.5	90.5	69.4	84.5	69.8	68.5	60.4	46.2
KPRNet Kochanov et al. (2020)	63.1	95.5	54.1	47.9	23.6	42.6	65.9	65.0	16.5	93.2	**73.9**	80.6	30.2	91.7	64.8	85.7	69.8	71.2	58.7	64.1
TORNADONet Gerdzhev et al. (2021)	63.1	94.2	51.2	48.1	40.0	38.2	63.6	60.1	34.9	89.7	66.7	74.5	28.7	91.3	65.8	85.6	71.5	70.1	58.0	49.2
SPVNAS Tang et al. (2020)	66.4	97.3	51.5	50.8	59.8	58.8	65.7	62.5	43.7	90.2	67.6	75.2	16.9	91.3	65.9	86.1	73.4	71.0	64.6	66.9
Cylinder3D Zhu et al. (2021)	68.9	97.1	67.6	50.8	50.8	58.5	73.7	69.2	48.0	92.2	65.0	77.0	32.3	90.7	66.5	85.6	72.5	69.8	62.4	66.2
PolarNet+M2S-KD Qiu et al. (2023)	58.0	93.5	45.9	36.3	27.6	34.9	55.0	51.4	15.8	91.1	64.7	73.8	26.1	92.5	67.0	84.6	63.4	67.4	50.7	59.5
C3D+M2S-KD Qiu et al. (2023)	65.6	96.4	60.8	54.8	42.8	51.2	69.1	67.8	34.8	92.2	66.5	76.7	30.4	91.1	65.7	85.5	69.8	68.6	60.7	61.0
C3D+SCPNet Xia et al. (2023)	71.5	97.5	60.9	56.3	58.6	65.9	70.7	71.8	58.7	93.6	72.1	80.9	36.2	93.3	72.1	86.2	74.1	71.6	66.7	71.8
U2MKD Liu & et al. (2024) (LC)	69.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
TPV-IGKD Li et al. (2024a)	69.3	96.7	63.3	65.8	54.6	63.1	76.5	**80.7**	41.6	90.0	60.3	75.4	29.1	91.1	66.3	85.9	74.2	70.3	63.8	69.1
KT-Weakly Wang et al. (2025)(LC)	71.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MV3Dseg Sun et al. (2025)	73.2	**97.6**	69.8	**76.1**	**65.4**	61.3	79.1	80.0	**71.1**	94.2	67.6	78.6	37.5	91.8	65.9	85.7	72.8	73.7	63.5	60.0
Teacher (PTv3)	**75.5**	97.5	**74.2**	62.5	60.1	**67.8**	**80.1**	76.5	60.2	**96.2**	72.1	**82.4**	**47.5**	93.2	**74.1**	**89.5**	**78.2**	**76.1**	**72.1**	**75.9**
Student PTv3 (w/o KD)	71.5	96.0	71.0	56.5	56.5	62.2	77.2	72.8	56.8	95.5	68.7	80.5	36.4	94.0	70.1	89.0	76.0	73.4	66.1	69.8
Student PTv3 (Ours)	73.1	96.8	72.6	58.8	58.8	65.1	77.6	73.9	56.5	92.8	70.5	80.3	43.6	91.6	71.7	87.4	76.6	74.4	68.3	71.6

Table 2: Comparison with state-of-the-art semantic segmentation methods on the SemanticKITTI test set. Methods denoted with (LC) utilize LiDAR-camera fusion, whereas all others, including our proposed framework, are strictly LiDAR-only. In our setup, both the teacher and student models are based on the Point Transformer V3 (PTv3) architecture. The baseline student (w/o KD) is trained using standard Cross-Entropy (CE) loss, while our method (Ours) employs the full distillation objective detailed in Section 3.4. With competitive performance, our model requires fewer parameters, achieves higher inference speed (FPS), and consumes less memory (see Tables 3, 8, and 9 for detailed efficiency analyses).

We further evaluate our framework on the Waymo Open Dataset, as summarized in Table 4. Among multi-modal methods utilizing both LiDAR and camera (LC), MSeg3D Li et al. (2023) and UMKD (B) Sun et al. (2024) demonstrate strong performance with validation mIoU scores of 69.6 and 71.1, respectively. Within the LiDAR-only (L) category, LidarMultiNet Ye et al. (2022) achieves a leading test mIoU of 71.1; however, it should be noted that this approach relies on 3D bounding boxes as an additional supervision signal during training. Our teacher model sets a high performance ceiling for LiDAR-only methods with a test mIoU of 71.3. Notably, our distilled student model achieves a test mIoU of 69.5, making it competitive with several heavier architectures and even approaching the performance of multi-modal methods. This performance is particularly significant given the student's efficiency; by explicitly targeting the global-structure and boundary-saliency problems inherent to strong compression, our model maintains competitive generalization while offering a substantial reduction in parameter count, as shown in Table 3.

4.2 Effectiveness of Individual Loss Components

Table 5 demonstrates the effectiveness of each proposed loss component. To isolate the impact of our contributions, our baseline model already incorporates standard traditional KD, combining the segmentation task loss with KL-divergence logit distillation. This traditional KD baseline achieves reasonable performance (76.9% on nuScenes, 71.8% on SemanticKITTI, and 67.9% on Waymo). However, because standard KD does not explicitly penalize structural fragmentation or boundary over-smoothing, its performance ceiling is

Method	Params (M)	FPS
RangeNet++ Milioto et al. (2019)	50.0	12.5
PolarNet Zhang et al. (2020b)	45.0	16.7
SalsaNext Cortinhal et al. (2020b)	6.7	23.8
Cylinder3D Zhu et al. (2021)	53.0	12.0
KPConv Thomas et al. (2019)	15.0	12.0
SPVNAS Tang et al. (2020)	1.0	16.0
KT-weakly Wang et al. (2025)	-	25.0
MV3DSeg Sun et al. (2025)	45.6	22.7
U2MKD Liu & et al. (2024)	55.8	10.64
TPV-IGKD Li et al. (2024a)	146.2	8.7
GFNet Qiu et al. (2022)	87.6	-
KPRNet Kochanov et al. (2020)	213.2	-
Teacher (PTv3)	46.16	16.61
Student PTv3 (Ours)	**2.78**	**27.64**

Table 3: Comparison of model size (in M parameters) and inference speed (FPS) on NuScenes.

Method	Input	mIoU (test / val)
MSeg3D Li et al. (2023)	LC	70.5 / 69.6
UMKD (B) Sun et al. (2024)	LC	70.0 / 71.1
PMF Zhuang et al. (2021)	LC	– / 58.2
SalsaNext Cortinhal et al. (2020a)	L	55.8 / –
Realsurf Sun et al. (2024)	L	67.6 / –
SPVCNN++ Tang et al. (2020)	L	67.7 / –
VueNet3D Sun et al. (2024)	L	68.6 / –
SphereFormer Lai et al. (2023)	L	– / 69.9
LidarMultiNet Ye et al. (2022)	L	71.1 / 69.9
Student PTv3 (w/o KD)	L	67.2 / 66.5
Student PTv3 (Ours)	L	**69.5** / **68.7**
Teacher (PTv3)	L	71.3 / 69.8

Table 4: Semantic segmentation mIoU on Waymo Open Dataset. Inputs: LiDAR (L), Camera (C).

limited under strong compression. The addition of topology ($\mathcal{L}_{\text{topo}}$) and gradient-guided alignment ($\mathcal{L}_{\text{grad}}$) consistently improves performance across all datasets over standard KD alone. This provides strong empirical evidence that while traditional KD is a necessary foundation, explicitly transferring geometric and boundary information is essential for optimal 3D point cloud compression.

$\mathcal{L}_{\text{KLD}}$	$\mathcal{L}_{\text{seg}}$	$\mathcal{L}_{\text{topo}}$	$\mathcal{L}_{\text{grad}}$	S.KITTI	Waymo	nuScenes
✓	✓			71.8	67.9	76.9
✓	✓	✓		72.6	68.7	77.6
✓	✓		✓	72.3	68.2	77.2
✓	✓	✓	✓	**73.1**	**69.5**	**78.2**

Table 5: Influence of each loss component on the final performance (mIoU %).

5 Conclusion and Future Work

We presented a novel distillation framework that bridges the gap between high-capacity 3D perception models and resource-constrained deployment environments. By integrating Topological Distillation with Saliency-Aware Feature Alignment, our method enables a lightweight student model to inherit not only the semantic precision of a teacher but also its underlying geometric manifold structure.

Extensive experiments on nuScenes, SemanticKITTI, and Waymo demonstrate that our approach achieves competitive performance among LiDAR-based distillation methods. Specifically, our student model retains competitive segmentation accuracy while delivering a 16× reduction in parameters and up to 1.9× acceleration across varying token lengths; 1.6× on the standard nuScenes setup. Crucially, our qualitative analysis reveals that enforcing topological consistency effectively mitigates the fragmentation common in lightweight networks, ensuring robust performance on complex, large-scale structures.

Future Work. Although our topology-preserving subsampling substantially reduces the cost of persistent-homology computation during training, a natural next step is to replace heuristic selection with learnable, task-adaptive topological token sampling and amortized PH computation, enabling larger filtrations and higher-order homology under the same budget. A second direction is extending topology- and gradient-guided distillation to spatio-temporal perception and multi-parameter topology, enforcing consistency across multi-sweep sequences and across geometric scales. Finally, we will study topology-aware distillation in cross-modal and open-world settings, distilling structural priors from camera/BEV and vision(-language) foundation models into deployment-oriented backbones (e.g., hardware-optimized attention or linear-time state-space models) to improve long-tail generalization and robustness under missing or misaligned modalities.

References

Ulrich Bauer. Ripser: efficient computation of Vietoris-Rips persistence barcodes. *J. Appl. Comput. Topol.*, 5 (3):391–423, 2021. ISSN 2367-1726. doi: 10.1007/s41468-021-00071-5. URL https://doi.org/10.1007/s41468-021-00071-5.

Jens Behley, Martin Garbade, Andres Milioto, Jonas Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 9297–9307, 2019.

Rickard Brüel-Gabrielsson, Bradley J Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1553–1563, 2020.

Holger Caesar, Alex Bankiti, Alexander H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11621–11631, 2020.

Chao Cao, Marius Preda, and Titus Zaharia. 3d point cloud compression: A survey. In *Proceedings of the 24th International Conference on 3D Web Technology*, Web3D '19, pp. 1–9, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367981. doi: 10.1145/3329714.3338130. URL https://doi.org/10.1145/3329714.3338130.

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

Jun Cen, Shiwei Zhang, Yixuan Pei, Kun Li, Hang Zheng, Maochun Luo, Yingya Zhang, and Qifeng Chen. Cmdfusion: Bidirectional fusion network with cross-modality knowledge distillation for lidar semantic segmentation. *IEEE Robotics and Automation Letters*, 9(1):771–778, 2024. doi: 10.1109/LRA.2023.3335771.

Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018.

Jungwook Choi, Swagath Venkataramani, Vijayalakshmi (Viji) Srinivasan, Kailash Gopalakrishnan, Zhuo Wang, and Pierce Chuang. Accurate and efficient 2-bit quantized neural networks. In A. Talwalkar, V. Smith, and M. Zaharia (eds.), *Proceedings of Machine Learning and Systems*, volume 1, pp. 348–359, 2019. URL https://proceedings.mlsys.org/paper_files/paper/2019/file/c443e9d9fc984cda1c5cc447fe2c724d-Paper.pdf.

Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3075–3084, 2019.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007. doi: 10.1007/s00454-006-1276-5.

Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. SalsaNext: Fast, Uncertainty-Aware Semantic Segmentation of LiDAR Point Clouds. In *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II*, volume 12510 of *Lecture Notes in Computer Science*, pp. 207–222. Springer, 2020a.

Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*, pp. 207–222. Springer, 2020b.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems (NeurIPS)*, volume 26, 2013.

Runpei Dong, Zhanhong Tan, Mengdi Wu, Linfeng Zhang, and Kaisheng Ma. Finding the task-optimal low-bit sub-distribution in deep neural networks. In *International Conference on Machine Learning*, pp. 5343–5359. PMLR, 2022.

Kilian Fatras, Thibault Séjourné, Nicolas Courty, and Rémi Flamary. Unbalanced minibatch optimal transport; applications to domain adaptation. In *International Conference on Machine Learning (ICML)*, pp. 3186–3197. PMLR, 2021.

Marcio Gameiro, Yasuaki Hiraoka, and Ippei Obayashi. Continuation of point clouds via persistence diagrams. *Physica D: Nonlinear Phenomena*, 334:118–132, 2016. doi: 10.1016/j.physd.2016.03.011.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361. IEEE, 2012.

Martin Gerdzhev, Ryan Razani, Ehsan Taghavi, and Liu Bingbing. Tornado-net: multiview total variation semantic segmentation with diamond inception module. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9543–9549. IEEE, 2021.

Tim Golla and Reinhard Klein. Real-time point cloud compression. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5087–5092, 2015. doi: 10.1109/IROS.2015.7354093.

Ruben Gomez-Ojeda, Jesus Briales, and Javier Gonzalez-Jimenez. Pl-svo: Semi-direct monocular visual odometry by combining points and line segments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4211–4216. IEEE, 2016.

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*, 7:187–199, 2021a.

Xiaoyang Guo, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Liga-stereo: Learning lidar geometry aware representations for stereo-based 3d detector. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3153–3163, 2021b.

Sangchul Hahn and Heeyoul Choi. Self-knowledge distillation in natural language processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pp. 423–430, 2019. doi: 10.26615/978-954-452-056-4_050. URL `https://aclanthology.org/R19-1050/`.

Dan Halperin and Niklas Eisl. Point cloud based scene segmentation: A survey, 2025. URL `https://arxiv.org/abs/2503.12595`.

Cheng Han, Qifan Wang, Sohail A. Dianat, Majid Rabbani, Raghuveer M. Rao, Yi Fang, Qiang Guan, Lifu Huang, and Dongfang Liu. AMD: Automatic multi-step distillation of large-scale vision models. In *European Conference on Computer Vision (ECCV)*, 2024.

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. URL `https://arxiv.org/abs/1510.00149`.

Haoyu He, Xingjian Shi, Jonas Mueller, Sheng Zha, Mu Li, and George Karypis. Distiller: A systematic study of model distillation methods in natural language processing. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pp. 119–133, Virtual, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.sustainlp-1.13. URL `https://aclanthology.org/2021.sustainlp-1.13/`.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1–2, 2014.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8479–8488, June 2022a.

Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8479–8488, 2022b.

Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8479–8488, 2022c.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *Advances in neural information processing systems (NeurIPS)*, volume 32, 2019.

Alejandro Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18146–18156, 2024.

Eun Som Jeon, Rahul Khurana, Aishani Pathak, and Pavan Turaga. Leveraging topological guidance for improved knowledge distillation. In *Proceedings of the ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024. URL `https://arxiv.org/abs/2407.05316`.

Jungeun Kim, Junwon You, Dongjin Lee, Ha Young Kim, and Jae-Hun Jung. Do topological characteristics help in knowledge distillation? In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=2dEH0u8w0b`.

Deyvid Kochanov, Fatemeh Karimi Nejadasl, and Olaf Booij. Kprnet: Improving projection-based lidar semantic segmentation. *arXiv preprint arXiv:2007.12668*, 2020.

Solomon Kullback. Kullback-leibler divergence, 1951.

Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based 3d recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17545–17555, 2023.

Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12697–12705, 2019.

Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. In *Proceedings of Robotics: Science and Systems (RSS)*, 2016. doi: 10.15607/RSS.2016.XII.042. URL `https://www.roboticsproceedings.org/rss12/p42.pdf`.

Jia-Chen Li, Jun-Guo Lu, Ming Wei, Hong-Yi Kang, and Qing-Hao Zhang. Tpv-igkd: Image-guided knowledge distillation for 3d semantic segmentation with tri-plane-view. *IEEE Transactions on Intelligent Transportation Systems*, 25(8):10405–10416, 2024a.

Jia-Chen Li, Jun-Guo Lu, Ming Wei, Hong-Yi Kang, and Qing-Hao Zhang. Tpv-igkd: Image-guided knowledge distillation for 3d semantic segmentation with tri-plane-view. *IEEE Transactions on Intelligent Transportation Systems*, 25(8):10405–10416, 2024b. doi: 10.1109/TITS.2024.3361163.

Jiale Li, Hang Dai, and Yong Ding. Self-distillation for robust lidar semantic segmentation in autonomous driving. In *European conference on computer vision*, pp. 659–676. Springer, 2022a.

Jiale Li, Hang Dai, Hao Han, and Yong Ding. Mseg3d: Multimodal 3d semantic segmentation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21694–21704, 2023.

Yiting Li, Liya Ma, Zongmin Tan, Shuo Chen, et al. Mass: Multi-attentional semantic segmentation of lidar data for dense top-view understanding. *IEEE Transactions on Intelligent Transportation Systems*, 23(9): 15824–15840, 2022b.

Venice Erin Liong, Thi Ngoc Tho Nguyen, Sergi Widjaja, Dhananjai Sharma, and Zhuang Jie Chong. Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. *CoRR*, abs/2012.04934, 2020. URL `https://arxiv.org/abs/2012.04934`.

Y. Liu and et al. Uni-to-multi modal knowledge distillation for bidirectional lidar-camera semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, PP(99):1–14, 2024.

Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2604–2613, 2019a.

Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3296–3305, 2019b.

Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through l_0 regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. URL `https://openreview.net/forum?id=H1Y8hhg0b&noteId=SkdVNJ6Bz`.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. URL `https://openreview.net/forum?id=3Pbra-_u76D`.

Sivabalan Manivasagam, Shenlong Wang, Jie Liao, Zehao Tang, and Raquel Urtasun. Pillarsegnet: Pillar-based semantic grid map estimation using sparse lidar data. *arXiv preprint arXiv:2105.04169*, 2021.

Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4213–4220. IEEE, 2019.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International Conference on Machine Learning (ICML)*, pp. 7045–7054, 2020.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1325–1334, 2019.

Yong-Joo Oh and Yoshio Watanabe. Development of small robot for home floor cleaning. In *Proceedings of the 41st SICE Annual Conference. SICE 2002.*, volume 5, pp. 3222–3223. IEEE, 2002.

Adrien Poulenard, Primoz Skraba, and Maks Ovsjanikov. Topological function optimization for continuous shape matching. *Computer Graphics Forum*, 37(5):13–25, 2018.

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

Haibo Qiu, Baosheng Yu, and Dacheng Tao. Gfnet: Geometric flow network for 3d point cloud semantic segmentation. *arXiv preprint arXiv:2207.02605*, 2022.

Shoumeng Qiu, Feng Jiang, Haiqiang Zhang, Xiangyang Xue, and Jian Pu. Multi-to-single knowledge distillation for point cloud semantic segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9303–9309. IEEE, 2023.

Ahmad Rashid, Vasileios Lioutas, Abbas Ghaddar, and Mehdi Rezagholizadeh. Towards zero-shot knowledge distillation for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6551–6561, 2021. doi: 10.18653/v1/2021.emnlp-main.526. URL `https://aclanthology.org/2021.emnlp-main.526/`.

Damien Robert, Hugo Raguet, and Loic Landrieu. Efficient 3d semantic segmentation with superpoint transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17195–17204, 2023.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL `http://arxiv.org/abs/1412.6550`.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, pp. 1–2, 2019.

Elchanan Solomon, Alexander Wagner, and Paul Bendich. From geometry to topology: Inverse theorems for distributed persistence. In *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 61:1–61:16. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPIcs.SoCG.2022.61.

Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1746–1754, 2017.

Maomao Sun, Ting Rui, Jianqing Liu, Dong Wang, and Chengsong Yang. Enhancing lidar point cloud semantic segmentation via multiscale multiview fusion and knowledge distillation. *IEEE Sensors Journal*, Early Access, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Alexandre Chouard, Anand Patnaik, Paul Tsui, Yin Guo, Yuning Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2446–2454, 2020.

Tianfang Sun, Zhizhong Zhang, Xin Tan, Yong Peng, Yanyun Qu, and Yuan Xie. Uni-to-multi modal knowledge distillation for bidirectional lidar-camera semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):11059–11072, 2024. doi: 10.1109/TPAMI.2024.3451658.

Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII*, volume 12373 of *Lecture Notes in Computer Science*, pp. 685–702. Springer, 2020.

Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6411–6420, 2019.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1365–1374, 2019.

Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Transactions On Graphics (TOG)*, 36(4):1–11, 2017.

Wenguan Wang, Tianfei Zhou, Fisher Yu, Jifeng Dai, Piotr Koniusz, and Luc Van Gool. Visual recognition with deep nearest centroids. In *International Conference on Learning Representations (ICLR)*, 2023.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Minhao Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 5776–5788, 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 (TOG)*, 38(5):1–13, 2019.

Zihan Wang, Yunhang Shen, Mengtian Li, Ke Li, Xing Sun, Shaohui Lin, and Lizhuang Ma. Knowledge transfer across modalities for weakly supervised point cloud semantic segmentation. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025.

Siyue Wu, Hongzhan Chen, Xiaojun Quan, Qifan Wang, and Rui Wang. AD-KD: Attribution-driven knowledge distillation for language model compression. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8449–8465. Association for Computational Linguistics, 2023.

Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a. URL `https://openreview.net/forum?id=I3mLa12s_H`.

Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35: 33330–33342, 2022b.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851, 2024. doi: 10.1109/CVPR52733.2024.00463.

Zhaoyang Xia, Youquan Liu, Xin Li, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, and Yu Qiao. SCPNet: Semantic scene completion on point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17642–17651, 2023.

Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16024–16033, 2021.

Xu Yan, Jiantao Gao, Chaoda Zheng, Chao Zheng, Ruimao Zhang, Shuguang Cui, and Zhen Li. 2dpass: 2d priors assisted semantic segmentation on lidar point clouds. In *European Conference on Computer Vision*, pp. 677–695. Springer, 2022.

Yu-Qi Yang, Yu-Xiao Guo, Jian-Yu Xiong, Yang Liu, Hao Pan, Peng-Shuai Wang, Xin Tong, and Baining Guo. Swin3d: A pretrained transformer backbone for 3d indoor scene understanding. *arXiv preprint arXiv:2304.06906*, 2023.

Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for lidar perception. In *AAAI Conference on Artificial Intelligence*, 2022. URL `https://api.semanticscholar.org/CorpusID:252383236`.

Feihu Zhang, Jin Fang, Benjamin Wah, and Philip Torr. Deep fusionnet for point cloud semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pp. 644–663. Springer, 2020a.

Linfeng Zhang, Runpei Dong, Hung-Shuo Tai, and Kaisheng Ma. Pointdistiller: Structured knowledge distillation towards efficient and compact 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21791–21801, June 2023.

S. Zhang, J. Deng, L. Bai, et al. Hvdistill: Transferring knowledge from images to point clouds via unsupervised hybrid-view distillation. *International Journal of Computer Vision*, 132:2585–2599, 2024. doi: 10.1007/s11263-023-01981-w. URL `https://doi.org/10.1007/s11263-023-01981-w`.

Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9601–9610, 2020b.

Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5565–5573, 2019.

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. doi: 10.1109/ICCV48922.2021.01595. URL `https://openaccess.thecvf.com/content/ICCV2021/papers/Zhao_Point_Transformer_ICCV_2021_paper.pdf`.

Lin Zhao, Siyuan Xu, Liman Liu, Delie Ming, and Wenbing Tao. Svaseg: Sparse voxel-based attention for 3d lidar point cloud semantic segmentation. *Remote Sensing*, 14(18):4471, 2022.

Tianfei Zhou, Wenguan Wang, Piotr Koniusz, and Huijun Wu. Rethinking semantic segmentation: A prototype view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2582–2593, 2022.

Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9939–9948, 2021.

Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16280–16290, 2021.

A Theoretical Foundations and Justifications

A.1 Topology of Feature vs. Topology of Spatial Coordinates

A critical distinction in our framework is the choice to distill the topology of latent feature spaces rather than physical spatial coordinates. This design is grounded in two fundamental geometric learning principles:

1. **Semantic vs. Physical Proximity:** Input spatial coordinates are constrained by Euclidean distance, capturing only physical proximity. In contrast, deep latent features evolve to capture semantic proximity. As demonstrated by Wang et al. (2019) in Dynamic Graph CNNs (DGCNN), feature-space neighborhoods in deeper layers group semantically similar objects (e.g., aggregating features from both the front and rear tires of a vehicle) even when they are geometrically distant. This allows the network to bypass physical distance, creating semantically coherent clusters that coordinate-based topology cannot capture.

2. **Invariance to Rigid Transformations:** Coordinate-based distillation objectives (e.g., Mean Squared Error on point coordinates) penalize any deviation in spatial alignment, making them sensitive to rotation and translation. Conversely, the topology of the feature space when characterized by persistent homology is theoretically invariant to such isometries. The Stability Theorem of persistence diagrams Cohen-Steiner et al. (2007) guarantees that topological descriptors depend solely on pairwise distances, rendering them invariant to the rigid transformation of the input. By aligning feature topology, we force the student to learn the intrinsic shape of the teacher's semantic manifold Moor et al. (2020), ensuring robust knowledge transfer that generalizes across geometric perturbations Jeon et al. (2024).

A.2 Subsampling and Stability of Vietoris–Rips Persistence

A major bottleneck in topological distillation is the computational cost of persistent homology (PH), which scales superlinearly with the number of points. We therefore compute PH on a subsample selected by Topology-Preserving Sampling (TPS) (Algorithm 5).

Let $H = \{h_1, \ldots, h_N\} \subset \mathbb{R}^C$ be the full feature set and let $H_I \subset H$ be the TPS subsample of size M. Define the (one-sided) covering radius

$$\varepsilon := \max_{x \in H} \min_{y \in H_I} \|x - y\|_2.$$

Because $H_I \subset H$, this equals the Hausdorff distance $d_H(H, H_I)$.

Proposition 1 (Rips interleaving under Hausdorff perturbations). Let A, B be finite subsets of a metric space $(\mathcal{X}, d)$ with $d_H(A, B) \leq \varepsilon$. Then the Vietoris–Rips filtrations $\mathrm{Rips}(A, r)$ and $\mathrm{Rips}(B, r)$ are 2ε-interleaved: for all $r \geq 0$,

$$\mathrm{Rips}(A, r) \to \mathrm{Rips}(B, r + 2\varepsilon) \quad \text{and} \quad \mathrm{Rips}(B, r) \to \mathrm{Rips}(A, r + 2\varepsilon),$$

via the nearest-neighbor correspondence. Consequently, their persistence modules are 2ε-interleaved, and the bottleneck distance between the corresponding persistence diagrams in any fixed homology dimension is at most 2ε.

Proof. For any edge (a, a') in $\mathrm{Rips}(A, r)$, choose $b, b' \in B$ with $d(a, b) \leq \varepsilon$ and $d(a', b') \leq \varepsilon$. By the triangle inequality, $d(b, b') \leq d(b, a) + d(a, a') + d(a', b') \leq r + 2\varepsilon$, so (b, b') is an edge in $\mathrm{Rips}(B, r + 2\varepsilon)$. This extends to all simplices, yielding a simplicial map. The reverse direction is analogous.

Implication for TPS. TPS directly minimizes the covering radius ε for a fixed budget M, so Proposition A.1 implies that the persistence diagrams computed on the TPS subset are a controlled approximation of the persistence diagrams of the full feature set, with error bounded (in bottleneck distance) by $O(\varepsilon)$. This motivates using TPS to reduce PH cost while preserving the dominant topological structures.

A.3 Noise Suppression via Persistence Weighting

This subsection provides the theoretical proof and Lemma 1 for Section 3.2.2, demonstrating how the logarithmic weighting function bounds the gradient influence of low-persistence features. It mathematically establishes that as the persistence of a topological feature approaches zero (representing noise), its contribution to the loss vanishes, thereby preventing the student model from overfitting to stochastic artifacts in the teacher's latent space.

Lemma 1 (Noise Suppression via Persistence Weighting)**.** *Let $\mathcal{D}^T$ and $\mathcal{D}^S$ be teacher and student persistence diagrams in a fixed homology dimension, containing only* finite *pairs. Define the persistence of a point $p = (b, d)$ as* $\mathrm{pers}(p) = d - b$*, and define the teacher weight*

$$w(p) = \log\big(1 + \mathrm{pers}(p)\big) \geq 0, \tag{9}$$

together with the stabilized normalizer

$$\bar{w}_T = \sum_{p \in \mathcal{D}^T} w(p) + \epsilon, \qquad \epsilon > 0. \tag{10}$$

Consider the weighted teacher-to-student Chamfer term

$$\mathcal{L}_{T \to S} = \frac{1}{\bar{w}_T} \sum_{p \in \mathcal{D}^T} w(p) \min_{q \in \mathcal{D}^S} \|p - q\|_2^2. \tag{11}$$

Then for any teacher point $p_0 \in \mathcal{D}^T$ with $\mathrm{pers}(p_0) = \delta$*, its contribution to $\mathcal{L}_{T \to S}$ is bounded by*

$$0 \leq \frac{w(p_0)}{\bar{w}_T} \min_{q \in \mathcal{D}^S} \|p_0 - q\|_2^2 \leq \frac{\log(1 + \delta)}{\epsilon} \min_{q \in \mathcal{D}^S} \|p_0 - q\|_2^2. \tag{12}$$

In particular, as $\delta \to 0$ (near-diagonal/noisy features), $\log(1 + \delta) \to 0$ *and this bound vanishes, so low-persistence teacher points have negligible influence on the weighted alignment term.*

Proof. The contribution of the Teacher's noise point p_{noise} to the Recall loss is given by:

$$L_p = d_{\min} \cdot \frac{w(p_{noise})}{\bar{w}_T} \tag{13}$$

Substituting the weight definition $w(p) = \log(1 + \delta)$ and the normalization factor:

$$L_p = d_{\min} \cdot \frac{\log(1 + \delta)}{\sum_k w(k) + \epsilon} \tag{14}$$

Using the first-order Taylor expansion $\log(1 + x) \approx x$ for small δ, and observing that the denominator is strictly bounded below by ϵ (since $w(k) \geq 0$):

$$L_p \approx d_{\min} \cdot \frac{\delta}{\bar{w}_T} \leq d_{\min} \cdot \frac{\delta}{\epsilon} \tag{15}$$

Taking the limit as the persistence $\delta \to 0$:

$$\lim_{\delta \to 0} L_p \leq \lim_{\delta \to 0} \left(\frac{d_{\min}}{\epsilon} \cdot \delta \right) = 0 \tag{16}$$

Since $\bar{w}_T$ is bounded away from zero ($\bar{w}_T \geq \epsilon$), the denominator does not vanish. Consequently, as the persistence $\delta \to 0$, the loss contribution $L_p \to 0$ linearly. This ensures that the Student model is not penalized for ignoring negligible artifacts in the Teacher's latent space, allowing it to focus its representational capacity on matching persistent geometric structures.

□

B Implementation and Algorithms

B.1 Differentiable Feature-Space Distillation

This subsection details the complete forward and backward pass of the topological distillation framework (Algorithm 1). It outlines the specific sequence of operations—including feature normalization, Topology-Preserving Sampling (TPS), and the calculation of pairwise distances—required to route gradients through the critical edges of the Vietoris-Rips filtration.

Algorithm 1 Differentiable Feature-Space Topological Distillation (with Subgradient Routing)

Require: Student features $\mathbf{H}^S \in \mathbb{R}^{B\times N\times C_S}$, Teacher features $\mathbf{H}^T \in \mathbb{R}^{B\times N\times C_T}$
Require: Subsample size M (1024), stabilizer $\epsilon > 0$, tie tolerance τ_{tie}
Ensure: Topological Loss $\mathcal{L}_{topo}$

$\mathcal{L}_{topo} \leftarrow 0$
for $k = 1$ to B **do**
  // Step 1: Feature Normalization
  $\tilde{\mathbf{H}}^S \leftarrow \text{Normalize}(\mathbf{H}^S[k])$
  $\tilde{\mathbf{H}}^T \leftarrow \text{Normalize}(\mathbf{H}^T[k])$
  // Step 2: Topology-Preserving Selection (See Algorithm 5)
  $I_k \leftarrow \text{TPS}(\tilde{\mathbf{H}}^T, M)$
  $\mathbf{S} \leftarrow \tilde{\mathbf{H}}^S[I_k], \quad \mathbf{T} \leftarrow \tilde{\mathbf{H}}^T[I_k]$
  // Step 3: Distance Matrix Computation
  $\mathbf{D}^S \leftarrow \text{PairwiseDist}(\mathbf{S}), \quad \mathbf{D}^T \leftarrow \text{PairwiseDist}(\mathbf{T})$
  // Step 4: Topological Summarization (Forward with Tie-breaking)
  $\{\mathcal{D}_d^S\}_{d\in\{0,1\}} \leftarrow \text{DiffRipser}(\mathbf{D}^S; \tau_{\text{tie}})$
  $\{\mathcal{D}_d^T\}_{d\in\{0,1\}} \leftarrow \text{Ripser}(\mathbf{D}^T)$
  // Step 5: Persistence Weighted Chamfer Distance (See Algorithm 2)
  for $d \in \{0, 1\}$ **do**
    $\mathcal{L}_{topo} \leftarrow \mathcal{L}_{topo} + \text{PWCD}(\mathcal{D}_d^T, \mathcal{D}_d^S; \epsilon)$
  end for
end for
return $\mathcal{L}_{topo}/B$

B.2 Persistence-Weighted Chamfer Loss

This subsection defines the specialized loss function used to align the persistence diagrams of the student and teacher. It explains the logarithmic weighting mechanism designed to suppress low-persistence sampling noise while prioritizing robust topological features like loops and clusters during optimization (Algorithm 2).

B.3 Implementation of Gradient Routing

While persistent homology is computed by discrete operations (sorting simplices and boundary-matrix reduction), the birth/death times are piecewise-smooth functions of the underlying pairwise distances. For the Vietoris–Rips filtration (simplex filtration value equals the maximum edge length), each birth/death value is realized by one or more maximizer edges in the distance matrix. At degenerate configurations where multiple edges tie for the same filtration value, the gradient is not unique; we use a standard subgradient convention by averaging the gradient uniformly over all tied maximizer edges. This yields stable optimization and avoids relying on fragile uniqueness assumptions. The method is described in Algorithm 3.

B.4 Complexity Reduction via Topology-Preserving Selection

To implement efficient subsampling, we employ Topology-Preserving Sampling (TPS), which is a greedy approximation to the k-center problem. TPS explicitly minimizes the covering radius (and thus the Hausdorff

Algorithm 2 Persistence-Weighted Chamfer Loss

Require: Teacher diagram $\mathcal{P}^T = \{p_i\}_{i=1}^M \subset \mathbb{R}^2$; Student diagram $\mathcal{P}^S = \{q_j\}_{j=1}^N \subset \mathbb{R}^2$; stabilizer $\epsilon > 0$.
Ensure: Topology loss $\mathcal{L}_{\text{topo}}$.

// 1) Trivial case
if $M = 0 \wedge N = 0$ **then**
  return 0
end if
// 2) Teacher weights
if $M > 0$ **then**
  $WT[i] \leftarrow log(1 + max(p_i^{(2)} - p_i^{(1)}, 0))$ # $p = (b, d)$
  $\bar{W}_T \leftarrow \sum_{i=1}^M W_T[i] + \epsilon$
end if
// 3) Helper: squared distance to diagonal
$\text{dist}_\Delta^2(b, d) \leftarrow \frac{(d-b)^2}{2}$
// 4) If both non-empty, compute pairwise costs
if $M > 0 \wedge N > 0$ **then**
  $C_{ij} \leftarrow \frac{1}{2}\|p_i - q_j\|_2^2$ # $M \times N$
end if
// 5) Recall term (Teacher → Student) with diagonal fallback
if $M = 0$ **then**
  $L_{\text{recall}} \leftarrow 0$
else if $N = 0$ **then**
  $L_{\text{recall}} \leftarrow \frac{1}{\bar{W}_T} \sum_{i=1}^M W_T[i]\ \text{dist}_\Delta^2(p_i^{(1)}, p_i^{(2)})$
else
  $\min_{T\to S}[i] \leftarrow \min_j C_{ij}$
  $L_{\text{recall}} \leftarrow \frac{1}{\bar{W}_T} \sum_{i=1}^M W_T[i]\ \min\Big\{\min_{T\to S}[i], \text{dist}_\Delta^2(p_i^{(1)}, p_i^{(2)})\Big\}$
end if
// 6) Precision term (Student → Teacher) with diagonal fallback
if $N = 0$ **then**
  $L_{\text{prec}} \leftarrow 0$
else if $M = 0$ **then**
  $L_{\text{prec}} \leftarrow \frac{1}{N} \sum_{j=1}^N \text{dist}_\Delta^2(q_j^{(1)}, q_j^{(2)})$
else
  $\min_{S\to T}[j] \leftarrow \min_i C_{ij}$
  $L_{\text{prec}} \leftarrow \frac{1}{N} \sum_{j=1}^N \min\Big\{\min_{S\to T}[j], \text{dist}_\Delta^2(q_j^{(1)}, q_j^{(2)})\Big\}$
end if
return $L_{\text{recall}} + L_{\text{prec}}$

distance) between the selected subset and the full feature set. This directly reduces the worst-case approximation error of geometric/topological summaries under a fixed budget. We use TPS to select indices, then compute PH only on the selected subset. The algorithm is described in Algorithm 5.

Quantitative Efficiency Analysis. To validate the computational efficiency of our hybrid pipeline (combining the Ripser Bauer (2021), critical-edge gradient routing, and TPS), we benchmark the total forward-and-backward runtime against a naive pure-PyTorch implementation. As shown in Table 6, the naive approach suffers from combinatorial complexity explosion, becoming computationally intractable beyond $N = 100$ and exhausting GPU memory at $N = 200$. In contrast, our method maintains real-time performance, achieving speedup factors exceeding 6700× at $N = 100$ and scaling efficiently to $N = 1024$ with negligible overhead.

Algorithm 3 Critical-Edge Gradient Routing

Require: Student features $F \in \mathbb{R}^{N\times C}$; teacher diagrams $\{P_T^{(d)}\}_{d\in\{0,1\}}$; stabilizers $\varepsilon_{\text{dist}}, \varepsilon_w > 0$; tie tolerance τ_{tie}.
Ensure: Feature gradients $\nabla_F \mathcal{L}$.

Compute distance matrix D: $D_{ij} \leftarrow \|F_i - F_j\|_2$ for $1 \le i < j \le N$.
Initialize edge accumulator $G \in \mathbb{R}^{N\times N}$ with zeros.
for $d \in \{0, 1\}$ **do**
  // Forward pass: compute diagrams and extract critical simplices
  Run Ripser(D) to obtain student diagram $P_S^{(d)}$ and pairing map $\Pi^{(d)}$.
  Compute diagram gradients: $(g_b, g_d) \leftarrow \text{PWCDGrad}(D_T^{(d)}, D_S^{(d)}; \varepsilon_w)$ *// See algorithm 4.*
  for $k = 1$ **to** $|D_S^{(d)}|$ **do**
    Let $(\sigma_{\text{birth}}, \sigma_{\text{death}}) \leftarrow \Pi^{(d)}(k)$.
    // Route birth gradient (only for H_1)
    if $d = 1$ **then**
      Let $\{i, j\} = \sigma_{\text{birth}}$.
      $G_{ij} \leftarrow G_{ij} + g_b[k]$.
    end if
    // Route death gradient
    if $d = 0$ **then**
      Let $\{u, v\} = \sigma_{\text{death}}$ be the destroying edge.
      $G_{uv} \leftarrow G_{uv} + g_d[k]$.
    else
      // For H_1, death is caused by a triangle $\{a, b, c\}$ completing a cycle
      Let $E_\Delta = \{(a, b), (b, c), (a, c)\}$.
      $m \leftarrow \max_{(u,v)\in E_\Delta} D_{uv}$.
      $E^\star \leftarrow \{(u, v) \in E_\Delta : |D_{uv} - m| \le \tau_{\text{tie}}\}$.
      for each $(u, v) \in E^\star$ **do**
        $G_{uv} \leftarrow G_{uv} + g_d[k]/|E^\star|$.
      end for
    end if
  end for
end for
// Backpropagate distances to features
Initialize $\nabla_F \mathcal{L} \leftarrow \mathbf{0}$.
for $1 \le i < j \le N$ **do**
  if $G_{ij} \neq 0$ **then**
    $u_{ij} \leftarrow (F_i - F_j)/(D_{ij} + \varepsilon_{\text{dist}})$.
    $\nabla_{F_i} \mathcal{L} \leftarrow \nabla_{F_i} \mathcal{L} + G_{ij} \cdot u_{ij}$.
    $\nabla_{F_j} \mathcal{L} \leftarrow \nabla_{F_j} \mathcal{L} - G_{ij} \cdot u_{ij}$.
  end if
end for
return $\nabla_F \mathcal{L}$.

C Experimental Setup and Hyperparameters

C.1 Datasets

SemanticKITTI Behley et al. (2019) provides LiDAR point clouds from urban and suburban scenes, featuring 22 sequences with dense semantic annotations across 19 classes (e.g., vehicles, pedestrians, roads). Its high resolution and detailed labels make it a rigorous testbed for semantic segmentation.

NuScenes Caesar et al. (2020) integrates LiDAR, camera, and radar data in 1,000 diverse scenes, including urban roads and highways. With 3D bounding box annotations for 23 object types, it challenges models with varied weather, occlusions, and dynamic elements suited for detection and segmentation tasks. In addition, we use nuScenes-lidar seg, which is an extension of nuScenes. This dataset has semantic labels of 32 categories and annotates each point from keyframes in nuScenes. We used the 700 scenes in the training

Algorithm 4 PWCDGrad: Diagram-space gradients for Persistence-Weighted Chamfer Distance

Function PWCDGrad(D_T, D_S, ε_w):
  Input: Teacher $D_T = \{p_i\}_{i=1}^{M}$, Student $D_S = \{q_k\}_{k=1}^{K}$, stabilizer ε_w
  Output: Gradient arrays g_b, g_d of size K
  Initialize $g_b \leftarrow \mathbf{0},\ g_d \leftarrow \mathbf{0}$
  Define $w(b, d) = \log(1 + \max(d - b, 0))$
  Compute $\bar{w}_T = \varepsilon_w + \sum_{i=1}^{M} w(b_i^T, d_i^T)$
  // 1. Recall Term (Teacher → Student)
  if $M > 0$ **and** $K > 0$ **then**
    for $i = 1$ **to** M **do**
      $k^\star \leftarrow \arg\min_k \|q_k - p_i\|_2^2$ *// Find nearest student*
      $dist_{match} \leftarrow \|q_{k^\star} - p_i\|_2^2$
      $dist_{diag} \leftarrow \frac{1}{2}(d_i^T - b_i^T)^2$
      if $dist_{match} \leq dist_{diag}$ **then**
        $s \leftarrow w(b_i^T, d_i^T)/\bar{w}_T$
        $g_b[k^\star] \mathrel{+}= 2s \cdot (b_{k^\star} - b_i^T)$
        $g_d[k^\star] \mathrel{+}= 2s \cdot (d_{k^\star} - d_i^T)$
      end if
    end for
  end if
  // 2. Precision Term (Student → Teacher)
  if $K > 0$ **then**
    for $k = 1$ **to** K **do**
      $q_k \leftarrow (b_k, d_k)$
      $is_matched \leftarrow$ **false**
      if $M > 0$ **then**
        $i^\star \leftarrow \arg\min_i \|q_k - p_i\|_2^2$ *// Find nearest teacher*
        $dist_{match} \leftarrow \|q_k - p_{i^\star}\|_2^2$
        $dist_{diag} \leftarrow \frac{1}{2}(d_k - b_k)^2$
        if $dist_{match} \leq dist_{diag}$ **then**
          $g_b[k] \mathrel{+}= \frac{2}{K}(b_k - b_{i^\star}^T)$
          $g_d[k] \mathrel{+}= \frac{2}{K}(d_k - d_{i^\star}^T)$
          $is_matched \leftarrow$ **true**
        end if
      end if
      // Fallback to diagonal if no match found or teacher empty
      if not $is_matched$ **then**
        $\delta \leftarrow d_k - b_k$
        $g_b[k] \mathrel{-}= \frac{1}{K}\delta$
        $g_d[k] \mathrel{+}= \frac{1}{K}\delta$
      end if
    end for
  end if
  return (g_b, g_d)

set with segmentation labels to fine-tune for the semantic segmentation task, and the 150 scenes in the validation set to verify the performance.

Waymo Open Dataset Sun et al. (2020) delivers high-resolution LiDAR data from 1,000 segments in various locations, with frequent sweeps and 3D annotations for vehicles, pedestrians and cyclists. Its long-range scans and varied conditions test robustness and generalization.

C.2 Model Configuration and Training

We use Point Transformer V3 Wu et al. (2024) as the backbone. The student model is approximately 40-50% of the block depth (2.33× fewer encoder blocks; 2× fewer decoder blocks), trained from scratch.

Algorithm 5 Topology-Preserving Sampling (TPS)

Require: Feature set $\mathcal{H} = \{h_1, \ldots, h_N\}$, target size M
Ensure: Indices $I \subset \{1, \ldots, N\}$ with $|I| = M$
Initialize $I \leftarrow \{i_1\}$ (random index); $D_{\min}[i] \leftarrow \|h_i - h_{i_1}\|_2$
for $k = 2$ to M **do**
  $i_k \leftarrow \arg\max_i D_{\min}[i]$
  $I \leftarrow I \cup \{i_k\}$
  $D_{\min}[i] \leftarrow \min\big(D_{\min}[i], \|h_i - h_{i_k}\|_2\big) \quad \forall i$
end for
return I

	Naive Baseline			Ours			
Input (N)	**Fwd (s)**	**Bwd (s)**	**Mem (MB)**	**Fwd (s)**	**Bwd (s)**	**Mem (MB)**	**Speedup**
30	19.31	0.12	17.1	0.24	0.002368	8.2	79×
60	321.12	0.32	40.2	0.05	0.000250	16.5	6,039×
100	963.30	1.06	128.6	0.14	0.000254	17.3	6,743×
200	*Infeasible (Timeout)*			0.36	0.000271	22.7	—
400	*Infeasible (Timeout)*			0.30	0.000265	29.1	—
800	*Infeasible (Timeout)*			0.40	0.000236	42.0	—
1024	*Infeasible (Timeout)*			0.37	0.000258	49.1	—

Table 6: **Computational Efficiency Benchmark.** Comparison of Forward (Fwd) and Backward (Bwd) pass runtimes and Peak Memory usage. While the Naive implementation suffers from cubic complexity, becoming infeasible beyond $N = 100$, our method maintains sub-second latency even at $N = 1024$.

- **Encoder Depths:** Teacher $(2, 2, 2, 6, 2) \rightarrow$ Student $(1, 1, 1, 2, 1)$.
- **Encoder Channels:** Teacher $(32, 64, 128, 256, 512) \rightarrow$ Student $(16, 16, 32, 64, 128)$.
- **Attention Heads:** Scaled down from Teacher $(2, 4, 8, 16, 32)$ to Student $(1, 1, 2, 4, 8)$.

Hyperparameter Selection. Exhaustively grid-searching the optimal distillation parameters across all three large-scale datasets from scratch is highly computationally demanding. Therefore, our selection strategy was to perform a targeted sweep using the nuScenes training split, selecting weights by validation mIoU; then we reused the same weights unchanged for SemanticKITTI and Waymo. We evaluated a grid where $\lambda_{\text{topo}} \in \{0.5, 4.0, 8.0\}$, and $\lambda_{\text{grad}}, \lambda_{\text{kd}} \in \{0.5, 1.0, 2.0\}$. The optimal configuration discovered on nuScenes ($\lambda_{\text{topo}} = 8.0$, $\lambda_{\text{grad}} = 1.0$, $\lambda_{\text{kd}} = 1.0$) was subsequently applied to SemanticKITTI and Waymo. The detailed grid search results on the nuScenes dataset are provided in Table 7.

Training Protocol. Models are trained for 50 epochs (batch size 12) using AdamW (LR 0.002, Weight Decay 0.005) with a OneCycleLR scheduler to ensure stable convergence and effective generalization. We employ a two-stage distillation strategy: the teacher is pre-trained to convergence before guiding the student.

C.3 Data Augmentation and Resources

We apply random rotation ($\pm 1°$), uniform scaling ($0.9 - 1.1\times$), random flipping, and Gaussian jittering ($\sigma = 0.005$) before grid sampling (0.05 m). All experiments were conducted on a University HPC Cluster node equipped with 2 NVIDIA A100 GPUs (81 GB VRAM).

C.4 Detailed Efficiency and Memory Profiling

Tables 8 and 9 compare the Teacher (46.16M params) and Student (2.78M params) models on nuScenes, showcasing the significant efficiency benefits of our distillation framework. The Teacher model possesses

λ_{topo}	λ_{grad}	λ_{KLD}	mIoU
0.5	0.5	0.5	77.0
0.5	0.5	1.0	77.2
0.5	0.5	2.0	77.1
0.5	1.0	0.5	77.1
0.5	1.0	1.0	77.3
0.5	1.0	2.0	77.2
0.5	2.0	0.5	76.9
0.5	2.0	1.0	77.1
0.5	2.0	2.0	77.0
4.0	0.5	0.5	77.5
4.0	0.5	1.0	77.7
4.0	0.5	2.0	77.6
4.0	1.0	0.5	77.6
4.0	1.0	1.0	77.8

λ_{topo}	λ_{grad}	λ_{KLD}	mIoU
4.0	1.0	2.0	77.7
4.0	2.0	0.5	77.3
4.0	2.0	1.0	77.6
4.0	2.0	2.0	77.5
8.0	0.5	0.5	77.8
8.0	0.5	1.0	78.0
8.0	0.5	2.0	77.9
8.0	1.0	0.5	77.9
8.0	**1.0**	**1.0**	**78.2**
8.0	1.0	2.0	78.0
8.0	2.0	0.5	77.7
8.0	2.0	1.0	77.9
8.0	2.0	2.0	77.8

Table 7: Ablation study of distillation loss weights on the nuScenes dataset.

16.6× more parameters than the Student, necessitating a deeper architecture suited for high-accuracy tasks on high-end hardware. In contrast, the Student's lightweight design significantly reduces computational overhead, achieving a 36.70× reduction in encoder FLOPs. This efficiency translates to a 2.47× faster total CPU time (203.10 ms vs. 501.11 ms) and a 4.19× faster total CUDA time (102.07 ms vs. 427.31 ms), making it highly suitable for real-time applications.Additionally, the Student requires 4.5× less peak CUDA memory for matrix multiplication operations (3.57 GB vs. 16.05 GB), which allows it to benefit more effectively from Flash Attention optimizations. At the operational level, the Teacher's resource demands are markedly higher, requiring between 2.91× and 4.69× more memory for operations such as Alloc, Idx, and LN. While the Teacher provides an accuracy ceiling on high-performance hardware, the Student's drastically reduced memory footprint and accelerated execution make it ideal for deployment on resource-constrained edge devices.

Metric	**Teacher (46.16M)**	**Student (2.78M)**	**Comparison**
Total Parameters	46,160,000 (∼46.16M)	2,780,000 (∼2.78M)	Student is 16.6× smaller
Encoder Depths	(2, 2, 2, 6, 2)	(1, 1, 1, 2, 1)	2.33× fewer blocks
Encoder Channels	(32, 64, 128, 256, 512)	(16, 16, 32, 64, 128)	2–4× smaller
Encoder Attention Heads	(2, 4, 8, 16, 32)	(1, 1, 2, 4, 8)	2–4× fewer
Decoder Depths	(2, 2, 2, 2)	(1, 1, 1, 1)	2× fewer blocks
Decoder Channels	(64, 64, 128, 256)	(64, 64, 128, 128)	Last stage is 2× smaller
Decoder Attention Heads	(4, 4, 8, 16)	(2, 2, 4, 8)	2× fewer
Patch Size	1024	1024	Same
Encoder GFLOPs	380.25	10.36	36.7× lower
Decoder GFLOPs	116.44	33.45	3.48× lower
Attention Compute (Encoder)	22.58	0.60	37.63× lower
Inference Time (excl. overhead)	∼0.0592 s	∼0.0362 s	1.64× faster
Batch Inference Time	∼7.34 s	∼4.38 s	Student faster
FPS	∼16.6	∼27.6	≈ 1.64× higher
Fixed Overhead	∼0.018 s	∼0.011 s	≈ 1.64× faster
Attention Mechanism	Flash Attention	Flash Attention	Student benefits more

Table 8: Comparison of Teacher and Student models on NuScenes.

Metric	Teacher	Student	Improvement
Total CPU Time[b]	501.11 ms	203.10 ms	2.47× faster
Total CUDA Time	427.31 ms	102.07 ms	4.19× faster
MM_{add} (CUDA Mem)[a]	16.05 GB	3.57 GB	4.5× less
MM_{add} (Self CUDA)[a]	173.51 ms	34.75 ms	5.0× faster
Alloc (CUDA Memory)	8.73 GB	3.00 GB	2.91× less
Idx (CUDA Memory)	7.27 GB	1.78 GB	4.08× less
GELU (CUDA Memory)	6.82 GB	1.67 GB	4.08× less
LN (CUDA Memory)[a]	4.62 GB	985.43 MB	4.69× less
Infer (Self CPU Time)	47.01 ms	37.24 ms	1.26× faster
MM_{add} (Self CPU Time)[a]	5.40 ms	2.72 ms	1.99× faster

[a] Attention-related ops: matrix multiplications for Q, K, V and layer normalization.
[b] Total CPU time is from a profiling run; separate memory-focused run shows 3.02× ratio (Teacher: 278.80 ms, Student: 92.44 ms).

Table 9: Comparison of memory and time usage between Teacher and Student.

D Additional Ablation and Sensitivity Analysis

D.1 Visualization of Prediction

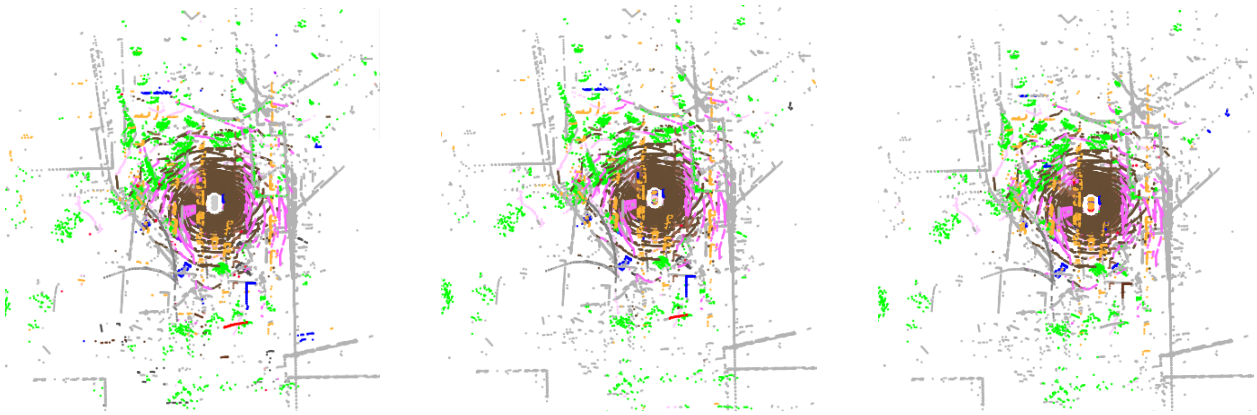
To intuitively assess the effectiveness of our distillation framework, we visualize the semantic segmentation predictions on the nuScenes validation set in Figure 3. The comparison includes the Ground Truth (a), the Teacher's prediction (b), and the Student's prediction (c). As observed, the Teacher model (b) generates high-fidelity predictions that closely align with the Ground Truth (a), particularly in capturing complex structural layouts such as road boundaries and vehicle clusters. Crucially, the Student model (c), despite its reduced capacity, successfully replicates this performance. The Student accurately preserves the global topology of the scene. Large-scale connected components, such as the drivable surface (brown) and the surrounding road network, are segmented with high continuity, which is effectively minimizing fragmentation. This contributes to our Topological Distillation, which enforces the preservation of persistent geometric features like loops and clusters Hu et al. (2019). Additionally, small and scattered classes, such as vegetation (green points) and thin linear structures like barriers (gray lines), are sharply resolved. Standard distillation methods often fail to capture these high-frequency details due to limited receptive fields or effective context aggregation Liu et al. (2019a). The sharpness of these boundaries in Figure 3c confirm that our Saliency-Aware Alignment successfully focused the Student's attention on these spatially discriminative regions, preventing them from being washed out by the dominant background classes.

Figure 4 illustrates how persistent homology captures the evolution of topological features across different filtration scales. Given a point cloud, we construct a simplicial complex and track the birth and death of topological structures as the filtration parameter ϵ increases. The persistence diagram $D = \{(b_i, d_i)\}_{i=1}^{M}$ quantifies these events, where each point represents a topological feature. Longer bars correspond to persistent structures that encode essential geometric patterns, while shorter bars typically represent noise or minor perturbations.

Visualizing the persistence diagrams allows us to better understand the types of geometric features captured by the teacher model, such as connected components (H_0), loops (H_1) and voids (H_2). By encouraging the student to mimic these persistent topological features through topology-aware distillation, we aim to transfer not only semantic knowledge but also the critical underlying geometric structures necessary for robust point-cloud understanding. This visualization supports the intuition behind our method, showing that topological summaries can effectively reflect meaningful geometric information beyond what is captured by the Euclidean feature alignment.

D.2 Boundary Visualization and Failure Cases

Qualitative Improvements. We visualise cropped regions where the baseline lightweight student (with standard KD) severely fractures continuous spatial structures and exhibits boundary discontinuity. Our



(a) Ground Truth (b) Teacher Prediction (c) Student Prediction

Figure 3: Visualization of our method on the nuScenes validation set. (a) Ground truth, (b) teacher model prediction, and (c) student model prediction. The student model closely follows the teacher's output and ground truth, successfully capturing almost all object classes, demonstrating the effectiveness of the knowledge distillation process.

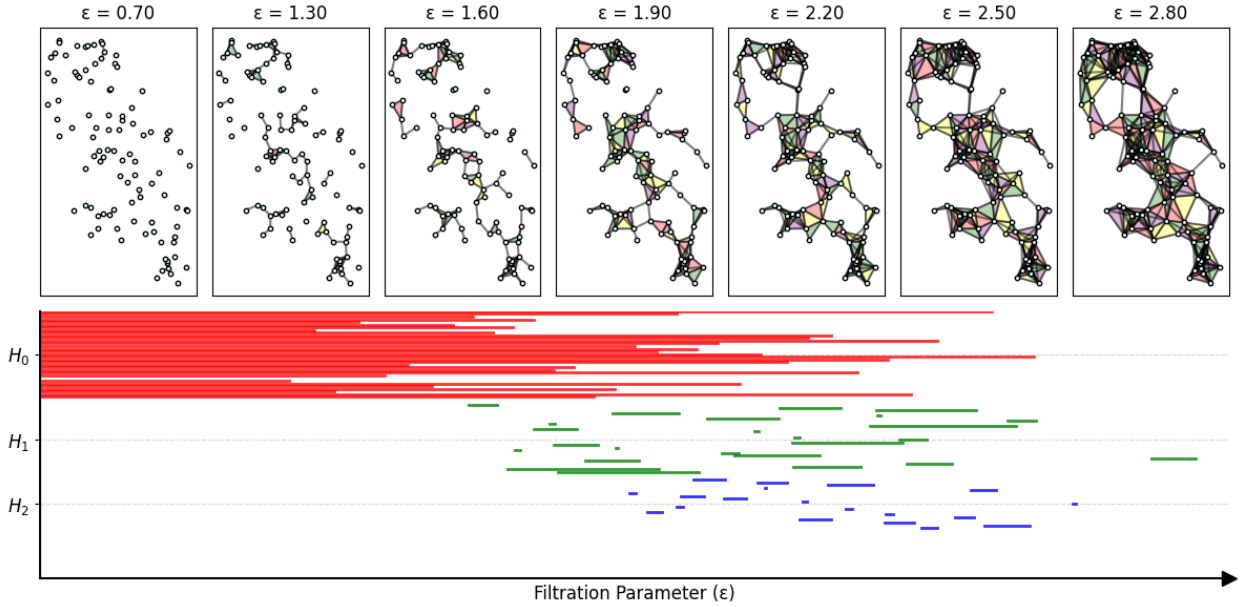


Figure 4: Visualization of topology-aware analysis through Vietoris–Rips filtration. The top row depicts the evolution of the simplicial complex as the filtration parameter ϵ increases. The bottom part shows the corresponding barcode representation of persistent homology groups in different dimensions (H_0, H_1, H_2).

distilled student successfully restores these unbroken, continuous rings and sharpens semantic boundaries. These qualitative examples show reduced fragmentation of large connected regions and cleaner boundaries, which is consistent with our goal of preserving dominant H_0 and H_1 structures in the distilled representation.

t-SNE Visualization. To showcase category boundaries, we have included a t-SNE visualization of the learned point embeddings (using the same feature level used by our distillation constraints) to illustrate how well different categories separate in the feature space. Concretely, we sampled points from the validation set and extracted the pre-classifier embeddings for (i) the teacher, (ii) the student baseline with standard KD, and (iii) our full method. The t-SNE plot shows that the standard-KD student exhibits feature over-

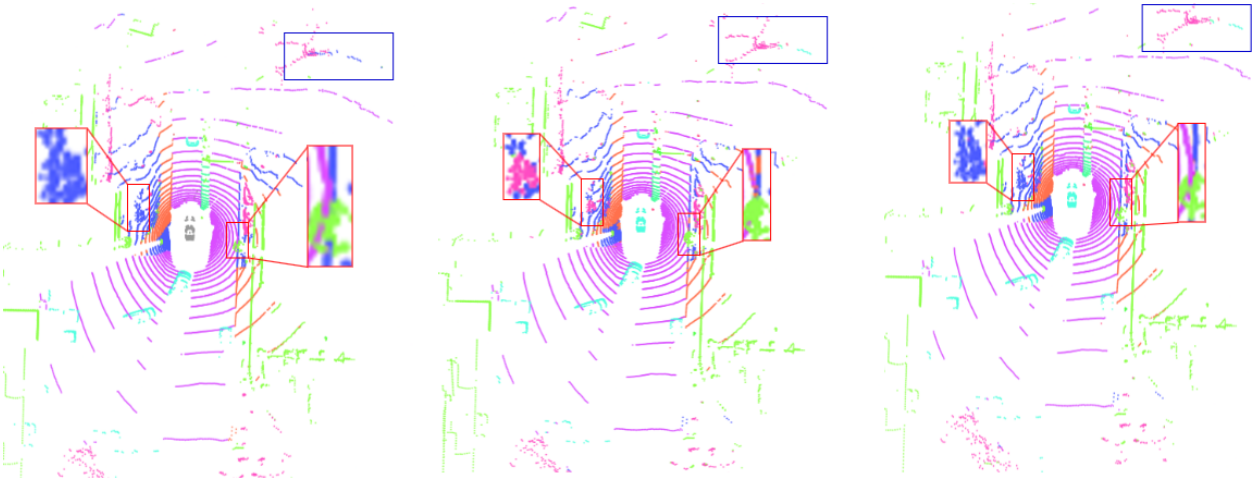


Figure 5: **Qualitative Comparison.** (Left) Ground Truth, (Middle) Student with standard KD, and (Right) Student with our full method (topology and gradient-guided KD). The baseline student produces fragmented predictions on continuous manifolds, whereas our Persistence-Weighted Chamfer loss corrects these structural errors by enforcing topological consistency. Red boxes highlight regions of improved continuity and boundary sharpness, while blue boxes highlight failure cases in extremely sparse regions.

smoothing, resulting in ambiguous, overlapping category boundaries. By introducing our gradient-guided Feature Alignment ($\mathcal{L}_{\text{grad}}$), the student focuses its representational capacity on high-gradient boundary regions, which discourages feature blending and leads to noticeably sharper and more distinct category boundaries. Moreover, the global structure of the student's latent manifold more closely mirrors the teacher, consistent with our topological loss ($\mathcal{L}_{\text{topo}}$) transferring global feature-space geometry without fragmentation.

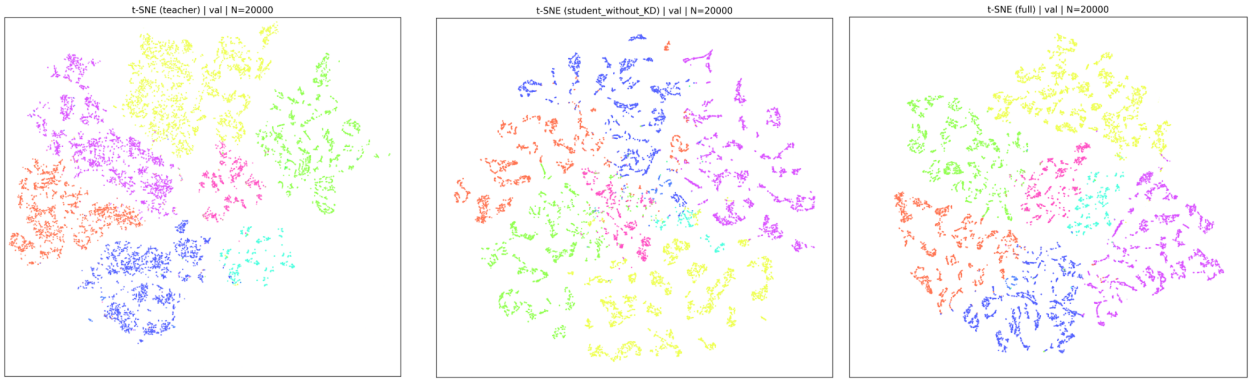


Figure 6: **t-SNE Visualization of Feature Embeddings.** (Left) Teacher, (Middle) Student with standard KD, and (Right) Student with our full method. Our approach yields sharper and more distinct category boundaries compared to the standard KD baseline by discouraging feature blending.

Failure Case Analysis. We also highlight and discuss scenarios where our framework currently struggles (indicated by the blue boxes in Figure 5). At extreme ranges, LiDAR points become very sparse, so local neighborhoods become unreliable and the Vietoris–Rips filtration yields few informative 1D structures (H_1) beyond trivial disconnected components (H_0). In these sparse regions, the topological term therefore provides weaker geometric supervision, and the student can struggle similarly to the baseline. We include a discussion of this sparsity limitation, as well as instances of topological ambiguity (where distinct semantic classes can share similar geometric topologies), which can lead to localized misclassifications.

D.3 Topological Sensitivity

Homology Dimension (H_1 vs. H_2). We restrict our experiments to H_0 and H_1. As shown in Figure 8, extending to H_2 causes factorial runtime growth due to the combinatorial explosion of 2-simplices (triangles) required for the filtration. Specifically, while the number of edges scales quadratically, the number of triangles scales cubically, causing the boundary matrix reduction step to become a computational bottleneck. Our benchmark reveals that for inputs as small as $N = 400$, computing H_2 incurs a latency spike orders of magnitude higher than H_1, making it infeasible for iterative training. Furthermore, for surface-based 2.5D LiDAR data, H_2 (voids) offers diminishing returns compared to clustering (H_0) and loops (H_1). Since LiDAR scans capture the *boundary* of objects rather than their volume, genuine enclosed voids are rarely observed; the topological signal is predominantly carried by 1-dimensional cycles (e.g., window frames, poles, and tires), which are fully captured by H_1.

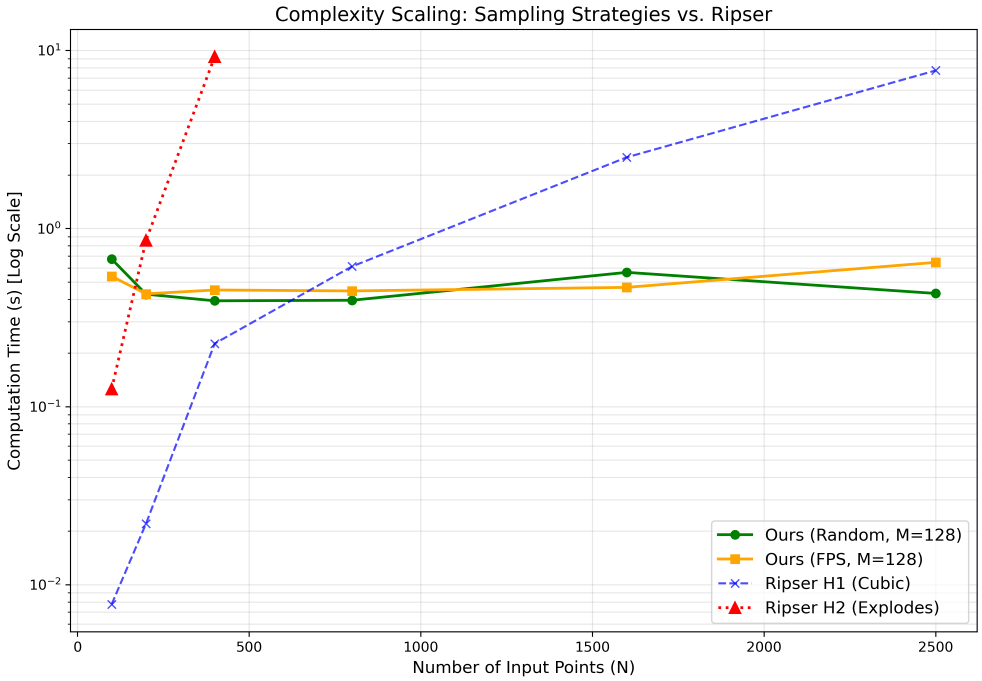


Figure 7: **Complexity Scaling of Subsampling Strategies.** Comparison of computation time between Topology-Preserving Sampling (TPS) as a function of input point cloud size N, with a fixed subsample size $M = 128$. Random sampling maintains nearly constant time, whereas TPS scales linearly with N, highlighting the trade-off between geometric coverage and computational efficiency for large inputs.

Noise Mitigation. Standard topological loss functions are often hypersensitive to sampling noise, treating transient, short-lived features with the same importance as robust structural loops. To prevent the student from overfitting to these artifacts, we introduce a logarithmic weighting function, $w(p) = \log(1 + \text{pers}(p))$, which scales the gradient magnitude based on feature persistence. As demonstrated in Figure 9, this function effectively suppresses the gradients for low-persistence noise ($p \approx 0$) while preserving strong training signals for high-persistence structural features. This ensures the student minimizes the impact of sampling anomalies and focuses solely on learning the robust manifold topology.

D.4 Gradient Alignment Sensitivity

Our current formulation computes gradient-based saliency with respect to hidden feature activations and aligns teacher/student saliency maps across multiple layers from the decoder. As shown in Table 10, attempting gradient alignment on the encoder stages degrades performance compared to the baseline, as early-stage

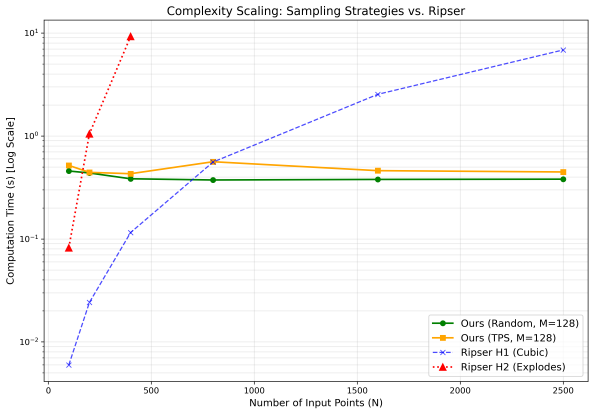


Figure 8: **Complexity Scaling (H_1 vs. H_2).** H_2 computation is intractable for training loops.

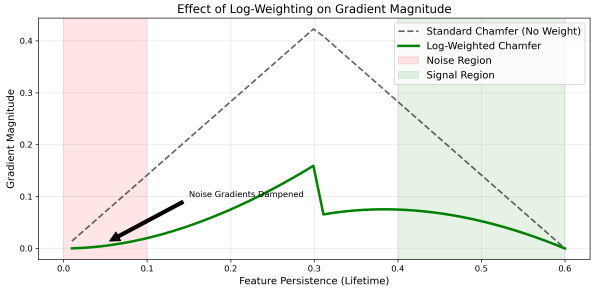


Figure 9: **Gradient Dampening.** Log-weighting suppresses noise (red) vs signal (green).

gradients are too diffuse and lack task-specific semantic focus. Conversely, alignment within the decoder stages yields significant gains. While aligning a single early-decoder layer performs strongly on simple geometric abstractions, our full method utilizes multi-scale alignment across all decoder layers. This multi-scale approach proves essential for complex, real-world datasets like nuScenes, as it combines the global semantic context from early decoder layers with the fine-grained boundary refinement of late decoder layers, yielding the best overall robustness and a peak mIoU of 78.2%.

Gradient Source Layers	**mIoU (%)**
None (Topo + KLD only)	77.6
Encoder Early (enc1-enc2)	76.5
Encoder Late (enc5)	76.8
Decoder Early (dec1)	77.9
Decoder Late (dec4)	77.7
All Decoder (dec1-dec2-dec3-dec4)	**78.2**

Table 10: Ablation on Gradient Selection Layers (nuScenes). We compare the impact of extracting gradient-guided saliency maps ($\mathcal{L}_{grad}$) from different subsets of the encoder and decoder stages.

D.5 Effectiveness of Persistence-Weighted Chamfer Distance over advanced Optimal Transport Algorithms

To justify our use of the Persistence-Weighted Chamfer Distance (PWCD) over advanced Optimal Transport methods, we replaced PWCD with entropically regularized Sinkhorn OT Cuturi (2013) in our topological distillation module ($\mathcal{L}_{\text{topo}}$). Because Sinkhorn is highly sensitive to the entropic regularization parameter ϵ, we swept multiple values.

As shown in Table 11, with tuned ϵ, Sinkhorn OT can slightly improve over the baseline, but it is sensitive to ϵ and can underperform for other values. We attribute this to gradient diffusion. Diagnostic tracking during training reveals that Sinkhorn's soft-matching creates high gradient entropy when the student's persistence diagram is noisy. Instead of suppressing noise, it partially matches noisy student features to distant teacher features. Conversely, PWCD acts as a hard nearest-neighbor assignment with a diagonal fallback, yielding highly localized gradients (lower entropy) that safely push topological noise to zero persistence. Furthermore, PWCD achieves this increased performance without requiring the exhaustive hyperparameter tuning inherent to Unbalanced OT.

Topological Distance Metric	Hyperparameters	Gradient Entropy ↓	mIoU (nuScenes) ↑
None (CE + KD Baseline)	-	-	76.9
Sinkhorn OT	$\epsilon = 0.02$	0.93	76.8
Sinkhorn OT	$\epsilon = 0.05$	0.88	77.0
Sinkhorn OT	$\epsilon = 0.10$	0.82	77.2
Sinkhorn OT	$\epsilon = 0.20$	0.84	77.1
PWCD (Ours)	**Parameter-free**	**0.75**	**77.6**

Table 11: Ablation comparing Sinkhorn Optimal Transport to our Persistence-Weighted Chamfer Distance. Lower gradient entropy indicates more localized, stable gradient routing.

D.6 Validation of Topological Correctness

To ensure the validity of our custom PyTorch implementation, we verified that it produces persistence diagrams topologically equivalent to the C++ `Ripser` library Bauer (2021). Figure 10 demonstrates this alignment across different geometric primitives.

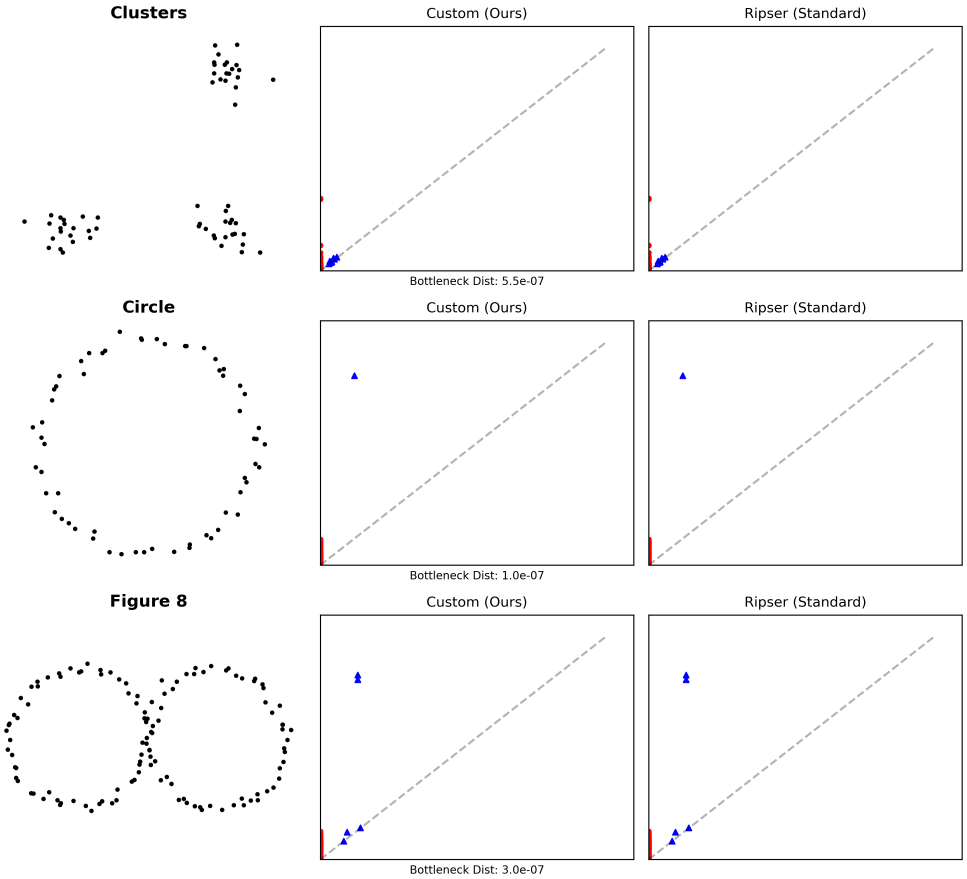


Figure 10: **Validation of Topological Correctness.** Visual comparison of persistence diagrams generated by our pure-PyTorch baseline versus `Ripser`. The diagrams are topologically equivalent with negligible bottleneck distances.