Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2026) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

30th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2026)

RangeViM: Full-Resolution Range-View LiDAR Segmentation with a Lightweight Hybrid Vision Mamba Backbone

Tuan-Kiet Huynh-Cao[a,b,**], Minh-Man Ly-Dinh[a,b,**], Ngoc-Thao Nguyen[a,b,*]

[a] *Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam*
[b] *Vietnam National University, Ho Chi Minh City, Vietnam*

Abstract

RangeViM addresses a core tension in LiDAR semantic segmentation: range-view methods are efficient because they process point clouds as structured 2D images, yet the global context needed for accurate dense prediction has historically demanded architectures too costly to run at full sensor resolution. We present RangeViM, a full-resolution range-view segmentation framework built on TinyViM, a lightweight hybrid backbone that decouples frequency processing by pairing convolutional blocks for local high-frequency detail with state-space blocks for efficient global low-frequency context modeling. This frequency-decoupled design enables direct full-frame inference on the complete 64×2048 range image in a single forward pass, preserving 360° scene context that windowed approaches fragment. We complement the backbone with an asymmetric stride design that keeps full vertical beam resolution throughout all stages, a domain-adapted FPN decoder with horizontal kernels tailored to range-image geometry, and range-image-level augmentation to improve robustness on rare classes. On SemanticKITTI, RangeViM achieves 67.8 mIoU, while on nuScenes it reaches 76.88% mIoU on the validation split. The results suggest that frequency-decoupled hybrid backbones are a practical path toward efficient, full-frame LiDAR segmentation. The code has been made publicly available at this url.

Keywords: LiDAR semantic segmentation; range-view projection; Vision Mamba; autonomous driving

1. Introduction

Range-view LiDAR semantic segmentation projects irregular point clouds into structured 2D range images, enabling efficient dense prediction with image-based backbones. CNN-based methods such as RangeNet++ [22], SalsaNext [7], and FIDNet [27] showed that this representation can achieve competitive accuracy-efficiency trade-offs, but local receptive fields limit long-range azimuthal modeling in full scans.

* Corresponding author.
** These authors contributed equally to this work.
E-mail address: nnthao@fit.hcmus.edu.vn (Ngoc-Thao Nguyen).

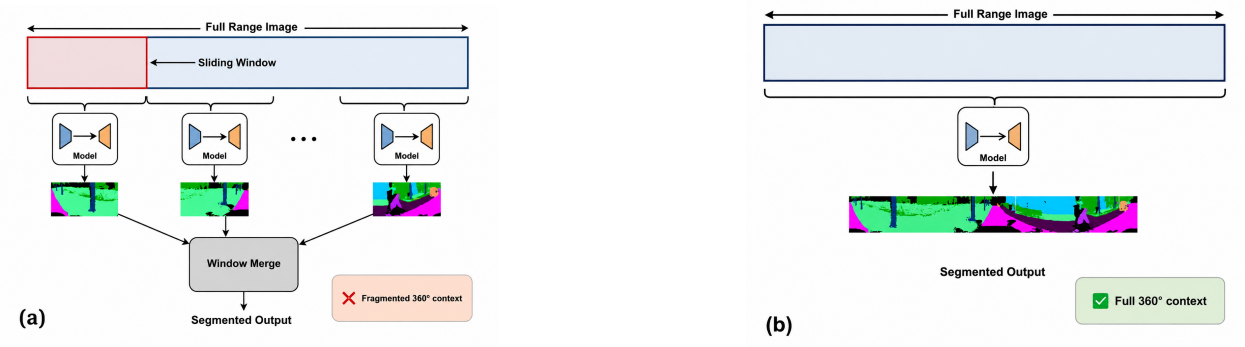


Fig. 1. **Sliding-window (a) vs. full-resolution range-view inference (b).** (a) Transformer-based methods such as RangeViT partition the 64×2048 range image into overlapping crops because full-resolution self-attention is costly; each crop is processed independently and the outputs are merged, which may fragment context across crop boundaries. (b) RangeViM replaces the ViT encoder with the hybrid TinyViM backbone, whose efficient state-space/convolution design makes direct full-resolution processing practical while preserving local detail for dense prediction. **Takeaway:** the hybrid TinyViM backbone supports full-frame inference, allowing the model to process the complete projected scan in one forward pass while preserving local structure.

Transformer-based methods improve broad dependency modeling for range images. RangeViT [1] showed that Vision Transformers can transfer to LiDAR segmentation, while RangeFormer [15] further improved the representation and training recipe. However, self-attention remains costly at full range-image resolution, often motivating cropped or sliding-window inference (Fig. 1a).

Selective state-space models offer a lower-cost alternative for long-range visual modeling. Mamba [10], Vision Mamba [29], MambaVision [12], and TinyViM [20] show the promise of sub-quadratic visual backbones, while RangeRet [23] demonstrates their relevance to range-view LiDAR segmentation.

To this end, we present **RangeViM**, a range-view LiDAR segmentation framework that adapts TinyViM for direct full-frame inference. We combine the hybrid backbone with asymmetric stride design and an FPN decoder adapted to the elongated geometry of range images. This focus is relevant to intelligent transportation systems, where dense LiDAR perception must balance scene-level understanding with memory and latency constraints. Our contributions are:

- We introduce RangeViM, a full-frame range-view LiDAR segmentation framework that leverages frequency-decoupled hybrid modeling to jointly capture local geometric detail and long-range azimuthal context efficiently.
- We propose an anisotropic architecture design for range images, including asymmetric horizontal-only downsampling and a domain-adapted FPN decoder tailored to the geometry of spinning LiDAR sensors.
- We show that lightweight hybrid state-space backbones make full-frame range-image inference substantially more practical than transformer-based sliding-window pipelines while retaining competitive segmentation accuracy.
- We validate the proposed design on SemanticKITTI and nuScenes through comparisons, ablations, efficiency analysis, and robustness sensitivity experiments.

2. Related Work

LiDAR semantic segmentation methods range from direct 3D processing to projection-based 2D representations. We focus on range-view methods and efficient sequence backbones.

2.1. 3D point-, voxel-, and projection-based segmentation

Voxel-based and sparse 3D approaches preserve point-cloud topology. Sparse convolution frameworks such as Minkowski Convolutional Networks [6] support efficient 3D dense prediction, while LiDAR-oriented models includ-

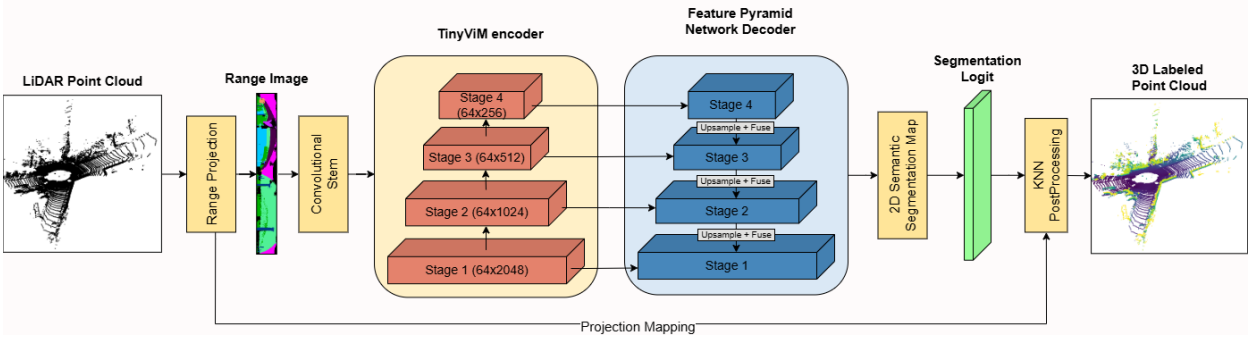


Fig. 2. Overview of RangeViM. A LiDAR point cloud is projected into a 5-channel range image (64×2048), encoded by the TinyViM backbone with asymmetric stride that preserves full vertical resolution, decoded through an FPN with domain-specific horizontal kernels, and re-projected to per-point predictions via KNN post-processing.

ing PolarNet [26] and Cylinder3D [28] introduce spatial partitions suited to outdoor scans. These methods often achieve strong accuracy, but their memory and computation scale with the 3D volume.

Projection-based methods convert LiDAR scans into 2D range images and apply image segmentation networks. RangeNet++ [22] and SalsaNext [7] demonstrate favorable accuracy-efficiency trade-offs, but local convolutional operators can struggle with long-range azimuthal dependencies.

Transformer-based range-view methods such as RangeViT [1] and RangeFormer [15] improve broad dependency modeling, but standard self-attention incurs quadratic cost in the number of tokens. This motivates lightweight sequence backbones that can retain broad context at lower computational cost.

2.2. Mamba-based visual backbones

Mamba [10] introduced selective state-space sequence modeling, and subsequent works extended this idea to vision. Vision Mamba [29] studied pure state-space visual backbones, MambaVision [12] explored hybrid Mamba-Transformer designs, and TinyViM [20] combined state-space blocks with local convolutional blocks in a lightweight hierarchical backbone. We adopt TinyViM because its hybrid structure and feature pyramid naturally support multi-scale range-image decoding.

3. Method

In this section, we present RangeViM, a projection-based LiDAR semantic segmentation architecture built around four components: a range-view representation module, a TinyViM backbone, an FPN [17] decoder, and a point-wise post-processing stage. The overall design follows the projection-based philosophy of RangeViT [1] while replacing the original transformer encoder with a more efficient hierarchical backbone and extending the training recipe with range-image-level augmentations.

3.1. Range-image representation

We represent a LiDAR scan of N points as a matrix $\mathbf{P} \in \mathbb{R}^{N\times 4}$, where each point $\mathbf{p}_n = (x_n, y_n, z_n, i_n)$ contains Cartesian coordinates and reflectance intensity. More generally, range-view LiDAR segmentation maps each point to an image coordinate (u_n, v_n) through a projection operator. A common formulation is spherical projection:

$$\begin{aligned} u_n &= \frac{1}{2}\left[1 - \arctan(y_n, x_n)\pi^{-1}\right] W, \\ v_n &= \left[1 - \left(\arcsin(z_n r_n^{-1}) + |f^{\text{down}}|\right) f^{-1}\right] H. \end{aligned} \tag{1}$$

where $r_n = \sqrt{x_n^2 + y_n^2 + z_n^2}$ is the range and $f = |f^{\text{up}}| + |f^{\text{down}}|$ is the vertical field of view, and H and W are the height and width of the range image, respectively. In the experiments reported in this paper, we use the scan-unfolded range-image projection setting, which preserves the native scan ordering of the sensor and produces a 64×2048 range image for SemanticKITTI [2]. The projected tensor uses five channels encoding (r, x, y, z, i). Each channel is normalized by dataset statistics and invalid cells are masked out. When multiple points project to the same image cell, points are processed in decreasing depth order, so nearer points overwrite farther points; the projected pixel and label therefore correspond to the closest point in range.

Compared with raw point-cloud processing, the range-view representation offers two main advantages. First, it converts irregular LiDAR data into a dense tensor that can be processed efficiently using 2D operators. Second, the horizontal axis preserves the azimuthal ordering of the scan, which makes long-range context modeling important along the width dimension.

3.2. TinyViM backbone

A key limitation of RangeViT [1] is its reliance on a plain Vision Transformer encoder with quadratic self-attention cost. At LiDAR range-image resolution, this cost makes full-resolution processing difficult and often leads to cropped or sliding-window inference. Moreover, the original ViT-style encoder produces a single-scale output, which restricts the decoder's ability to recover fine spatial structure.

We replace the transformer encoder with TinyViM [20], a lightweight hybrid backbone that interleaves mobile-friendly convolutional blocks with selective state-space blocks. This choice is motivated by the needs of dense range-view prediction: broad horizontal context is useful for scene consistency, while local geometric detail is important for object boundaries and thin structures. This local-global complementarity supports dense LiDAR prediction while keeping the backbone lightweight. TinyViM's hierarchical outputs also provide a natural interface for multi-scale decoding.

3.3. Asymmetric stride design

Range images exhibit strong anisotropy: the vertical dimension corresponds to a small set of fixed LiDAR beams, while the horizontal dimension densely samples the azimuthal sweep. As a result, the two axes have different structural properties. Early symmetric downsampling can unnecessarily discard beam-aligned geometric information, whereas the horizontal dimension contains greater redundancy and longer-range contextual dependencies.

To preserve beam-level structure while reducing computation, we maintain full vertical resolution throughout the encoder and apply downsampling only along the horizontal axis. Concretely, the adapted stem uses unit stride and all stage transitions reduce only the width dimension. This design preserves the correspondence between feature rows and physical LiDAR beams while progressively expanding the horizontal receptive field.

Table 1. RangeViM TinyViM-Base encoder and FPN decoder details. The model preserves the 64-beam vertical resolution and downsamples only along the azimuth dimension after the stem.

Stage	Blocks	Channel	Output	Decoder refinement
0	4	48	64×2048	1×1 lateral; 3×3 FPN/head
1	3	96	64×1024	1×1 lateral; 3×3 FPN/head
2	10	192	64×512	1×1 lateral; 1×5 FPN/head
3	5	384	64×256	1×1 lateral; 1×5 FPN/head

The deeper stages use wider horizontal kernels because the horizontal context becomes increasingly important after azimuthal downsampling. The FPN projects all stage features to 256 channels, refines them using the listed kernels, upsamples them to the first-stage resolution, and fuses them through summation followed by a 3×3 convolution, dropout 0.1, and a 1×1 classifier.

3.4. FPN decoder

The hierarchical nature of TinyViM motivates the use of a feature pyramid decoder [17]. We therefore replace the original single-scale decoding strategy with an FPN that performs lateral projection, top-down feature fusion, per-stage refinement, and full-resolution fusion before classification. This decoder is designed to exploit all backbone stages and recover thin structures and fine boundaries in the projected range image.

More specifically, each stage output is first projected to a common channel dimension through a lateral 1×1 convolution. Starting from the deepest stage, a top-down path progressively upsamples and merges features with shallower stages. The refined stage features are then projected to a common head dimension, upsampled to the original range-image resolution, and fused before the final classifier.

This decoder is well suited to the proposed backbone for two reasons. First, it directly leverages the hierarchical outputs produced by TinyViM rather than discarding them in favor of a single-scale representation. Second, it improves the recovery of thin structures and rare small objects, which are often poorly represented when segmentation relies only on the deepest feature map.

3.5. Range-image-level augmentations

To support generalization and mitigate class imbalance, we incorporate a suite of range-image-level augmentations applied directly to projected tensors before the forward pass. These include azimuthal region mixing, beam-wise mixing, hole completion, and tail-class range-pixel pasting [15]. In our work, these operations are used as part of the training recipe for the proposed full-frame TinyViM-based segmentation pipeline rather than as a standalone augmentation contribution. Because they operate on the structured range image, they can be implemented efficiently on the GPU and integrated into the standard training loop with low overhead.

Unlike conventional point-level augmentation, these operations modify the range-view tensor after projection. This choice is practical in two ways. First, it makes the augmentations lightweight and easy to batch on the GPU. Second, it allows the training process to directly perturb the representation seen by the 2D backbone and decoder. In our setting, this is particularly useful for rare or small classes such as bicycles, motorcyclists, and traffic signs, which otherwise occupy very few pixels.

4. Experiments

4.1. Datasets and Metrics

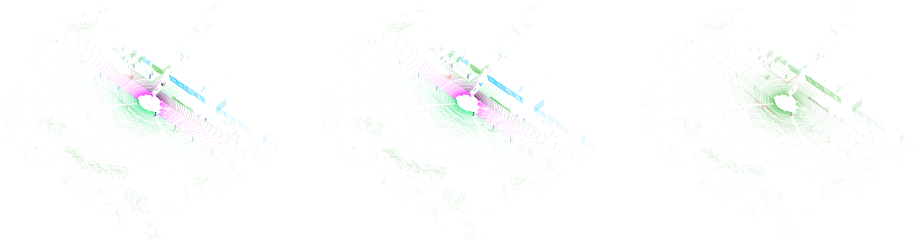
SemanticKITTI [2]. The majority of our experiments are conducted on SemanticKITTI. Derived from the KITTI Odometry sequences [9], this dataset provides dense point-wise labels for 64-beam LiDAR scans in urban driving scenes. We follow the standard protocol, training on sequences 00–10 with sequence 08 used for validation, and report results on the official 19 semantic classes used in the single-scan challenge.

nuScenes [4]. To evaluate transfer across different sensors and environments, we also use the nuScenes dataset. In contrast to KITTI, nuScenes uses a 32-beam LiDAR and covers diverse scenes across Boston and Singapore. We follow the official train/validation split and evaluate across the 16 semantic categories defined for lidar segmentation.

We report the standard mean intersection-over-union (mIoU) [8] metric for semantic segmentation. Given the strong class imbalance in both SemanticKITTI and nuScenes, mIoU provides a more representative evaluation than global accuracy.

4.2. Implementation details

Unless otherwise stated, our main SemanticKITTI experiments use the configuration implemented in the official public repository for the reported RangeViM variant. The LiDAR scan is projected into a scan-unfolded range image of size 64×2048 with five input channels (r, x, y, z, i). Training and inference use the same full-frame spatial resolution, and sliding-window inference is disabled for the main RangeViM setting.



(a) Ground Truth (b) RangeViM Prediction (c) Error Map

Fig. 3. Qualitative BEV visualization on SemanticKITTI sequence 08, frame 219.(a) Ground truth semantic labels. (b) RangeViM predictions. (c) Prediction error map: green indicates correct predictions, red indicates misclassifications.

For the backbone, we use the Base variant of TinyViM [20] together with an FPN [17] decoder. In the current implementation, the encoder uses TinyViM's native two-layer convolutional stem rather than the original RangeViT convolutional stem. The stem outputs 48 channels, after which the hierarchical backbone preserves the full vertical resolution and progressively downsamples only the horizontal dimension. The decoder operates without the skip-connection branch used by the original up-convolution design. For SemanticKITTI, the backbone receives full-resolution range images of size 64×2048, while the nuScenes variant uses a 32×2048 projection to match the 32-beam sensor geometry.

We optimize the network with AdamW [19] and a warmup cosine learning-rate schedule. Unless otherwise stated, all experiments and profiling are conducted on NVIDIA A100 GPUs. On SemanticKITTI, the main RangeViM configuration is trained for 60 epochs with a batch size of 6, a peak learning rate of 3×10^{-4}, and 6 warmup epochs. Mixed-precision training is enabled. For the nuScenes configuration, we use 120 epochs, batch size 4, peak learning rate 3×10^{-4}, and 10 warmup epochs. We use an equal combination of focal loss [18] and Lovász-softmax loss [3]. The focal term reduces the influence of well-classified pixels to handle class imbalance, while the Lovász-softmax term acts as a differentiable surrogate for IoU to improve region-level mask quality.

For point-wise prediction recovery, we use KNN-based post-processing [14] rather than KPConv [25] in the main RangeViM configuration. Each 3D point is refined by aggregating neighboring range-image predictions in a local window using a distance-aware weighting scheme, which helps mitigate discretization artifacts. Specifically, the KNN module uses a search window of 7, $k = 5$ neighbors, Gaussian weighting with $\sigma = 1.0$, and cutoff 1.0.

4.3. Main results

Table 2 compares RangeViM against representative point-based, voxel-based, and projection-based methods on the SemanticKITTI test set (single-scan). Baseline numbers are reproduced from published papers or official benchmark reports, and rows are grouped by input representation to make the comparison with stronger 3D methods explicit.

RangeViM achieves 67.8% mIoU, a +3.8 pp improvement over RangeViT [1]. This result is competitive with strong 3D voxel methods such as Cylinder3D [28], while WaffleIron [24] and SphereFormer [16] show the remaining advantage of high-capacity point and sparse 3D transformer modeling. Notably, RangeViM obtains this result with direct full-frame inference on the full 64×2048 range image, without sliding-window decomposition at this resolution. Efficiency is analyzed in Section 4.4.3.

Within the retained projection-based rows, RangeViM obtains the highest IoU on several thin structures and rare objects, including *pedestrian* (71.5%, +7.6 pp over RangeViT), *pole* (63.2%), *fence* (71.3%), *building* (92.9%), *other-ground* (35.6%), and *traffic-sign* (68.3%). These classes involve elongated or fine-grained geometry in the range image, where TinyViM's hierarchical multi-scale features and the FPN decoder's horizontal kernels may be beneficial. On broad surface classes such as *road* and *sidewalk*, RangeViM remains competitive compared to other methods.

The qualitative error map in Fig. 3 shows that large static regions are segmented consistently, while remaining errors tend to appear near object boundaries and sparse small objects. This behavior is consistent with the range-view formulation: thin structures may occupy only a few projected pixels, and nearby objects can become difficult to

Table 2. SemanticKITTI test (single-scan) class-wise IoU (%) across point-, voxel-, and projection-based methods. Baseline numbers are reproduced from the corresponding published papers or official benchmark reports. Within projection-based rows only, **bold** and underline indicate the best and second-best results per column, respectively.

Method	Representation	mIoU	car	bicy	moto	truc	o.veh	ped	bicl.	m.lst	road	park	sidew	o.gro	build	fence	veget	trun	terr	pole	sign
RandLA-Net [13]	Point-based	53.9	94.2	26.0	25.8	40.1	38.9	49.2	48.2	7.2	90.7	60.3	73.7	20.4	86.9	56.3	81.4	61.3	66.8	49.2	47.7
KPConv [25]	Point-based	58.8	96.0	30.2	42.5	33.4	44.3	61.5	61.6	11.8	88.8	61.3	72.7	31.6	90.5	64.2	84.8	69.2	69.1	56.4	47.4
WaffleIron [24]	Point-based	70.8	97.2	70.0	69.8	40.4	59.6	77.1	75.5	41.5	90.6	70.4	76.4	38.9	93.5	72.3	86.7	75.7	71.7	66.2	71.9
PolarNet [26]	Voxel-based	54.3	93.8	40.3	30.1	22.9	28.5	43.2	40.2	5.6	90.8	61.7	74.4	21.7	90.0	61.3	84.0	65.5	67.8	51.8	57.5
Cylinder3D [28]	Voxel-based	67.8	97.1	67.6	64.0	50.8	58.6	73.9	67.9	36.0	91.4	65.1	75.5	32.3	91.0	66.5	85.4	71.8	68.5	62.6	65.6
SphereFormer [16]	Voxel-based	74.8	97.5	70.1	70.5	59.6	67.7	79.0	80.4	75.3	91.8	69.7	78.2	41.3	93.8	72.8	86.7	75.1	72.4	66.8	72.9
RangeViT [1]	Projection-based	64.0	95.4	55.8	43.5	29.8	42.1	63.9	58.2	38.1	**93.1**	70.2	**80.0**	32.5	92.0	69.0	85.3	**70.6**	**71.2**	60.8	64.7
RangeRet [23]	Projection-based	64.5	93.5	50.6	55.4	51.3	**52.9**	67.3	**66.3**	27.3	90.3	68.5	73.5	32.0	90.1	63.3	84.5	65.6	70.4	56.0	65.8
CENet [5]	Projection-based	64.7	91.9	**58.6**	50.3	40.6	42.3	68.9	65.9	43.5	90.3	60.9	75.1	31.5	91.0	66.2	84.5	69.7	70.0	61.5	67.6
MaskRange [11]	Projection-based	66.1	94.2	56.0	**55.7**	**59.2**	52.4	67.6	64.8	31.8	91.7	70.7	77.1	29.5	90.6	65.2	84.6	68.5	69.2	60.2	66.6
RangeViM (ours)	Projection-based	**67.8**	**95.9**	55.5	50.8	40.9	51.8	**71.5**	64.2	**52.7**	**93.1**	**73.4**	**80.0**	**35.6**	**92.9**	**71.3**	**85.6**	70.0	70.8	**63.2**	**68.3**

Table 3. nuScenes validation class-wise IoU (%) across point-, voxel-, and projection-based methods. Baseline numbers are reproduced from the corresponding published papers. Within projection-based rows only, **bold** and underline indicate the best and second-best results per column, respectively.

Method	Representation	mIoU	barr.	bicy.	bus	car	constr.	moto.	ped.	cone	trail.	truck	driv.	o.flat	sidew.	terr.	manm.	veg.
RandLA-Net [13]	Point-based	62.9	72.5	12.6	36.6	81.8	38.7	72.3	68.5	37.3	44.7	59.7	95.3	87.0	69.7	71.1	73.2	85.9
PointMLP [21]	Point-based	67.9	72.3	27.8	88.2	86.3	37.2	51.0	60.7	50.6	56.4	71.1	95.7	70.6	70.9	72.0	88.8	87.2
WaffleIron [24]	Point-based	77.6	78.7	51.3	93.6	88.2	47.2	86.5	81.7	68.9	69.3	83.1	96.9	74.3	75.6	74.2	87.2	85.2
PolarNet [26]	Voxel-based	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
Cylinder3D [28]	Voxel-based	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
SphereFormer [16]	Voxel-based	78.4	77.7	43.8	94.5	93.1	52.4	86.9	81.2	65.4	73.4	85.3	97.0	73.4	75.4	75.0	91.0	89.2
RangeNet++ [22]	Projection-based	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
SalsaNext [7]	Projection-based	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
RangeViT [1]	Projection-based	75.2	**75.5**	40.7	88.3	**90.1**	49.3	79.3	**77.2**	**66.3**	65.2	80.0	96.4	71.4	73.8	73.8	**89.9**	**87.2**
RangeViM (ours)	Projection-based	**76.88**	74.80	**45.68**	**93.77**	90.05	**54.19**	**83.36**	74.66	64.80	**72.30**	**83.76**	**96.46**	**71.87**	**75.41**	**74.00**	88.53	86.36

separate after projection and KNN re-projection. These observations complement the class-wise results and motivate future boundary-aware or point-level refinement.

Table 3 reports the corresponding nuScenes validation results together with representative point-based and voxel-based baselines. Since the main objective of this work is to improve the projection-based range-view pipeline, the most direct comparison is against other projection-based methods. On this dataset, RangeViM reaches **76.88%** mIoU, improving over RangeViT [1] by +1.68 pp in mIoU and ranking above the other projection-based baselines listed in the table. Strong 3D methods such as WaffleIron [24] and SphereFormer [16] remain higher in overall mIoU, reflecting the accuracy advantage of direct point or sparse 3D processing. Nevertheless, the gap between these 3D methods is reduced compared to earlier range-view baselines, suggesting that a full-frame TinyViM-FPN design can move projection-based segmentation closer to point- and voxel-based performance while retaining the simplicity and efficiency of a 2D range-image representation. Within the projection-based group listed here, the largest differences appear on several object classes, including *bicycle* (45.68% vs. 40.7%), *bus* (93.77% vs. 88.3%), *construction vehicle* (54.19% vs. 49.3%), *motorcycle* (83.36% vs. 79.3%), *trailer* (72.30% vs. 65.2%), and *truck* (83.76% vs. 80.0%). These results indicate that the same full-frame TinyViM-FPN design transfers from the 64-beam SemanticKITTI setting to the 32-beam nuScenes sensor while retaining competitive fine-grained recognition.

4.4. Ablation studies

4.4.1. Backbone capacity

Table 4 summarizes the backbone capacity ablation on the SemanticKITTI validation set, comparing TinyViM-Small, TinyViM-Base, and TinyViM-Large while also including the RangeViT baseline as an efficiency reference. The table reports point-level mIoU together with model size and computational cost so that the accuracy-efficiency trade-off across capacities can be analyzed directly. Among the evaluated TinyViM variants, performance improves from 65.00% mIoU for TinyViM-Small to 65.96% for TinyViM-Base, indicating that the additional capacity is beneficial up to the Base setting. TinyViM-Large reaches 65.07%, suggesting that simply increasing capacity is not sufficient under the same training and post-processing recipe. This makes TinyViM-Base the most balanced operating point

among the evaluated configurations. The latency values should be interpreted as backbone-decoder inference profiling rather than end-to-end deployment latency;

Table 4. Backbone capacity and efficiency comparison on the SemanticKITTI validation setting. Parameters, GFLOPs, latency, and peak inference VRAM are profiled for network inference on projected tensors on an NVIDIA A100 GPU; Latency is reported as mean ± standard deviation.

Method	mIoU (%)	Params (M)	GFLOPs	Latency (ms)	Peak VRAM (MB)
RangeViT	60.80	26.99	1873.80	97.97±0.24	995.76
TinyViM-Small	65.00	8.49	726.01	32.51±3.33	1032.42
TinyViM-Base	65.96	13.83	952.47	38.50±3.79	1083.79
TinyViM-Large	65.07	24.28	1415.38	39.94±0.25	1174.71

4.4.2. Component

Table 5 isolates two components under the same TinyViM-Base full-frame setting. Removing range-image augmentation lowers point mIoU from 65.96% to 63.18%. Replacing FPN with a simple decoder that upsamples TinyViM stage features, concatenates them by channel, and applies two convolutional refinements, gives 64.76%; FPN is 1.20 points higher, suggesting a modest benefit from structured top-down fusion.

Table 5. Component ablation on the SemanticKITTI validation set. Results are point-level mIoU with TinyViM-Base and full-frame inference under the same post-processing setting.

Setting	Decoder	Point mIoU (%)
Full model with range-image augmentation	FPN	65.96
Without range-image augmentation	FPN	63.18
Full model with range-image augmentation	Simple fusion	64.76

4.4.3. Window size

Table 6 evaluates whether the proposed backbone benefits from processing the complete range image at once rather than using reduced-width sliding windows. We compare the full 64×2048 setting with 64×1024 and 64×512 windows under both non-overlapping and overlapping inference.

The full-frame setting achieves the best validation mIoU, indicating that preserving the complete horizontal field of view is beneficial for range-view segmentation. Non-overlapping windows substantially reduce accuracy because each crop is processed independently and object or scene context can be truncated at crop boundaries. Overlapping windows partially recover this loss for the 64×1024 setting, but they require more forward passes and therefore increase latency and memory use. The 64×512 setting remains worse even with overlap, suggesting that very narrow crops provide insufficient context for prediction.

Table 6. Effect of window size on the SemanticKITTI validation set. mIoU is point-level after KNN post-processing. Profiling uses validation-style full-scan execution with AMP on a shared A100 server and excludes projection and KNN post-processing. Latency is reported as mean ± standard deviation and should be interpreted as indicative.

Window	Mode	Stride	mIoU (%)	Params (M)	GFLOPs	Latency (ms)	VRAM (MiB)
64×2048	Full frame	–	65.96	13.83	952.47	38.50±3.79	1083.79
64×1024	Non-overlap	1024	62.38	13.83	952.47	42.10±4.03	1086.04
64×1024	Overlap	512	65.48	13.83	1428.71	76.77±5.11	1634.64
64×512	Non-overlap	512	62.33	13.83	952.47	40.37±4.42	1086.04
64×512	Overlap	256	62.61	13.83	1666.83	104.00±28.71	1910.64

4.4.4. Robustness sensitivity

To test sensitivity under degraded inputs, we apply validation-time range-image corruptions without retraining. Corruptions are applied only to normalized input features; labels and KNN re-projection geometry are unchanged. Thus, this is a controlled input-sensitivity analysis rather than a physical adverse-weather benchmark. Projected-pixel and beam dropout reduce mIoU by 0.59 and 0.96 points, respectively, while normalized coordinate noise causes a larger 3.95-point drop. This suggests that RangeViM is more sensitive to incoherent geometric feature perturbations than to sparse missing projected evidence.

Table 7. Validation-time robustness sensitivity analysis on SemanticKITTI. Results are point-level mIoU averaged over three deterministic seeds. Corruptions are applied only to normalized range-image input features without retraining; labels and KNN re-projection geometry are kept unchanged.

Validation setting	Point mIoU (%)
Clean	65.96
Projected-pixel dropout ($p = 0.10$)	65.37 ± 0.01
Beam dropout ($p = 0.10$)	65.00 ± 0.09
Normalized range-coordinate noise ($\sigma = 0.03$)	62.01 ± 0.05

5. Discussion

The results indicate that RangeViM improves the accuracy-efficiency trade-off of several projection-based baselines, but they should be interpreted in the broader context of 3D point and voxel methods. Strong 3D models such as WaffleIron and SphereFormer still benefit from operating more directly on 3D geometry. Thus, the main advantage of RangeViM is not absolute dominance over all representations, but a practical balance between accuracy, memory cost, and complete-scan range-view inference. The gains over RangeViT suggest that the TinyViM-FPN combination is effective for dense range-view prediction: state-space blocks provide broad context, convolutional blocks preserve local structure, and the FPN decoder reuses hierarchical features for multi-scale prediction. The augmentation ablation further shows that perturbing the projected representation improves mIoU under the same post-processing setting.

Limitations and Future Work. RangeViM still inherits limitations from the range-view formulation. Projection can collapse multiple 3D points into the same range-image cell, and final point-wise predictions remain sensitive to KNN post-processing after re-projection. Likely failure cases include object boundaries, adjacent objects at similar azimuths, rare moving classes, and thin structures that occupy only a few projected pixels. The current robustness evidence is limited to cross-dataset transfer and controlled validation behavior; more complete evaluation under adverse weather, sensor noise, and missing returns remains necessary. The efficiency results also reflect profiling on an NVIDIA A100 GPU rather than embedded automotive hardware. Therefore, the current measurements support the relative computational efficiency of the architecture, but real-time deployment claims should be validated on deployment-oriented accelerators. Future work will investigate temporal or multi-scan fusion, stronger 2D–3D feature coupling, robustness under corrupted LiDAR inputs, and embedded-system profiling.

6. Conclusion

This paper presented RangeViM, a full-resolution range-view LiDAR semantic segmentation framework based on a lightweight TinyViM backbone and an FPN decoder with domain-specific horizontal kernels. By replacing the heavier transformer encoder used in RangeViT, the proposed design makes direct processing of the complete 64×2048 projected scan feasible, improving the accuracy-efficiency trade-off in our evaluated setting. On SemanticKITTI, RangeViM achieves 67.8% mIoU, while on nuScenes it reaches 76.88% mIoU on the validation split, suggesting that the same full-frame design can transfer across different LiDAR configurations. These results, together with the efficiency and inference ablations, suggest that lightweight state-space-inspired backbones are a promising direction for efficient full-frame LiDAR segmentation, and we plan to extend this approach to multi-scan temporal fusion and additional sensor configurations in future work.

Acknowledgement

This research is funded by University of Science, VNU-HCM under grant number CNTT 2025-34.

References

[1] Ando, A., Gidaris, S., Bursuc, A., Puy, G., Boulch, A., Marlet, R., 2023. Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5240–5250.

[2] Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J., 2019. Semantickitti: A dataset for semantic scene understanding of lidar sequences, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

[3] Berman, M., Triki, A.R., Blaschko, M.B., 2018. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[4] Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O., 2020. nuscenes: A multimodal dataset for autonomous driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[5] Cheng, H.X., Han, X.F., Xiao, G.Q., 2022. Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving, in: 2022 IEEE international conference on multimedia and expo (ICME), IEEE. pp. 01–06.

[6] Choy, C., Gwak, J., Savarese, S., 2019. 4d spatio-temporal convnets: Minkowski convolutional neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[7] Cortinhal, T., Tzelepis, G., Erdal Aksoy, E., 2020. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds, in: Bebis, G., Yin, Z., Kim, E., Bender, J., Subr, K., Kwon, B.C., Zhao, J., Kalkofen, D., Baciu, G. (Eds.), Advances in Visual Computing, Springer International Publishing, Cham. pp. 207–222.

[8] Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2015. The pascal visual object classes challenge: A retrospective. International journal of computer vision 111, 98–136.

[9] Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the kitti vision benchmark suite, in: 2012 IEEE conference on computer vision and pattern recognition, IEEE. pp. 3354–3361.

[10] Gu, A., Dao, T., 2023. Mamba: Linear-time sequence modeling with selective state spaces. ArXiv abs/2312.00752.

[11] Gu, Y., Huang, Y., Xu, C., Kong, H., 2022. Maskrange: A mask-classification model for range-view based lidar segmentation.

[12] Hatamizadeh, A., Kautz, J., 2025. Mambavision: A hybrid mamba-transformer vision backbone, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 25261–25270.

[13] Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2020. Randla-net: Efficient semantic segmentation of large-scale point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11108–11117.

[14] Kochanov, D., Karimi Nejadasl, F., Booij, O., 2020. Kprnet: Improving projection-based lidar semantic segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV).

[15] Kong, L., Liu, Y., Chen, R., Ma, Y., Zhu, X., Li, Y., Hou, Y., Qiao, Y., Liu, Z., 2023. Rethinking range view representation for lidar segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 228–240.

[16] Lai, X., Chen, Y., Lu, F., Liu, J., Jia, J., 2023. Spherical transformer for lidar-based 3d recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17545–17555.

[17] Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017a. Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[18] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017b. Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988.

[19] Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization, in: International Conference on Learning Representations (ICLR).

[20] Ma, X., Ni, Z., Chen, X., 2025. Tinyvim: Frequency decoupling for tiny hybrid vision mamba, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 23519–23529.

[21] Ma, X., Qin, C., You, H., Ran, H., Fu, Y., 2022. Rethinking network design and local geometry in point cloud: A simple residual mlp framework, in: International Conference on Learning Representations (ICLR).

[22] Milioto, A., Vizzo, I., Behley, J., Stachniss, C., 2019. Rangenet ++: Fast and accurate lidar semantic segmentation, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4213–4220.

[23] Mosco, S., Fusaro, D., Li, W., Pretto, A., 2026. Revisiting retentive networks for fast range-view 3d lidar semantic segmentation, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 2499–2509.

[24] Puy, G., Boulch, A., Marlet, R., 2023. Using a waffle iron for automotive point cloud semantic segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3379–3389.

[25] Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. Kpconv: Flexible and deformable convolution for point clouds, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6411–6420.

[26] Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., Foroosh, H., 2020. 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 (CVPR).

[27] Zhao, Y., Bai, L., Huang, X., 2021. Fidnet: Lidar point cloud semantic segmentation with fully interpolation decoding, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4453–4458.

[28] Zhou, H., Zhu, X., Song, X., Ma, Y., Wang, Z., Li, H., Lin, D., 2020. Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation. ArXiv abs/2008.01550.

[29] Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X., 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model, in: International Conference on Machine Learning.