This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

FLARES: Fast and Accurate LiDAR Multi-Range Semantic Segmentation

Bin Yang[1,2] Alexandru Paul Condurache[1,2]
[1]Automated Driving Research, Robert Bosch GmbH
[2]Institute for Signal Processing, University of Lübeck
{Bin.Yang3, AlexandruPaul.Condurache}@de.bosch.com

Abstract

3D scene understanding is a critical yet challenging task in autonomous driving due to the irregularity and sparsity of LiDAR data, as well as the computational demands of processing large-scale point clouds. Recent methods leverage range-view representations to enhance efficiency, but they often adopt higher azimuth resolutions to mitigate information loss during spherical projection, where only the closest point is retained for each 2D grid. However, processing wide panoramic range-view images remains inefficient and may introduce additional distortions. Our empirical analysis shows that training with multiple range images, obtained from splitting the full point cloud, improves both segmentation accuracy and computational efficiency. However, this approach also poses new challenges of exacerbated class imbalance and increase in projection artifacts. To address these, we introduce FLARES, a novel training paradigm that incorporates two tailored data augmentation techniques and a specialized post-processing method designed for multi-range settings. Extensive experiments demonstrate that FLARES is highly generalizable across different architectures, yielding 2.1%–7.9% mIoU improvements on SemanticKITTI and 1.8%–3.9% mIoU on nuScenes, while delivering over 40% speed-up in inference.[1]

1. Introduction

LiDAR is one of the most common sensors for perception in autonomous driving. Semantic segmentation on LiDAR point clouds is essential for getting useful and reliable information of the surrounding 3D environment. To solve this 3D scene understanding task, many prior works propose to integrate deep learning techniques because of its remarkable advancements in the past few years. The publication of various annotated datasets [13, 14, 34] in the domain of autonomous driving further promotes research in the field.

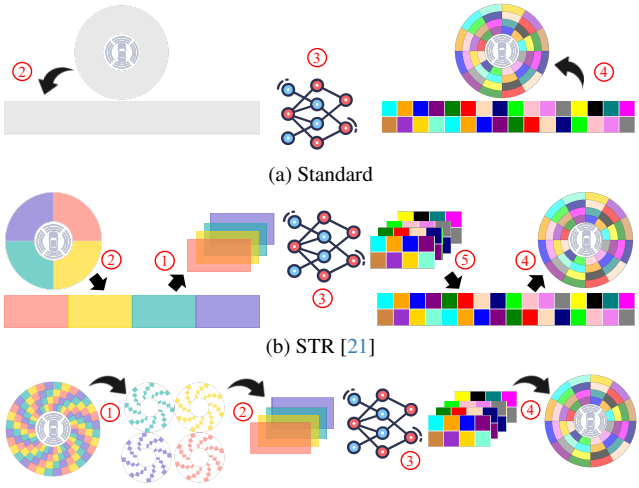


Figure 1. Visual comparison among different training procedures for range-view LiDAR semantic segmentation: ① Splitting, ② Range-view projection, ③ Network prediction, ④ Post-processing, ⑤ Image concatenation.

In general, those methods can be categorized based on LiDAR data representation into point-based [30, 42, 51], voxel-based [17, 53] and projection-based methods [6, 28, 50, 52]. Both point and voxel-based approaches typically require substantial computational resources due to the need to process data through networks with numerous 3D convolutional layers, intensive feature pre-processing, and deep architectures involving multiple downsampling and upsampling operations. These requirements can result in slow inference speeds, limiting their suitability for real-time applications [25]. In contrast, rasterizing point cloud into range-view images [8] is more advantageous in fast and scalable LiDAR perception, because it allows the use of 2D operators for efficient computation and facilitates the transfer of knowledge from camera images [1, 21].
Nevertheless, learning from range-view representations can suffer from a "many-to-one" conflict among adjacent points. This issue causes an irreversible loss of spatial information and leads to inferior performance compared to other methods that utilize 3D LiDAR representations. Al-

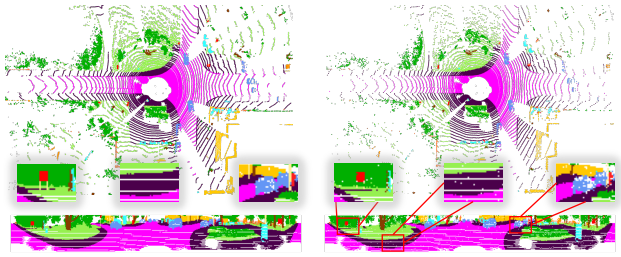
[1]Project page: https://binyang97.github.io/FLARES

though increasing the azimuth resolution of range images can alleviate this loss, it forces the network to process wide panoramic images (Fig. 1a), which significantly increases computational overhead [6, 8, 16, 21, 28, 52]. Building on this standard setting, recent work introduces a scalable training framework with range-view images (STR) to improve efficiency [21]. As shown in Fig. 1b, by splitting the point clouds into multiple distinct views, STR processes all crops in a batch during inference. The resulting predictions are then merged to reconstruct the full range image, which is subsequently post-processed to generate 3D coordinates. However, the splitting strategy of STR is heuristic and compromises scene completeness. Empirical results in [21] reveal a slight performance drop within some networks. Furthermore, the gain in inference speed is limited when maintaining the high resolution of range images during the projection and post-processing stages. To solve these challenges, we propose *FLARES*, a novel training paradigm for range-view LiDAR semantic segmentation. Its procedure is visually illustrated in Fig. 1c. In contrast to cartesian grid splits of STR [21], *FLARES* divides point clouds along the LiDAR's spherical coordinate and then projects each split into lower-resolution range images. This design preserves **local geometric coherence** and **scene integrity** during partitioning, which enables **more effective semantic context sharing** across generated range images compared to STR. Furthermore, our method exclusively processes low-resolution images, achieving **greater efficiency** than both STR and standard settings.

Moreover, multi low-resolution range-view images created by *FLARES* are generally a more informative and powerful representation compared to the high-resolution projection. Due to ego-motion and sensor calibration error, points mapped to identical azimuthal positions often misalign with actual laser beam orientations [49]. This inherent mismatch demonstrates that jointly prioritizing azimuth and elevation resolutions, rather than solely expanding image width, achieves higher projection fidelity. Statistically, as quantified in our occupancy measurement (Fig. 2), doubling the elevation resolution achieves approximately 80% of the projection gain obtained by doubling the azimuth resolution in SemanticKITTI [2] dataset. This property indicates that our new representation can restore more 3D points, thereby alleviating the "many-to-one" conflict among adjacent points more effectively.



Figure 2. Statistics on SemanticKITTI [2]: 3D validity (proportion of projected points) with different azimuth (**W**) and elevation (**H**) resolutions. Comparable increases are observable when doubling azimuth and elevation resolution ($\Delta V_{azi}, \Delta V_{ele}$).



(a) Before splitting (b) After splitting

Figure 3. An example from SemanticKITTI [2] dataset is visualized in top-down view and range-view. Three crops are magnified to specify *three limitations* brought by the splitting of point clouds. **1) Exacerbated class imbalance**: objects with small sizes, likewise low occurrences in annotations, tend to fade away in the range image after downsampling (e.g. pole in the left crop). **2) Intensified noise**: reduction in points results in increasing clutters (middle crop) that may disrupt the training stability. **3) Deteriorated distortion**: decrease in point density can introduce more projection artifacts, thus corrupting the sharpness of local geometry (e.g. blurry boundaries of car in the right crop).

While *FLARES* offers significant advantages, it introduces three key challenges, as visualized in Fig. 3. First, point cloud splitting exacerbates class imbalance issues [5], where classes representing small objects with inherently sparse distributions become further undersampled. Second, the splitting operation increases point cloud sparsity, causing substantial reductions in occupancy of range images. These unoccupied pixels act as noise that destabilize training convergence [21]. Finally, the resulting projection artifacts degrade image sharpness, particularly at partition boundaries where geometric discontinuities emerge.

To tackle these incidental issues, we extend the pipeline with two data augmentation steps. Furthermore, we investigate in exploring a more effective post-processing technique tailored to the multi-range settings to further enhance the advantages of our method.

In summary, our contributions are:

1. *FLARES*, a scalable training paradigm that enhances LiDAR semantic segmentation with improved speed, accuracy, and architectural agnosticism.
2. Two specialized data augmentation techniques addressing exacerbated class imbalance and noise amplification within *FLARES*.
3. A novel interpolation-based post-processing method that refines range-view predictions using precise nearest-neighbor information.
4. Comprehensive experiments across four network architectures and two benchmarks demonstrating consistent superiority over baselines.

2. Related Works

Point and Voxel-based methods Some recent works [30, 41, 51] use raw point cloud data as direct network input, eliminating the need for post-processing after prediction. However, these methods often face high computational complexity and memory usage. To address these issues, Hu et al. [20] introduced sub-sampling and feature aggregation techniques for large-scale point clouds to reduce computational costs. Despite these efforts, performance degradation remains significant. Other works [22, 53] use 3D voxel grids as input, achieving point-based accuracy with reduced computational costs by utilizing sparse 3D convolutions [7]. Nonetheless, voxelization and de-voxelization steps continue to be time- and memory-intensive.

Range-view-based methods To address inefficiencies, some prior works [28, 39, 40] convert the large-scale point cloud to panoramic range image through spherical projection and leverage image segmentation techniques for LiDAR data. SalsaNext [8] uses a Unet-like network with dilated convolutions to broaden receptive fields for more accurate segmentation, while Lite-HDSeg [32] introduces an efficient framework using a lite version of harmonic convolutions. Additionally, FIDNet [52] and CENet [6] interpolate and concatenate multi-scale features with a minimal decoder for semantic prediction. These methods share the benefit of lightweight network design, significantly improving efficiency and enabling real-time applications. Nevertheless, they generally underperform 3D methods due to the "many-to-one" issue, where multiple points project to the same pixel. To offset the performance drop caused by the problem, some other recent works propose to use Vision Transformer (ViT) [10, 38, 44]. RangeViT [1] deploys standard ViT backbone as encoder, followed by a light-weight decoder for refining the coarse patch-wise ViT representations, while RangeFormer [21] utilizes a pyramid-wise ViT-encoder to extract multi-scale features from range images. ViTs offer higher model capacities and excel at capturing long-range dependencies by modeling global interactions between different regions, enhancing segmentation performance over traditional CNNs [9]. However, the quadratic computational complexity of self-attention mechanisms in ViTs introduces challenges in achieving an optimal balance between efficiency and accuracy.

Training Paradigm While existing methods address inefficiencies in high-resolution range images by employing compact networks to reduce model capacity [6, 8, 52], these approaches still require substantial memory for processing high-resolution inputs, limiting scalability in batch size and data throughput. Reducing resolution mitigates memory demands but worsens spatial information loss, degrading segmentation accuracy. To resolve this trade-off, Kong et al. [21] introduced Scalable Training from Range-view (STR), which splits range images into sub-images from multiple perspectives to lower memory consumption. However, STR's use of partial scene views during training inherently limits segmentation accuracy. Furthermore, the method does not optimize for computational efficiency, leaving runtime improvements unaddressed.

Post-Processing Addressing the prevalent "many-to-one" problem in range-view representations often necessitates a post-processing step to upsample 2D predictions, a critical yet under-explored area in prior research. RangeViT [1] introduced a trainable 3D refiner using KPConv [36]. However, while it directly optimizes 3D semantics, the performance improvement is limited, and the approach adds significant computational overhead. Some methods rely on conventional unsupervised techniques to infer semantics for 3D points. For instance, Milioto et al. [28] proposed a KNN-based voting approach, and Zhao et al. [52] introduced Nearest Label Assignment (NLA), which assigns labels based on the closest labeled point in 3D space. Nevertheless, these unsupervised techniques often struggle with accurately predicting boundaries and distant points, with performance further declining as range-image resolution decreases. To overcome these limitations, we design a new post-processing method that adaptively weights neighboring contributions for interpolating predictions in 3D space.

3. Methodology

3.1. Proposed Framework

A LiDAR point cloud consists of points captured during a single revolution, denoted as $P = \{p_1, ..., p_n\}$, where each measurement represents a 4D point including the cartesian coordinates $p_i = \{x_i, y_i, z_i\}$ and intensity t_i. In semantic segmentation task, each point has an additional feature of class label, denoted as c_i, for training. Our splitting configuration operates as follows: given a predefined partition count N, we distribute points into sub-clouds by along the spherical coordinate of the scan pattern. Specifically, a sub-cloud is derived as $P_i = \{p_j | j \mod N = i - 1\}$ for $i \in \{1, 2, ..N\}$. This modulo-based assignment ensures equal partitioning while preserving spatial coherence within each sub-cloud. For the spherical projection, we specify the function f as following:

$$f(p) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{W}{2} - \frac{W}{2\pi} arctan(\frac{y}{x}) \\ \frac{H}{\Theta_{max} - \Theta_{min}} * (\Theta_{max} - arcsin(\frac{z}{d})) \end{bmatrix} \tag{1}$$

, where W and H as the image width and height, while v and u correspond to the elevation and azimuth angles of LiDAR points. Angular values Θ_{max} and Θ_{min} define the upper and lower bound of the LiDAR's vertical field of views and the depth value is calculated by $d = \sqrt{x^2 + y^2 + z^2}$. Note that H is typically determined by the number of LiDAR sensor beams, while W can be assigned with ran-

dom value based on the requirements. Similar to prior studies [6, 8, 52], we adopt a five-channel input representation (x, y, z, t, d).

Next, we project all sub-clouds into range images simultaneously $R_i = f(P_i)$. During *training*, since our splitting strategy leverages shared semantic contexts across sub-clouds, we randomly select *only one* for network optimization. During **inference**, all images R_i are **stacked as a batch** and processed by the network to extract the 2D predictions. Unlike STR [21], which wraps all images to the original resolution, we treat each sub-cloud as an independent complete scene and apply post-processing to low-resolution range images individually. Final 3D predictions are reconstructed by merging outputs from all sub-clouds.

An discussed in Sec. 1, *FLARES* offers three key advantages: 1) Enhanced restoration of original 3D information through multi-range projection, 2) Reduced memory consumption by decreasing the image resolution, enabling the scalable training on smaller GPUs, 3) Preservation of the full field of view, which maintains contextual integrity despite downsampling.

3.2. Data Augmentation

In particular, the point cloud splitting strategy of FLARES implicitly introduces two new challenges: exacerbated class imbalance and amplified projection noise. To address class imbalance, prior works [12, 16, 46] have primarily proposed augmenting long-tail classes by pasting them from other frames. While effective to some extent, we find that the augmentation strength is insufficient to counteract the heightened imbalance in our setting. To this end, we propose an enhanced data augmentation strategy specifically designed for class balancing. As for projection noise, recent approaches [47] attempt to mitigate it by interpolating pseudo pixels into unoccupied regions. However, this often introduces new artifacts in the range-view image. In contrast, tailored to the multi-range configuration of *FLARES*, we leverage the unused sub-clouds to increase pixel occupancy in the sampled range-view image, thereby reducing projection artifacts.

Weighted Paste-Drop+ (WPD+) To address the problem of class imbalance, a pervasive challenge in LiDAR semantic segmentation benchmarks [2, 13], Weighted Paste-Drop (WPD) [16] technique provides a simple yet effective solution by selectively pasting pixels from rare classes while dropping pixels from abundant ones. Building on this approach, we introduce WPD+, a fundamentally re-engineered augmentation strategy specifically designed for *FLARES*.

Unlike traditional methods that apply augmentation uniformly on range-view images [16, 21], WPD+ operates directly in 3D space, This design not only avoids repeated geometric transformations on sampled and current frames, but also harnesses multiple scenes to more effectively balance class distributions and mitigate the severe imbalance induced by point cloud splitting.

To further enhance occurrence of underrepresented classes, particularly those corresponding to small and dynamic objects in the scene, we incorporate a curated synthetic dataset from the Carla Simulator [11] (more details are provided in Appendix A.5). From our empirical results, despite inherent domain gaps, this approach has led to notable accuracy gains in long-tail classes.

Multi-Cloud Fusion (MCF) Due to the point cloud splitting in *FLARES*, the decrease in 2D occupancy leads to intensified projection artifacts in range images. To address the issue, we propose MCF, a strategy tailored to multi-range settings of *FLARES*. We formalize this process as follows. Let $R_i : \Omega \to \mathbb{R} \cup \{\varnothing\}$ denote the range image from the i-th sub-cloud over the pixel domain Ω. For each pixel $x \in \Omega$, We first define the set using other range images:

$$S(x) = \{\, R_j(x) \mid j \neq i \text{ and } R_j(x) \neq \varnothing \,\}.$$

Then, for a range image R_i over the pixel domain Ω, the occupancy-filled value $\tilde{R}_i(x)$ is given by:

$$\tilde{R}_i(x) = \begin{cases} R_i(x), & \text{if } R_i(x) \neq \varnothing, \\ f(S(x)), & \text{if } R_i(x) = \varnothing \text{ and } S(x) \neq \emptyset, \\ \varnothing, & \text{otherwise.} \end{cases}$$

Here, $f(\cdot)$ is an aggregation function that combines the occupied pixel values from the remaining $N-1$ range images. Consequently, this method maximizes the 2D occupancy in the range image of a sub-cloud while maintaining the structural consistency of the scene.

Despite their simplicity, both augmentation strategies are effective in enhancing segmentation performance. Moreover, when combined with the multi-range representations of *FLARES*, they significantly outperform prior methods (see Appendix B.3 for related experiments).

3.3. Post-Processing

After images are processed by the network, their 2D predictions must be reprojected into 3D space using a post-processing technique. Conventional methods [28, 52] typically rely on hard voting from nearest neighbors to infer predictions for points. However, they are designed for the single range image and requires an iterative processing of all images when working on *FLARES*, which results in increased computational complexity. Furthermore, these approaches are limited in their ability to appropriately weight the contributions of each neighbor in 3D coordinates, thus leading to sub-optimal performance. To leverage the advantages of multi-range settings and overcome these limitations, we propose a novel algorithm called *Nearest Neighbors Range Interpolation (NNRI)*. Its pseudo-code is shown

in Algo. 1.

Algorithm 1 Nearest Neighbors Range Interpolation

Define : N sub-clouds.
The annotation contains C classes

Input : Range images R_{ranges} with size $N \times H \times W$,
Softmax scores I_{scores} with size $N \times C \times H \times W$,
Arrays $R_{all}(p)$ with range values for all points,
Image coordinates (u_{all}, v_{all}) for all points,
Kernel size k,
Padding pad,
Cut-off factor α,
Mean of all range values r_{mean},
Standard Deviation of all range values r_{std}

Output: Array $Labels$ with predicted labels for all points.

1: **Unfold scores and ranges with $k \times k$ kernel:**
$S_s(n, h, w, k) \leftarrow \texttt{unfold}(I_{scores}, k, pad)$
$S_r(n, h, w, k) \leftarrow \texttt{unfold}(R_{ranges}, k, pad)$
2: **Extract nearest-neighbors for each point p:**
$\mathbf{N}_s(n, p, k) \leftarrow S_s(n, h, w, k)[..., u_{all}, v_{all}]$
$\mathbf{N}_r(n, p, k) \leftarrow S_r(n, h, w, k)[..., u_{all}, v_{all}]$
3: **Compute relative depths:**
$\mathbf{N}_{rel}(n, p, k) \leftarrow ||(\mathbf{N}_r(n, p, k) - R_{all}(p)||$
4: **Compute the cut-off value for each point p:**
$D(p) = \texttt{exp}(\frac{R(p) - r_{mean}}{r_{std}}) * \alpha$
5: **Filter the valid neighbors and compute weights:**
$\mathbf{N}_{valid}(n, p, k) \leftarrow \texttt{clamp}(\mathbf{N}_{rel}(n, p, k), \texttt{max} = D(p))$
$W(n, p, k) = 1 - \texttt{Normalize}(\mathbf{N}_{valid}(n, p, k))$
6: **Weighted Sum for 3D Projection:**
$Scores(p) = \sum_i^{k^2 \times n} W(n, p, k) * \mathbf{N}_s(n, p, k)$
$Labels = \texttt{argmax}_{c \in C}(Scores(p))$
7: **Return** $Labels$

After applying softmax to the network output, we begin by kernelizing 2D predictions and range images using a predefined kernel size (3×3 in our experiments). Next, we assign each point's nearest neighbors in 2D space with corresponding 2D coordinates and stack them along the sub-cloud dimension. The relative depth between each point and its neighbors is computed by taking the absolute difference in depth values. To extract valid data for interpolation, a threshold is needed to filter out distant neighbors. According to the prior knowledge [19, 23], using a constant threshold is sub-optimal due to differing point densities in LiDAR data: closer points are more likely to be affected by outliers due to high density, while farther points struggle to find valid neighbors due to sparsity. To fit this underlying geometry, the range value of each point is employed to determine its cut-off value. By normalizing the range using pre-computed mean and standard deviation, the cut-off value is derived from an exponential function, which approximates the relationship between point-sensor distance and density [24]. This approach simplifies computation by avoiding the costly nearest neighbor search in 3D space to calculate exact density values and adaptively assigns a threshold to each point. Once valid nearest neighbors are identified, they are normalized within the range of $[0, 1]$ to compute interpolation weights. Finally, softmax scores of all 3D points are interpolated by the weighted sum of their nearest neighbors.

Overall, NNRI is designed to effectively mitigate the "many-to-one" issue inherent in range-view methods by leveraging distance-wise local neighborhood information in both 2D and 3D. Moreover, the new approach eliminates the need for sub-cloud concatenation, as used in STR [21], and processes all range images in parallel, which significantly reduces the inference time.

4. Experimental Analysis

4.1. Settings

Datasets We conduct experiments on two public LiDAR semantic segmentation datasets. **SemanticKITTI** [2] dataset [2] consists of 22 sequences captured with a 64-beam LiDAR sensor, encompassing 19 semantic classes. The dataset is split as follows: sequences 00 to 10 (excluding 08) are used for training, sequence 08 is reserved for validation, and sequences 11 to 21 are designated for testing. **nuScenes** dataset [13] comprises 1,000 driving scenes recorded in Boston and Singapore using a 32-beam LiDAR sensor, leading to a relatively sparse point cloud. After merging similar and infrequent classes, the dataset includes 16 distinct semantic classes.

Networks We revisited prior works and selected three lightweight CNN-based networks: FIDNet [52], SalsaNext [8] and CENet [6]) for integration. To further test the effectiveness across heterogeneous architectures, we additionally deploy RangeViT [1], a network composed of a series of Vision Transformer blocks [10], in the experimental phase. Original RangeViT uses a trainable KPConv-based 3D projector to get the point-wise predictions. We replace it with our post-processing component to achieve the full integration of our framework and train the model from scratch.

Implementation Details Prior works experimented mostly with the resolution of 64×2048 for **SemanticKITTI** [6, 8, 28], and 32×960 [21] or 32×2048 [1] for **nuScenes**. In contrast, we reduce the azimuth resolution while increasing the projection rate in *FLARES* mode: resolutions of 64×512 for **SemanticKITTI** and 32×480 for **nuScenes** are fixed for the input and the full point cloud is split into up 3 and 2 sub-clouds during training and inference, respectively. *FLARES* uses the same loss configurations of prior works [1, 6, 8, 52], including a cross-entorpy loss (focal loss for RangeViT) and Lovász–Softmax loss [3]. For training the selected models (excluding RangeViT) on the

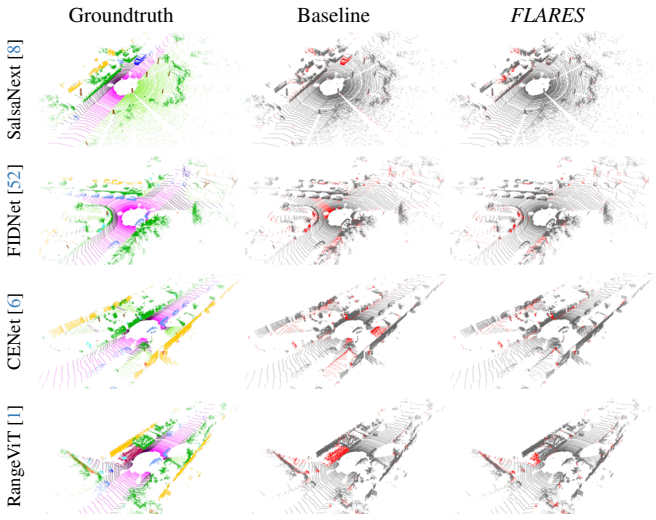


Figure 4. **Qualitative results on SemanticKITTI**[2] Points in red and gray represent incorrect and correct predictions, respectively. *More examples are provided in the supplementary material.

nuScenes dataset, we standardize the hyperparameter set since no default configurations are provided. Specifically, we use the AdamW optimizer [26] along with a OneCycle scheduler [33], setting the maximum learning rate to $1e^{-3}$ and training for 150 epochs. All models are trained on four NVIDIA GeForce GTX 1080Ti in distributed mode.

4.2. Comparative Study

We compare *FLARES* with baseline models across two datasets. As shown in Tab. 1, all four networks see significant improvements: SalsaNext has mIoU gains of 5.3% on SemanticKITTI and 2.3% on nuScenes, FIDNet improves by 7.9% and 3.9%, and CENet by 3.3% and 3.1%. RangeViT, as a ViT-based network, also exhibits huge enhancement in performance, confirming *FLARES*'s generalization across different architectures. This improvement is especially prominent for smaller, dynamic, and under-represented classes such as *truck*, *motorcycle*, *bicycle*, *pedestrian* and *bicyclist*. Notably, with the support of *FLARES*, the network demonstrates improved accuracy in segmenting foreground objects. For instance, as shown in Fig. 4, our method's predictions exhibit significantly higher correctness compared to the baseline.

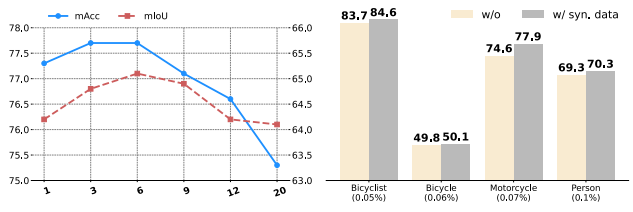
An exception arises with the *motorcyclist* class in SemanticKITTI, where IoU scores decrease compared to the baseline. Diving into the problem, this can be traced back to the extremely low occurrence of annotations for that class in the dataset. In standard training on low-resolution range images, this class already suffers from poor representation. In *FLARES* mode, the occurrence is further reduced by splitting of the point cloud. This accumulation of downsampling prevents the network from optimizing on that rare class effectively and lead to inferior performance. In contrast, the improvement on nuScenes is more consistent as class frequencies are better balanced. We regard this as a corner case when testing on an class-imbalanced dataset. As a future work to resolve the issue, we aim to explore 3D reconstruction techniques to generate real-world-like pseudo LiDAR point clouds for enhanced augmentation [4, 27].

In Tab. 2, we compare the performance of networks boosted by *FLARES* with other state-of-the-art approaches across various modalities. Despite using relatively fewer parameters, *FLARES*-enhanced models achieve segmentation accuracy comparable to other point- or voxel-based methods that deploy much larger and deeper networks. Moreover, our approach significantly outperforms these alternatives in terms of latency, achieving an excellent trade-off between accuracy and efficiency.

4.3. Ablation Study

To perform the ablation study, we test with CENet [6] on *val* set of SemanticKITTI [2] dataset.

Component Design In Tab. 3, we evaluate the contributions of each component in *FLARES*. Starting with the baseline results, we observe that applying STR [21] slightly reduces mIoU with the limited improvement in latency. Incorporating *FLARES* into the framework then yields a significant boost in performance compared to the baseline. However, since KNN post-processing must be iteratively applied to multiple range images, the inference speed becomes lower. Next, our two data augmentation methods effectively addressing the exacerbated class imbalance and amplified noise introduced by *FLARES*. Consequently, this leads to a further increase of 1.7% in mIoU. Finally, replacing the standard post-processing with NNRI not only improves the overall performance but also achieves a remarkable speed-up in inference due to its ability to run parallel computations across multiple range images. Essentially, compared to the baseline performance, we achieve 4.4% mIoU gain and around 45% speed-up through the new framework.



(a) (b)

Figure 5. a) Ablation on the number of sampled frames for WPD+. b) Comparative plot of IoU scores for top rare classes with and without the synthetic dataset, alongside the class frequency distribution in the validation set. For all inferences, KNN [28] is used as the post-processing approach.

WPD+ To address the challenge of class imbalance within *FLARES*, WPD+ is incorporated as a data augmentation

SemanticKITTI *test* set																				
Method	mIoU	car	bicy	moto	truc	o.veh	ped	b.list	m.list	road	park	walk	o.gro	build	fenc	veg	trun	terr	pole	sign
SalsaNext [8]	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	62.1
SalsaNext	63.3	94.7	52.9	55.7	57.3	50.2	65.5	70.9	13.0	92.6	**69.0**	77.7	20.5	90.4	65.8	80.8	65.0	63.4	55.4	62.4
◇**SalsaNext**	**64.8**	**95.1**	**55.5**	**56.5**	**60.1**	**53.7**	**69.6**	**74.1**	11.4	**93.0**	68.9	**78.9**	20.4	**91.1**	**67.6**	**82.0**	**66.7**	65.0	**58.1**	**64.1**
†FIDNet [52]	51.3	90.4	28.6	30.9	34.3	27.0	43.9	48.9	16.8	90.1	58.7	71.4	19.9	84.2	51.2	78.2	51.9	64.5	32.7	50.3
FIDNet	59.5	93.9	54.7	48.9	27.6	23.9	62.3	59.8	**23.7**	90.6	59.1	75.8	26.7	88.9	60.5	84.5	64.4	69.0	53.3	62.8
FIDNet	65.1	95.3	51.0	57.0	54.8	58.1	68.1	68.9	14.4	92.3	68.3	78.0	32.3	91.6	67.6	83.7	66.6	68.8	55.1	64.8
◇**FIDNet**	**67.4**	**95.8**	**56.7**	**60.7**	**58.1**	**60.3**	**72.5**	**72.9**	15.8	**93.2**	**69.2**	**79.9**	**34.2**	**91.9**	**69.0**	**84.6**	**68.7**	**70.3**	**59.9**	**66.9**
◇†CENet [6]	60.7	92.1	45.4	42.9	43.9	46.8	56.4	63.8	29.7	91.3	66.0	75.3	31.1	88.9	60.4	81.9	60.5	67.6	49.5	59.1
◇CENet	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
CENet	66.6	95.6	58.5	61.6	51.7	50.2	74.5	72.4	23.2	91.4	69.6	77.1	31.7	91.1	66.6	83.8	69.9	68.3	60.3	68.7
◇**CENet**	**68.0**	**95.9**	**61.1**	**62.1**	**57.2**	**59.0**	**77.2**	**74.2**	12.2	**92.2**	**69.9**	**78.7**	**32.9**	**91.8**	**68.8**	**84.7**	**71.3**	69.9	**62.9**	**70.3**
RangeViT [1]	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
RangeViT	**66.1**	**95.6**	**56.3**	**60.5**	**52.4**	**57.1**	**72.0**	**69.7**	16.0	91.6	**71.1**	77.3	**32.7**	91.4	67.4	83.1	68.0	68.1	58.0	**67.5**

nuScenes *val* set																	
Method (year)	mIoU	barrier	bicy	bus	car	const	moto	ped	traffic.c	trailer	truck	driv	o.flat	side	terrain	manm	veg
SalsaNext	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
SalsaNext	**74.5**	**75.0**	**34.6**	**90.4**	**90.0**	**43.8**	**79.4**	**72.9**	58.8	**65.8**	**79.9**	**96.5**	70.1	**74.0**	**73.9**	**87.6**	**85.6**
FIDNet	72.7	73.0	36.0	87.8	86.0	45.6	74.1	73.9	62.5	67.1	77.7	94.3	69.8	72.2	72.1	86.1	84.5
FIDNet	**76.6**	**77.6**	**43.5**	**92.9**	**88.1**	**56.5**	**79.5**	**77.7**	**65.3**	**67.0**	**83.1**	**96.6**	**72.8**	**75.0**	**74.5**	**88.5**	**86.8**
CENet	73.7	73.6	32.9	92.7	87.1	**53.5**	76.1	69.0	58.7	66.8	81.6	95.6	71.1	73.7	73.2	87.5	85.7
CENet	**76.8**	**76.7**	**45.2**	**93.5**	**90.3**	49.6	**83.1**	**78.1**	**66.4**	**69.0**	**82.5**	**96.6**	**73.9**	**75.1**	**74.6**	**88.3**	**86.3**
RangeViT	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**
RangeViT	**77.0**	**76.7**	39.2	**93.0**	**92.0**	**55.2**	**81.6**	**77.2**	64.9	**70.9**	**84.1**	**96.8**	**74.1**	**75.6**	**75.1**	88.6	86.7

Table 1. Comparisons of state-of-the-art LiDAR semantic segmentation methods on the *test* set of SemanticKITTI [2] and *val* set of nuScenes [13] in standard and *FLARES* mode. IoU scores are reported in percentages (%). For each method block, **bold** and underline indicate the **best** and second best result in the column. †Baseline results trained on low-resolution (64 × 512) range images. ◇Models inferred with test-time augmentation [21]. Note that we did not use model ensembling to further boost the model performance.

Method (year)	Size	Lat.	Modality	⊕	⊞	⊡
SalsaNext	6.7M	**29**	Range	**74.5**	**64.8**	64.8
PolarNet [50] ['20]	13.6M	71	Polar	71.0	54.9	57.5
SPVNAS [35] ['20]	12.5M	259	Voxel	-	64.7	**66.4**
RandLA-Net [45] ['20]	1.2M	55	Point	-	-	53.9
Tornado-Net [15] ['20]	-	-	Multiple	-	64.5	63.1
FIDNet	6.1M	**26**	Range	76.6	65.6	67.4
Cylinder3D [53] ['21]	56.3M	170	Voxel	76.1	67.8	65.9
RPVnet [46] ['21]	24.8M	168	Multiple	**77.6**	**68.2**	**70.3**
FPS-Net [43] ['21]	55.7M	48	Range	-	54.9	57.1
Lite-HDSeg [32] ['21]	-	50	Range	-	64.4	63.8
CENet	6.8M	**24**	Range	76.8	67.5	68.0
Meta-RSeg [37] ['22]	6.8M	46	Range	-	60.3	61.0
PVKD [18] ['22]	14.1M	76	Voxel	76.0	66.4	71.2
PTv2 [41] ['22]	12.8M	213	Point	**80.2**	**70.3**	**72.6**
2DPASS [48] ['22]	26.5M	119	Multiple	79.4	69.3	72.2
GFNet [31] ['22]	-	100	Multiple	76.8	63.2	65.4
WaffleIron [29] ['23]	6.8M	143	Point	79.1	68.0	70.8
SphereFormer [22] ['23]	32.3M	165	Multi	78.4	67.8	74.8
PTv3 [42] ['24]	46.2M	67	Point	80.4	70.8	74.2

Table 2. Comparisons of state-of-the-art LiDAR semantic segmentation methods in accuracy (mIoU [%]) and efficiency (Latency [ms]). All methods are categorized by year of publication. ⊕ represents *val* set of nuScenes [13], while ⊞ and ⊡ stand for *val* and *test* set of SemanticKITTI [2]. Latency with integration of *FLARES* is measured by processing all sub-clouds.

step. It includes two tunable parameters: the number of sampled frames from the original dataset and the inclusion of the synthetic dataset. To determine the optimal configuration, we conducted two ablation experiments. Fig. 5a shows that sampling 6 frames results in the optimal performance, while increasing the number of frames beyond this point leads to performance degradation. Fusing too many

STR [21]	*FLARES*	WPD+	MCF	NNRI	**mIoU**	**Lat.**
					63.1	44 ms
✓					62.6	41 ms
	✓				64.2	46 ms
	✓	✓			65.5	-
	✓	✓	✓		65.9	-
	✓	✓	✓	✓	67.5	24 ms

Table 3. Full ablation study on SemanticKITTI [2] dataset. The first row denotes the baseline results trained in standard mode with high-resolution range images (64 × 2048). For STR [21], we split the full range image into 4 sub-images, each has the resolution of 64 × 512. For all experiments without NNRI, we employ the conventional KNN [28] post-processing.

frames can introduce noise and redundant information that may overwhelm the model, for instance, excessive fusion might result in overlapping objects and distorted boundaries when too many new objects are pasted into the scene. Furthermore, Fig. 5b illustrates that the synthetic dataset plays a key-role in refining semantic prediction of top-rare classes. Notably, using the synthetic dataset is both efficient and practical, as it allows us to customize sensor configurations to align with the target dataset and define specific objects within the scene for downstream applications without incurring any labor cost.

Post-Processing We explore the performance of various post-processing techniques on both efficacy and efficiency in Fig. 6. Regarding conventional KNN [28] as the baseline, NLA [52] demonstrates similar performance in both accuracy and latency. In contrast, we deploy our approach

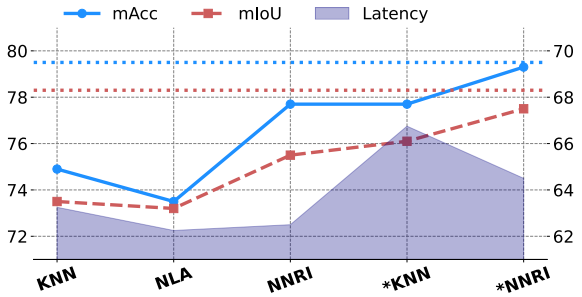
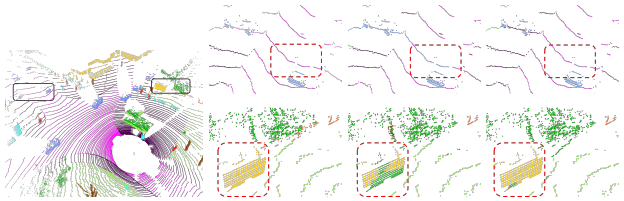


Figure 6. Ablation on various post-processing approaches: KNN [28], NLA [52], and NNRI are first compared in a single-range setting. *NNRI denotes the multi-range variant of our post-processing method. For fair evaluation, we also extend KNN to operate on multiple range images (*KNN). Dotted lines indicate the metrics for 2D predictions.

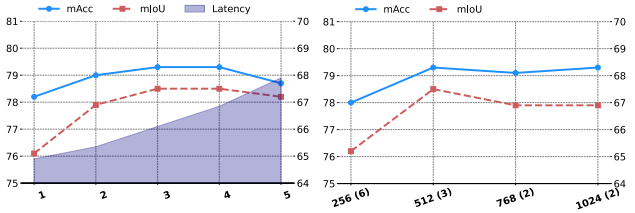


(a) Scene (b) GT (c) Constant (d) Adaptive

Figure 7. Segmentation results with different cut-off values in NNRI: in the case of constant value (set at 1), overlapping points of Road are partially misclassified as Car in the top image. Similarly, in the bottom image, half of the points that belong to Building are incorrectly predicted as Vegetation.

(NNRI) in the standard setting as well and observe a significant improvement: inference time is cut nearly 16% compared to KNN, while mAcc and mIoU increase by 2.8% and 2%, respectively. Unlike KNN, NNRI avoids the computational cost of Gaussian kernel calculations for distance weighting and directly performs nearest neighbor searches on the range image instead of in 3D space, which further reduces computational overhead. NNRI interpolates class-wise scores based on relative depths rather than directly voting on hard labels, relying more on weighted information from nearest neighbors, which is the major reason why it outperform other post-processing approaches.

Next, switching to *FLARES* mode, we first implement an extension of KNN, which iteratively gathers votes from all points in each sub-cloud and aggregates them for the final prediction. While this extension improves the accuracy, it comes at approximately doubled latency cost. Conversely, when NNRI is adapted to all sub-clouds, it consistently provides notable improvements in both efficacy and efficiency. As a reference, we included evaluation scores on 2D predictions, showing that *FLARES* with NNRI significantly narrows the accuracy gap between 2D and 3D predictions. This suggests that our approach effectively mitigates the "many-to-one" problem and offers substantial gains in segmentation performance. Furthermore, we optimize the implementation of NNRI by leveraging an adaptive cut-off value to filter valid nearest neighbors. This parameter is derived by approximating the internal LiDAR geometry and the distance-density relationship of 3D points. To illustrate its impact, qualitative results are provided in Fig. 7. As shown, the adaptive cut-off refines semantic predictions by better accommodating objects with varying density scales.



(a) (b)

Figure 8. (a) Ablation on number of sub-clouds (N) with fixed image resolution of 64×512. (b) Ablation on image width W (N is fine-tuned for each configuration, respectively).

Input Configuration In Fig. 8a, we investigate in the effect on the number of splitting. It shows that increasing the number of sub-clouds beyond 4 degrades performance due to insufficient occupancy and projection artifacts. Conversely, fewer sub-clouds speeds up the inference but compromises accuracy. Overall, our choice of 3 sub-clouds optimally balances efficiency and accuracy. In Fig 8b, we further test various azimuth resolution of input images. The experimental results indicate that width of 512 delivers the best mIoU scores. Increasing or decreasing the azimuth resolution beyond this point causes a slight performance drop.

5. Conclusion

In this work, we introduce *FLARES*, an optimized multi-range training paradigm designed for range-view LiDAR semantic segmentation. Our framework seamlessly integrates into any range-view-based network. We also develop two data augmentation techniques to address the challenges of exacerbated class imbalance and amplified projection noise within *FLARES*. Additionally, we propose a novel post-processing method tailored to multi-range settings to effectively tackle the "many-to-one" issue. Our approach yields significant improvements in both accuracy and efficiency over baselines across various network architectures on two widely used LiDAR benchmarks. The ***limitations*** of our current work are twofold: (1) although our data augmentation is powerful, it remains limited in corner cases of extremely low occurrence of certain classes, motivating further exploration of solutions to class imbalance; and (2) the fixed nature of the splitting step may result in the loss of critical information. In future work, we aim to address these limitations and develop a more robust framework that can be generalized to more challenging scenarios, such as adverse weather conditions or limited annotations.

References

[1] Angelika Ando, Spyros Gidaris, Andrei Bursuc, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5240–5250, 2023. 1, 3, 5, 6, 7

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019. 2, 4, 5, 6, 7

[3] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. 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*, pages 4413–4421, 2018. 5

[4] Mincheol Chang, Siyeong Lee, Jinkyu Kim, and Namil Kim. Just add $100 more: Augmenting nerf-based pseudo-lidar point cloud for resolving class-imbalance problem. *arXiv preprint arXiv:2403.11573*, 2024. 6

[5] Mincheol Chang, Siyeong Lee, Jinkyu Kim, and Namil Kim. Just add 100 more: Augmenting pseudo-lidar point cloud for resolving class-imbalance problem. In *Advances in Neural Information Processing Systems*, pages 66226–66259. Curran Associates, Inc., 2024. 2

[6] Hui-Xian Cheng, Xian-Feng Han, and Guo-Qiang Xiao. Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving. In *2022 IEEE international conference on multimedia and expo (ICME)*, pages 01–06. IEEE, 2022. 1, 2, 3, 4, 5, 6, 7

[7] 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*, pages 3075–3084, 2019. 3

[8] 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*, pages 207–222. Springer, 2020. 1, 2, 3, 4, 5, 6, 7

[9] Luca Deininger, Bernhard Stimpel, Anil Yuce, Samaneh Abbasi-Sureshjani, Simon Schönenberger, Paolo Ocampo, Konstanty Korski, and Fabien Gaire. A comparative study between vision transformers and cnns in digital pathology. *arXiv preprint arXiv:2206.00389*, 2022. 3

[10] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3, 5

[11] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR, 2017. 4

[12] A.Xiao et.al. Polarmix: A general data augmentation technique for lidar point clouds. 2022. 4

[13] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7(2):3795–3802, 2022. 1, 4, 5, 7

[14] 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*, pages 3354–3361. IEEE, 2012. 1

[15] 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)*, pages 9543–9549. IEEE, 2021. 7

[16] Yi Gu, Yuming Huang, Chengzhong Xu, and Hui Kong. Maskrange: A mask-classification model for range-view based lidar segmentation. *arXiv preprint arXiv:2206.12073*, 2022. 2, 4

[17] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Lidar-based 4d panoptic segmentation via dynamic shifting network. *arXiv preprint arXiv:2203.07186*, 2022. 1

[18] Yuenan Hou, Xinge Zhu, Yuexin Ma, Chen Change Loy, and Yikang Li. Point-to-voxel knowledge distillation for lidar semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8479–8488, 2022. 7

[19] Jordan SK Hu, Tianshu Kuai, and Steven L Waslander. Point density-aware voxels for lidar 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8469–8478, 2022. 5

[20] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11108–11117, 2020. 3

[21] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 228–240, 2023. 1, 2, 3, 4, 5, 6, 7

[22] 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*, pages 17545–17555, 2023. 3, 7

[23] Felix Järemo Lawin, Martin Danelljan, Fahad Shahbaz Khan, Per-Erik Forssén, and Michael Felsberg. Density adaptive point set registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3829–3837, 2018. 5

[24] Quan Liu, Hongzi Zhu, Zhenxi Wang, Yunsong Zhou, Shan Chang, and Minyi Guo. Extend your own correspondences: Unsupervised distant point cloud registration by progressive distance extension. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20816–20826, 2024. 5

[25] Romain Loiseau, Mathieu Aubry, and Loic Landrieu. Online segmentation of lidar sequences: Dataset and algorithm. *ECCV*, 2022. 1

[26] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[27] Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich, Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simulation by leveraging the real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11167–11176, 2020. 6

[28] 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)*, pages 4213–4220, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[29] Gilles Puy, Alexandre Boulch, and Renaud Marlet. Using a waffle iron for automotive point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3379–3389, 2023. 7

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1, 3

[31] Haibo Qiu, Baosheng Yu, and Dacheng Tao. GFNet: Geometric flow network for 3d point cloud semantic segmentation. *Transactions on Machine Learning Research*, 2022. 7

[32] Ryan Razani, Ran Cheng, Ehsan Taghavi, and Liu Bingbing. Lite-hdseg: Lidar semantic segmentation using lite harmonic dense convolutions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9550–9556. IEEE, 2021. 3, 7

[33] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017. 6

[34] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 1

[35] 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 *European conference on computer vision*, pages 685–702. Springer, 2020. 7

[36] 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*, pages 6411–6420, 2019. 3

[37] Song Wang, Jianke Zhu, and Ruixiang Zhang. Meta-rangeseg: Lidar sequence semantic segmentation using multiple feature aggregation. *IEEE Robotics and Automation Letters*, 7(4):9739–9746, 2022. 7

[38] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 3

[39] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1887–1893. IEEE, 2018. 3

[40] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *2019 international conference on robotics and automation (ICRA)*, pages 4376–4382. IEEE, 2019. 3

[41] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *NeurIPS*, 2022. 3, 7

[42] 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 *CVPR*, 2024. 1, 7

[43] Aoran Xiao, Xiaofei Yang, Shijian Lu, Dayan Guan, and Jiaxing Huang. Fps-net: A convolutional fusion network for large-scale lidar point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 176:237–249, 2021. 7

[44] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34: 12077–12090, 2021. 3

[45] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 1–19. Springer, 2020. 7

[46] 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*, pages 16024–16033, 2021. 4, 7

[47] et al. Xu X. Frnet: Frustum-range networks for scalable lidar segmentation. *arXiv preprint arXiv:2312.04484*, 2023. 4

[48] 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*, pages 677–695. Springer, 2022. 7

[49] Wen Yang, Zheng Gong, Baifu Huang, and Xiaoping Hong. Lidar with velocity: Correcting moving objects point cloud distortion from oscillating scanning lidars by fusion with camera. *IEEE Robotics and Automation Letters*, 7(3):8241–8248, 2022. 2

[50] 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*, pages 9601–9610, 2020. 1, 7

[51] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 1, 3

[52] Yiming Zhao, Lin Bai, and Xinming Huang. Fidnet: Lidar point cloud semantic segmentation with fully interpolation decoding. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4453–4458. IEEE, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[53] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Wei Li, Yuexin Ma, Hongsheng Li, Ruigang Yang, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar-based perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6807–6822, 2021. 1, 3, 7