

# ⚡FLARES⚡: Fast and Accurate LiDAR Multi-Range Semantic Segmentation

## Supplementary Material

### A. More Implementation details

In this section, we provide additional details regarding to the overall workflow, proposed components and configurations during training and inference.

**Evaluation Metrics** Following prior works, we assess the performance using Intersection-over-Union (IoU) and Accuracy (Acc) for each class, and calculate the mean Intersection-over-Union (mIoU) and mean Accuracy (mAcc) across all classes.

#### A.1. Detailed Overview

Each step in the training and testing process is detailed in Fig. 1. To maximize the efficiency, the sequence of pre-processing and data augmentation steps is fixed. WPD+ is applied first, before Geometric Data Augmentation (GDA), to avoid performing the same geometric transformation on all sampled frames. Once these two point-level steps are completed, the augmented point cloud is downsampled and divided into multiple sub-clouds, on which projection is performed. Multi-Cloud Fusion (MCF) is applied last, conducting pixel-wise fusion on the range images from each sub-cloud. The same steps are repeated for ground-truth labels to obtain their 2D representation. The final input and ground-truth label are then used to train the 2D semantic segmentation network. During testing, only the pre-processing steps are included, and all range images are fed into the network. Ultimately, we apply NNRI to upsample 2D outputs to gather 3D predictions in an unsupervised manner.

#### A.2. Geometric Data Augmentation

The standard augmentation is utilized commonly in the previous works [2, 4, 15]. Hence, we follow the default configurations and include the following geometric transformations in our pipeline:

- **Random Flipping** The point cloud is randomly flipped along the  $X$ -axis.
- **Random Translation** The point cloud is randomly jittered in translation. The jittering values range from -5m to 5m, -3m to 3m, and -1m to 0m for  $X$ -axis,  $Y$ -axis, and  $Z$ -axis, respectively.
- **Random Rotation** The point cloud is randomly and globally rotated along yaw, pitch and roll direction. The ranges are set at  $[-5, 5]$  in degrees for all axes.
- **Probability** the value of randomness is set at 0.5 for all augmentation components.

#### A.3. Runtime Measurement

In Table 2, we report the inference time of various methods. All measurements are conducted using a single NVIDIA GTX 1080Ti GPU with a fixed batch size of 1. To eliminate the impact of initialization overhead, we discard the first 100 iterations and compute the average runtime over the subsequent 100 iterations. For methods that could not be measured directly due to GPU limitations, we report the inference times provided in their original papers, most of which were obtained using more powerful GPUs. Notably, all our reported times include data loading, pre-processing, and post-processing steps to ensure a fair comparison.

#### A.4. Weighted Paste Drop+

We directly follow the configuration of Weighted Paste Drop (WPD) introduced by Gu et al. [11]: we first compute the normalized weights from the class-wise frequencies and set the threshold of classifying long-tail and non-long-tail classes at 0.1. More specifically, we paste the points of classes with weights greater than 0.1 and drop them vice versa. The normalized weight of each class is directly utilized as the probability of pasting and dropping. For reference, we provide the relevant statistics on SemanticKITTI<sup>1</sup> [3] and nuScenes<sup>2</sup> [10] in Fig. 3.

When applying WPD+ to the point clouds in the SemanticKITTI dataset, we observe that the resulting range images sometimes contain artifacts outside of the contextual distributions. As shown in Fig. 4, points at farther distances are left unlabeled in the dataset, as they are less relevant for most perceptual tasks. However, these points can introduce additional noise into the projection when non-long-tail foreground points are excluded. Such outliers can adversely impact training stability, so we further clean the projected range images by filtering out unlabeled representations when training the model with SemanticKITTI dataset.

#### A.5. Synthetic Dataset

In addition to copy-pasting points of long-tail classes from the same dataset, we use Carla Simulator [9] to further enhance class balance. This subsection details the process of data collection.

Since WPD+ pastes points without spatial transformation and does not account for sensor difference, it is essential that the simulator precisely matches the sensor configuration of the target dataset to prevent introducing out-of-

<sup>1</sup><https://github.com/PRBonn/semantic-kitti-api>

<sup>2</sup><https://www.nuscenes.org/nuscenes>

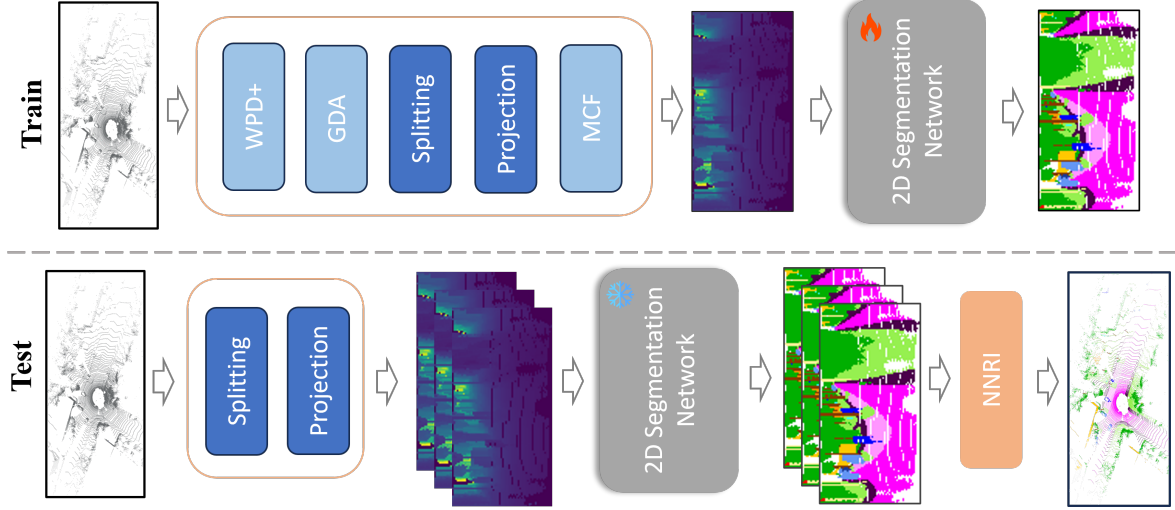


Figure 1. Detailed workflow overview: During training, the raw point cloud is pre-processed and augmented to generate the range image. A 2D segmentation network is then trained to predict 2D semantic labels. In testing, all augmentation steps are removed, and range images are stacked as a batch. The 2D predictions are then gathered and processed by NNRI, our proposed post-processing component, to map them into 3D space.

Dataset	V.FoV [deg]	H.FoV [deg]	Range [m]
Sem.KITTI	-25 ~ 3	-180 ~ 180	2 ~ 50
nuScenes	-30 ~ 10	-180 ~ 180	2 ~ 80

Table 1. LiDAR specifications of two datasets.

context points during augmentation. We therefore configure the simulator separately for each dataset, based on their respective sensor specifications, as detailed in Tab. 1. Next, we set up the scene-related contexts. Based on statistics shown in Fig. 3, we initialize the scene with 20 dynamic actors randomly selected from vehicle categories *busses*, *motorcycles* and *bicycles* in the Carla Simulator, along with 30 *pedestrians* with random movement. Town10 serves as the static background map for the scene, and we record 2000 frames each for SemanticKITTI and nuScenes.

After collecting the dataset, we manually define a class mapping scheme between the real and synthetic datasets. The complete mapping function is detailed in Tab. 2. Generally, positive correspondences can be identified; however, an exception exists where the *rider* class in the synthetic dataset represents both *motorcyclist* and *bicyclist*. To make more effective use of the synthetic dataset, we split and re-assign the labels for this class. Since *rider* always appears alongside *motorcycle* and *bicycle* in the scene, we assign new labels by finding the nearest neighbor in 3D space for each *rider* point and categorizing it in a binary manner. We provide an example of the pasting process with the synthetic data in Fig. 2 for the comprehensive understanding.

Sem.KITTI	Carla	nuScenes	Carla
bicycle	bicycle	barrier	-
motorcycle	motorcycle	bicycle	bicycle
truck	truck	bus	bus
other vehicle	bus, train	const. vehicle	-
person	pedestrian	motorcycle	motorcycle
bicyclist	rider*	pedestrian	pedestrian
motorcyclist	rider*	traffic cone	-
other ground	ground	trailer	-
trunk	-	other flat	ground
pole	pole		
traffic sign	traffic sign		

Table 2. Class mapping dictionaries: we map only the long-tail classes and omit mappings where there is ambiguity in finding corresponding class names.

## A.6. Multi-Cloud Fusion

To better illustrate the impact of Multi-Cloud Fusion (MCF) during data augmentation, we provide an example in Fig. 5. Notably, range images with high azimuth resolution tend to have a greater number of empty pixels, which are distributed randomly and can distort object geometry within the scene. Reducing the azimuth resolution produces a more complete range image; however, occupancy declines again if only the sub-cloud is available for projection. MCF addresses this issue by filling unoccupied pixels, revisiting data from other sub-clouds at the same 2D positions. In this way, MCF effectively mitigates the problem while preserving the underlying geometry of the scene.

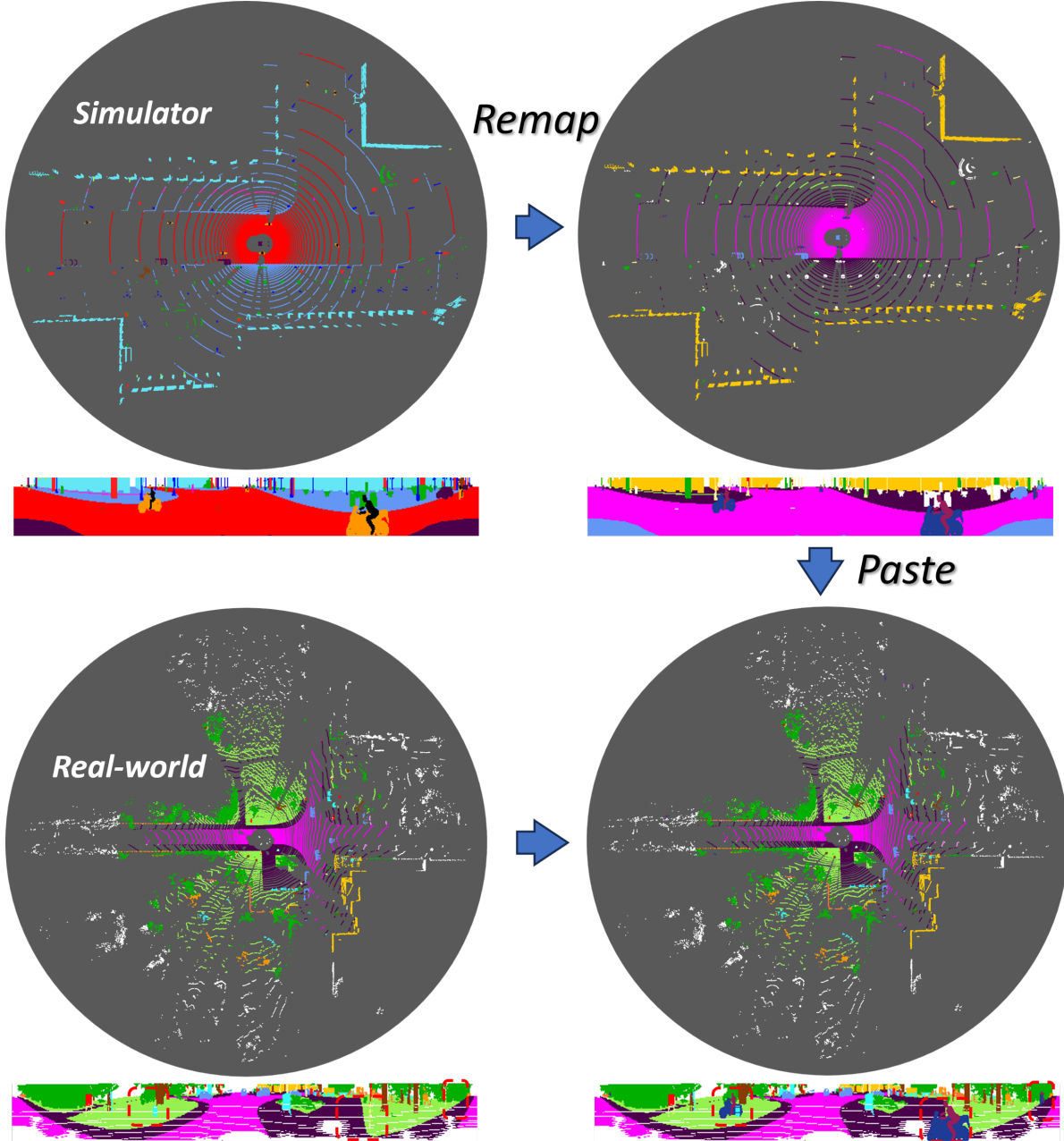


Figure 2. An example of pasting points of long-tail classes from synthetic data onto real-world data is shown, with the fused point cloud and range image visualized in the bottom right. Since WPD+ does not account for spatial differences in scene-related contexts, some unrealistic scenarios may arise, such as a motorcyclist riding on a non-drivable surface. Nevertheless, these augmentation steps enhance the semantic richness of the scene. Given that long-tail classes typically correspond to small and dynamic objects, the negative impact of domain differences remains minimal in this case.

### A.7. Multi-range KNN

The KNN-based post-processing approach [19] has been widely used in prior works. To apply this method with the *FLARES* setup, we extend it to handle multiple range images, allowing us to infer 3D semantic labels for a single full point cloud. The pseudo code is provided in Algo. 1.

Briefly, we gather KNN votes for all points from each image iteratively, accumulating them across images. Each point is then assigned the class with the highest vote count.

### A.8. Pre-training

In our main experiments, we test the effectiveness of *FLARES* using four different networks: SalsaNext [6], FID-

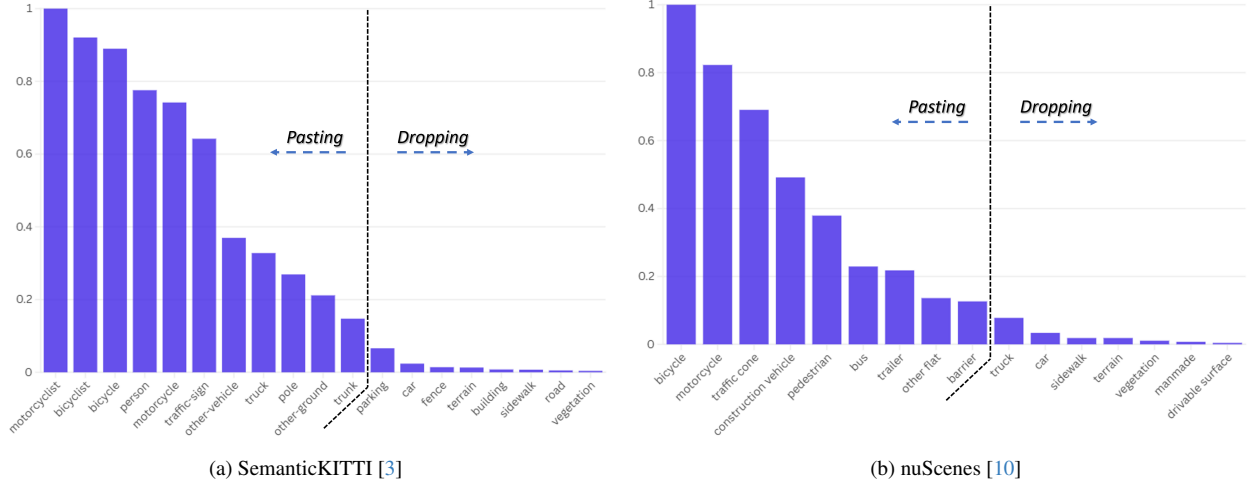


Figure 3. Statistical results of class-wise normalized weights on two datasets.

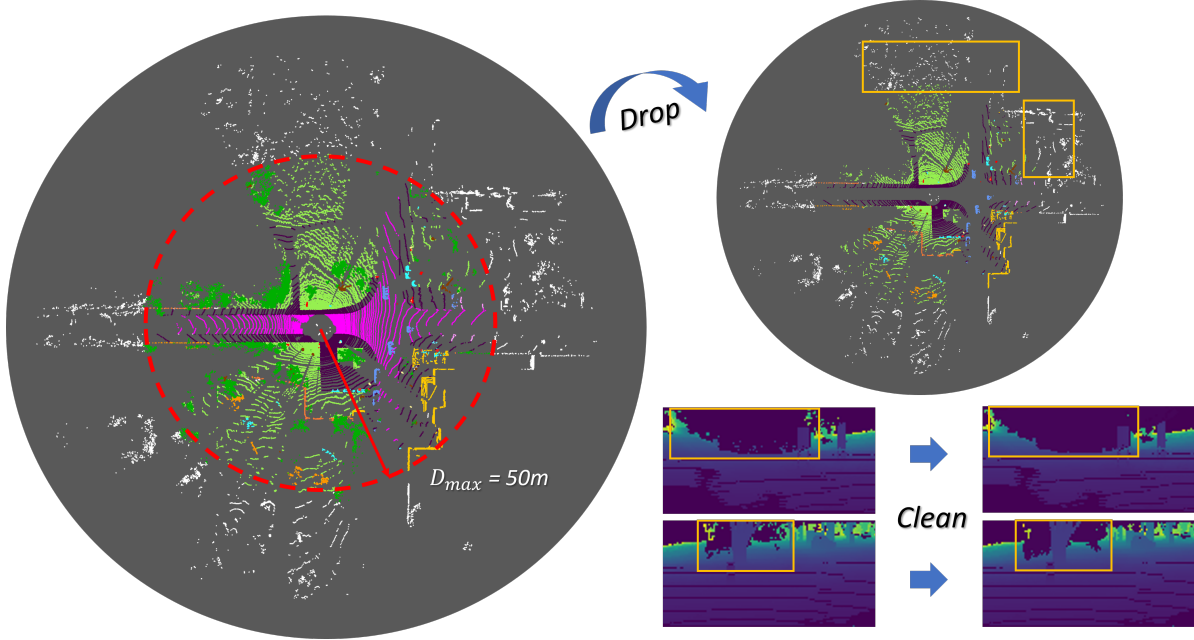


Figure 4. An example of dropping points of non-long-tail classes in a point cloud from SemanticKITTI [3] dataset: since the dataset is only annotated within the range of 50 meters, points that are out of this range can result in some artifacts in the projection after points of foreground objects are dropped in the scene. Hence, we insert an additional cleaning step after projection for the dataset.

Net [35], CENet [4], and RangeViT [2]. For specific network designs, we refer readers to the respective original papers. Notably, RangeViT [2] demonstrated the advantages of using pretrained models on image datasets [25]. Following this approach, we experiment with initializing network weights pre-trained on Cityscapes [5]. While we apply this pre-training strategy to the other three convolution-based networks, it yields only minor improvements compared to random initialization. We hypothesize that Convolutional Neural Networks (CNNs) are less effective than Vision Transformers (ViTs) in terms of transferability and

scalability, as ViTs are better equipped to learn long-range dependencies within the dataset and significantly less biased towards local textures than CNNs [8, 27]. Consequently, we do not conduct further experiments with other pre-trained weights on the three CNNs.

## B. More quantitative results

### B.1. Detailed Results

In Tab. 7, we present the class-wise evaluation results of FLARES-boosted approaches compared to various LiDAR



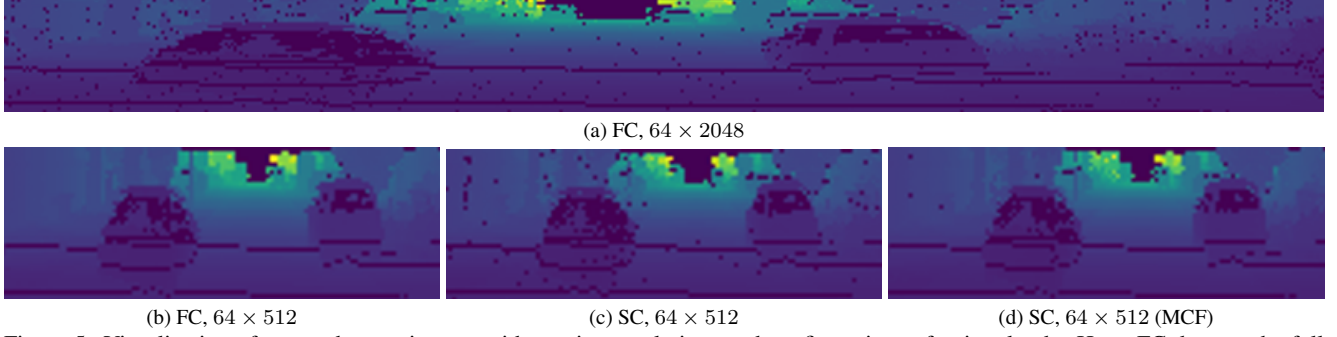


Figure 5. Visualization of cropped range images with varying resolutions and configurations of point clouds. Here, FC denotes the full cloud, and SC indicates the sub-cloud. The final image is augmented with Multi-Cloud Fusion (MCF).

---

**Algorithm 1** Multi-range KNN

---

**Define :**  $N = N_{max}$  sub-clouds.

The annotation contains  $C$  classes

**Input :** Range images  $R_{ranges}$  with size  $N \times H \times W$ ,  
Label images  $L_{labels}$  with size  $N \times H \times W$ ,  
Arrays  $R_{all}(p)$  with range values for all points,  
Image coordinates  $(u_{all}, v_{all})$  for all points,

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

$Vote(p) = \text{zeros}(p)$

**foreach**  $i$  in  $1 : N$  **do**

$R_{range} = R_{ranges}(i)$

$L_{label} = L_{labels}(i)$

$Vote(p) += \text{KNN}(R_{range}, L_{label}, R_{all}, (u_{all}, v_{all}))$

**end**

$Labels = \text{argmax}_{c \in C}(Votes(p))$

---

semantic segmentation methods on SemanticKITTI [3], including point-wise, voxel-wise, hybrid, and range-wise approaches. In Tab. 8, we additionally demonstrate the results from nuScenes [10] leaderboard. Due to a limited number of entries, we do not categorize methods by publication year and evaluate only a single model using *FLARES*. Clearly, *FLARES* enables the model to achieve superior accuracy compared to most methods and delivers performance comparable to the best-performing approach, all while maintaining significantly higher computational efficiency.

Since range-view methods generally offer greater efficiency, it is valuable to compare performance without the use of test-time augmentation. We summarize the results in Tab. 9. Rather than comparing with a broad set of range-view approaches introduced over the past decade, we focus on a selection of the most recent and representative methods, as they demonstrate the best performance among them. Although *FLARES*-boosted networks may not exhibit significantly superior segmentation accuracy compared to the latest approaches, they excel in computational efficiency and usage flexibility. For instance, RangeFormer [15] requires high computational resources due to its multiple Vi-

sion Transformer blocks [8], which contain a substantial number of model parameters. Similarly, TFNet [17] processes a sequence of LiDAR data rather than a single frame to incorporate temporal features, and its performance is highly dependent on the sequence length.

## B.2. RangeFormer

To further assess the potential of our method in approaching the state-of-the-art performance, we reproduced the code for RangeFormer [15], the current best-performing range-view method. Results are shown in Tab. 3. Due to the absence of official code, minor discrepancies exist between our reproduced results and the original paper. Nonetheless, integrating our method significantly boosts inference speed while achieving comparable accuracy to SOTA methods of other representations.

Model	Params.	S.KITTI <i>val</i>	nuScenes <i>val</i>	Lat.
*RangeFormer	24.3M	68.1	77.1	-
RangeFormer	-	67.0	76.5	87 ms
+ <i>FLARES</i>	24.1M	68.9	78.2	55 ms
Cylinder3D [36]	-	65.9	76.1	-
SphereFormer [16]	-	67.8	78.4	-
PVKD [12]	-	66.4	76.0	-

Table 3. \*: Results reported in the original paper [20].

## B.3. Comparison of Data Augmentation

We compare our two data augmentation methods with existing approaches that share the same objective. The results are presented in Tables 4 and 5, evaluated on the *val* set of the SemanticKITTI dataset [3].

Aug.	mIoU
RangePaste [15]	65.4
Mix3D [20]	64.6
Ins. CutMix [30]	66.2
WPD+	67.5

Table 4. Augmentation methods for class imbalance.

Aug.	mIoU
RangeUnion [15]	66.1
RangeIP [31]	66.9
MCF	67.5

Table 5. Augmentation methods for projection noise.

## B.4. Do other standard hyper-parameters matter?

For consistency, we unify the hyperparameter settings across all experiments. To ensure that improvements in seg-

mentation accuracy are not simply due to changes in hyper-parameters, we additionally retrain each network using the default settings specified in their respective original papers. The corresponding results are reported in Table 6.

Model	Baseline		<i>FLARES</i>	
	mIoU	Lat.	mIoU	Lat.
CENet [5]	62.6	44 ms	65.7	24 ms
FIDNet [49]	58.9	46 ms	63.4	26 ms
SalsaNext [7]	59.0	51 ms	64.1	29 ms

Table 6. We retrain the networks with default parameter settings of the original baselines on SemanticKITTI *val* set. These include batch size, optimizer, learning rate, weight decay, training epochs and loss weights.

### C. More qualitative results

We provide additional qualitative results of various networks, with and without the integration of *FLARES*, in Fig. 6~9. Compared to the baseline, *FLARES* enhances predictions in scenes with diverse geometries. Notably, segmentation accuracy is significantly improved for foreground and close-to-sensor objects, which are more critical for perception systems than distant ones.

### D. Video Demo

In addition to the figures, we have included two video demos per network tested (eight videos in total) in the supplementary materials to showcase a more comprehensive evaluation of our approach. Each video contains hundreds of frames, with each frame presenting three visualizations: groundtruth, baseline prediction, and *FLARES* prediction. Each visualization includes both top-down-view and range-view images of the LiDAR point cloud. For videos labeled with the suffix *errormap*, predictions are displayed in a binary format, with incorrect predictions shown in **red** and correct ones in **gray**.

SemanticKITTI <i>test</i> set																				
Method (year)	mIoU	car	bicy	moto	truc	o.veh	ped	b.list	m.list	road	park	walk	o.gro	build	fenc	veg	trun	terr	pole	sign
RandLA-Net [13] [20]	50.3	94.0	19.8	21.4	42.7	38.7	47.5	48.8	4.6	90.4	56.9	67.9	15.5	81.1	49.7	78.3	60.3	59.0	44.2	38.1
PolarNet [34] [20]	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
SqSegV3 [29] [20]	55.9	92.5	38.7	36.5	29.6	33.0	45.6	46.2	<b>20.1</b>	91.7	63.4	74.8	26.4	89.0	59.4	82.0	58.7	65.4	49.6	58.9
KPConv [26] [20]	58.8	<u>96.0</u>	32.0	42.5	33.4	44.3	61.5	61.6	<u>11.8</u>	88.8	61.3	72.7	<u>31.6</u>	<b>95.0</b>	64.2	84.8	69.2	<u>69.1</u>	56.4	47.4
FusionNet [23] [20]	61.3	95.3	47.5	37.7	41.8	34.5	59.5	56.8	11.9	<u>91.8</u>	68.8	<u>77.1</u>	30.8	<u>92.5</u>	<b>69.4</b>	<b>85.6</b>	<u>69.8</u>	68.5	<u>60.4</u>	<u>66.5</u>
AMVNet [18] [20]	<b>65.3</b>	<b>96.2</b>	<b>59.9</b>	<u>54.2</u>	<u>48.8</u>	<u>45.7</u>	<b>71.0</b>	<u>65.7</u>	11.0	90.1	<b>71.0</b>	75.8	<b>32.4</b>	92.4	<u>69.1</u>	<b>85.6</b>	<b>71.7</b>	<b>69.6</b>	<b>62.7</b>	<b>67.2</b>
⚡SalsaNext [6] [20]	<u>64.8</u>	95.1	<u>55.5</u>	<b>56.5</b>	<b>60.1</b>	<b>53.7</b>	<u>69.6</u>	<b>74.1</b>	11.4	<b>93.0</b>	<u>68.9</u>	<b>78.9</b>	20.4	91.1	67.6	82.0	66.7	65.0	58.1	64.1
MPF [1] [21]	55.5	93.4	30.2	38.3	26.1	28.5	48.1	46.1	18.1	90.6	62.3	74.5	30.6	88.5	59.7	83.5	59.7	69.2	49.7	58.1
KPRNet [14] [21]	63.1	95.5	54.1	47.9	23.6	42.6	65.9	65.0	16.5	<b>93.2</b>	<b>73.9</b>	<b>80.6</b>	30.2	91.7	68.4	<b>85.7</b>	<u>69.8</u>	71.2	58.7	64.1
LiteHDSeg [24] [21]	63.8	92.3	40.0	55.4	37.7	39.6	59.2	<u>71.6</u>	<b>54.3</b>	93.0	68.2	78.3	29.3	91.5	65.0	78.2	65.8	65.1	59.5	<u>67.7</u>
JS3C-Net [32] [21]	66.0	<u>95.8</u>	<u>59.3</u>	52.9	<u>54.3</u>	46.0	69.5	65.4	39.9	88.9	61.9	72.1	31.9	<b>92.5</b>	<b>70.8</b>	84.5	<u>69.8</u>	67.9	<u>60.7</u>	<b>68.7</b>
Cylinder3D [36] [21]	<b>68.9</b>	<b>97.1</b>	<b>67.6</b>	<b>63.8</b>	50.8	<u>58.5</u>	<b>73.7</b>	69.2	<u>48.0</u>	92.2	65.0	77.0	<u>32.3</u>	90.7	66.5	<u>85.6</u>	<b>72.5</b>	<u>69.8</u>	<b>62.4</b>	66.2
⚡FIDNet [35] [21]	<u>67.4</u>	<u>95.8</u>	56.7	<u>60.7</u>	<b>58.1</b>	<b>60.3</b>	<u>72.5</u>	<b>72.9</b>	15.8	<b>93.2</b>	<u>69.2</u>	<u>79.9</u>	<b>34.2</b>	<u>91.9</u>	<u>69.0</u>	84.6	68.7	<b>70.3</b>	59.9	66.9
Meta-RSeg [28] [22]	61.0	93.9	50.1	43.8	43.9	43.2	63.7	53.1	18.7	90.6	64.3	74.6	29.2	91.1	64.7	82.6	65.5	65.5	56.3	64.2
PCSCNet [21] [22]	62.7	95.7	48.8	46.2	36.4	40.6	55.5	68.4	<b>55.9</b>	89.1	60.2	72.4	23.7	89.3	64.3	84.2	68.2	68.1	60.5	63.9
GFNet [22] [22]	65.4	<u>96.0</u>	53.2	48.3	31.7	47.3	62.8	57.3	44.7	<b>93.6</b>	<b>72.5</b>	<b>80.8</b>	<u>31.2</u>	<b>94.0</b>	<b>73.9</b>	<u>85.2</u>	71.1	69.3	61.8	68.0
MaskRange [11] [22]	66.1	94.2	56.0	55.7	<u>59.2</u>	52.4	67.6	64.8	31.8	91.7	<u>70.7</u>	77.1	29.5	90.6	65.2	84.6	68.5	69.2	60.2	66.6
GASNet [33] [22]	<b>70.7</b>	<b>96.9</b>	<b>65.8</b>	<u>58.0</u>	<b>59.3</b>	<b>61.0</b>	<b>80.4</b>	<b>82.7</b>	<u>46.3</u>	89.8	66.2	74.6	30.1	<u>92.3</u>	<u>69.6</u>	<b>87.3</b>	<b>73.0</b>	<b>72.5</b>	<b>66.1</b>	<b>71.6</b>
⚡CENet [4] [22]	<u>68.0</u>	95.9	<u>61.1</u>	<b>62.1</b>	57.2	<u>59.0</u>	<u>77.2</u>	<u>74.2</u>	12.2	<u>92.2</u>	69.9	<u>78.7</u>	<b>32.9</b>	91.8	68.8	84.7	<u>71.3</u>	<u>69.9</u>	<u>62.9</u>	<u>70.3</u>

Table 7. The class-wise IoU scores of *different* LiDAR semantic segmentation approaches on the *test* set of SemanticKITTI [3]. All IoU score are given in percentage (%). All approaches are categorized by the year of publication. In each block, **bold** and underline indicate the **best** and second best result in the column.

nuScenes <i>test</i> set																	
Method (year)	mIoU	barrier	bicy	bus	car	const	moto	ped	traffic.c	trailer	truck	driv	o.flat	side	terrain	manm	veg
PolarNet [34] [20]	69.4	72.2	16.8	77.0	86.5	51.1	69.7	64.8	54.1	69.7	63.5	96.6	67.1	77.7	72.1	87.1	84.5
JS3C-Net [32] [21]	73.6	80.1	26.2	<u>87.8</u>	84.5	55.2	72.6	71.3	66.3	76.8	<u>71.2</u>	96.8	64.5	76.9	74.1	87.5	86.1
AMVNet [18] [20]	77.3	80.6	<u>32.0</u>	81.7	88.9	<u>67.1</u>	<u>84.3</u>	76.1	<u>73.5</u>	<b>84.9</b>	67.3	<u>97.5</u>	67.4	79.4	<u>75.5</u>	<b>91.5</b>	88.7
Cylinder3D [36] [21]	77.2	<u>82.8</u>	29.8	84.3	89.4	63.0	79.3	<b>77.2</b>	73.4	84.6	69.1	<b>97.7</b>	<u>70.2</u>	<b>80.3</b>	<u>75.5</u>	90.4	87.6
RangeFormer [15] [23]	<b>80.1</b>	<b>85.6</b>	<b>47.4</b>	<b>91.2</b>	<b>90.9</b>	<b>70.7</b>	<b>84.7</b>	<u>77.1</u>	<b>74.1</b>	83.2	<b>72.6</b>	<u>97.5</u>	<b>70.7</b>	79.2	75.4	<u>91.3</u>	<u>88.9</u>
⚡CENet [4] [22]	<u>77.5</u>	82.2	31.1	85.2	<u>90.7</u>	65.1	82.5	74.9	72.3	<u>84.7</u>	70.5	97.4	69.5	<u>79.6</u>	<b>76.1</b>	90.5	<b>90.1</b>

Table 8. The class-wise IoU scores compared with *different* LiDAR semantic segmentation approaches on the *test* set of nuScenes [10]. All IoU score are given in percentage (%). Due to limited number of submission times, we only test *FLARES* with CENet [4]. **Bold** and underline indicate the **best** and second best result in the column.

SemanticKITTI <i>test</i> set																					
Method (year)	#params	mIoU	car	bicy	motor	truck	ovh	ped	b.list	m.list	road	park	walk	ogro	build	fenc	veg	trun	terr	pole	sign
RangeFormer [15] [23]	24.3M	<b>69.5</b>	94.7	<u>60.0</u>	<b>69.7</b>	<b>57.9</b>	<b>64.1</b>	72.3	<b>72.5</b>	<b>54.9</b>	90.3	<u>69.9</u>	74.9	<b>38.9</b>	90.2	66.1	<b>84.1</b>	68.1	<b>70.0</b>	58.9	63.1
LENet [7] [23]	4.7M	64.5	93.9	57.0	51.3	44.3	44.4	66.6	64.9	<u>36.0</u>	91.8	68.3	76.9	30.5	91.2	66.0	83.7	68.3	67.8	58.6	63.2
TFNet [17] [24]	-	66.1	94.3	<b>60.7</b>	58.5	38.4	48.4	<u>74.3</u>	72.2	35.5	90.6	68.5	75.3	29.0	<b>91.6</b>	67.3	<u>83.8</u>	<b>71.1</b>	67.0	<b>60.8</b>	<b>68.7</b>
⚡SalsaNext [20]	6.7M	63.3	94.7	52.9	55.7	57.3	50.2	65.5	70.9	13.0	<b>92.6</b>	69.0	<u>77.7</u>	20.5	90.4	65.8	80.8	65.0	63.4	55.4	62.4
⚡FIDNet [21]	6.1M	65.1	95.3	51.0	57.0	54.8	<u>58.1</u>	68.1	68.9	14.4	<u>92.3</u>	68.3	<b>78.0</b>	32.3	<b>91.6</b>	<b>67.6</b>	83.7	66.6	<u>68.8</u>	55.1	64.8
⚡CENet [22]	6.8M	<u>66.6</u>	<b>95.6</b>	58.5	<u>61.6</u>	51.7	50.2	<b>74.5</b>	<u>72.4</u>	23.2	91.4	69.6	77.1	31.7	91.1	66.6	<u>83.8</u>	<u>69.9</u>	68.3	<u>60.3</u>	<b>68.7</b>
⚡RangeViT [23]	23.6M	66.1	<b>95.6</b>	56.3	60.5	52.4	<b>57.1</b>	72.0	69.7	16.0	91.6	<b>71.1</b>	77.3	<u>32.7</u>	91.4	<u>67.4</u>	83.1	68.0	68.1	58.0	67.5

Table 9. The class-wise IoU scores compared with *latest range-view* LiDAR semantic segmentation approaches on the *test* set of SemanticKITTI [3]. All IoU score are given in percentage (%). All approaches are evaluated *without test-time augmentation*. **Bold** and underline indicate the **best** and second best result in the column.

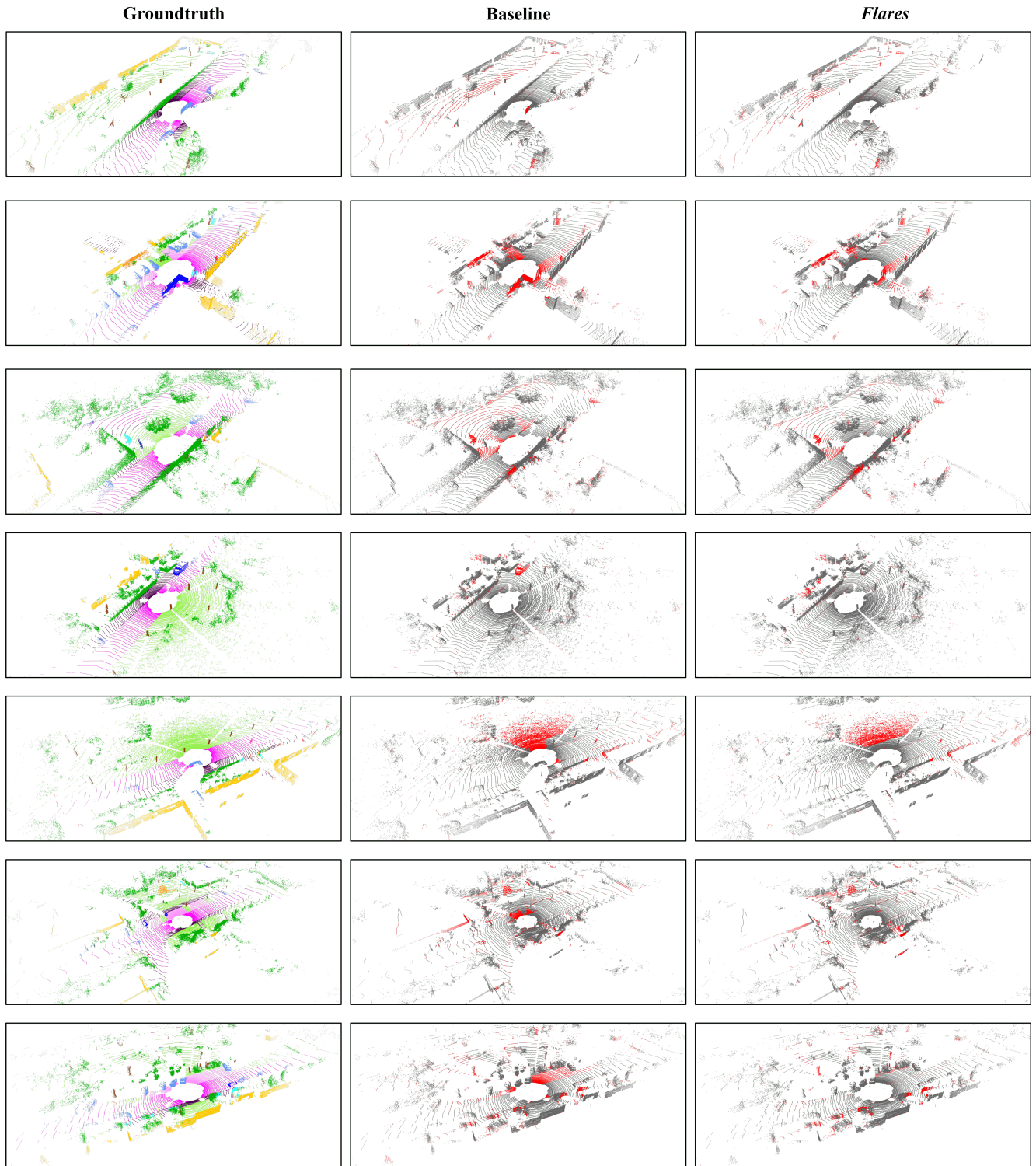


Figure 6. Additional qualitative results of **SalsaNext** [6]: points in **red** and **gray** represent incorrect and correct predictions, respectively. The groundtruth image is color-coded to differentiate between classes.



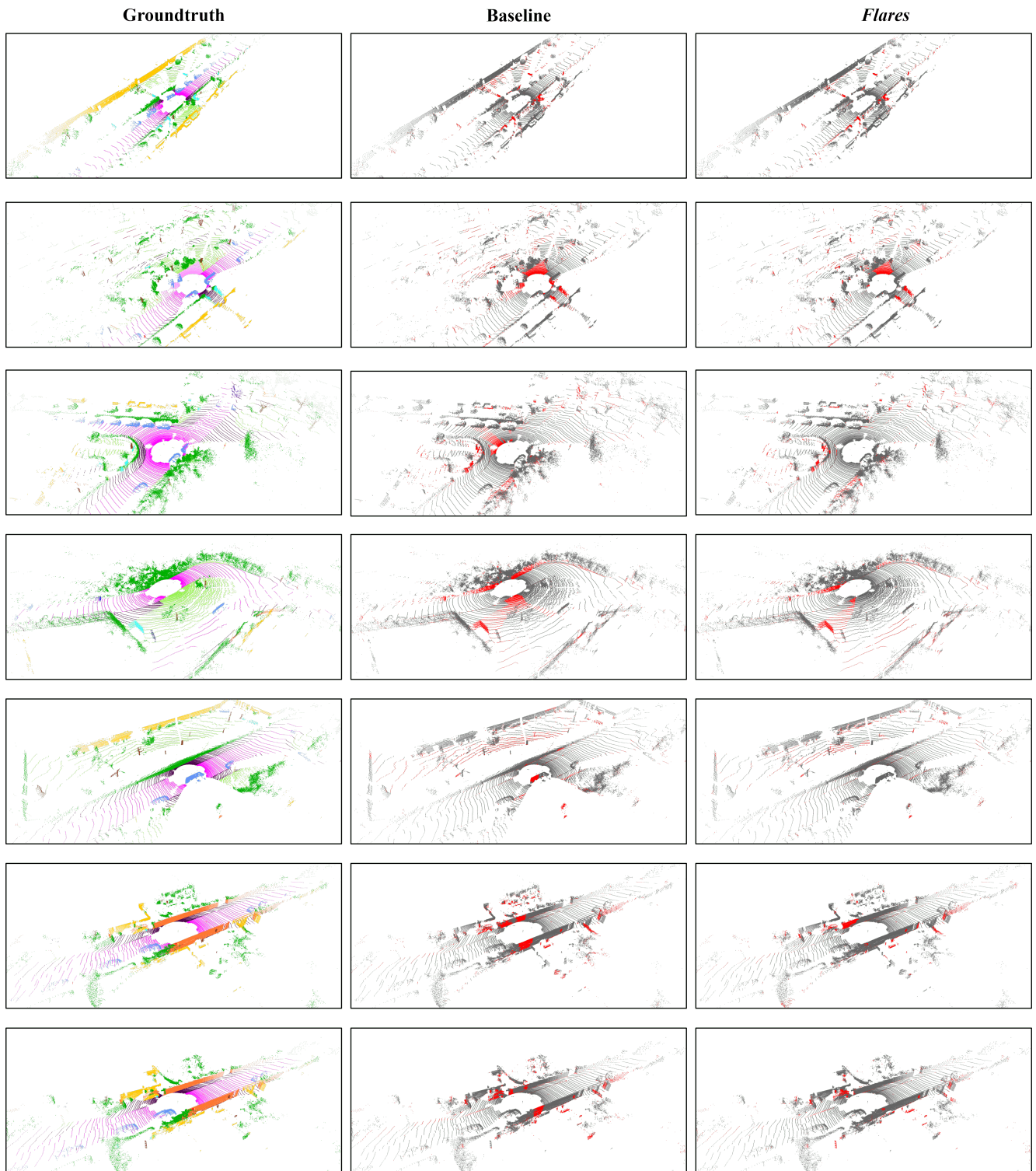


Figure 7. Additional qualitative results of **FIDNet** [35]: points in **red** and **gray** represent incorrect and correct predictions, respectively. The groundtruth image is color-coded to differentiate between classes.



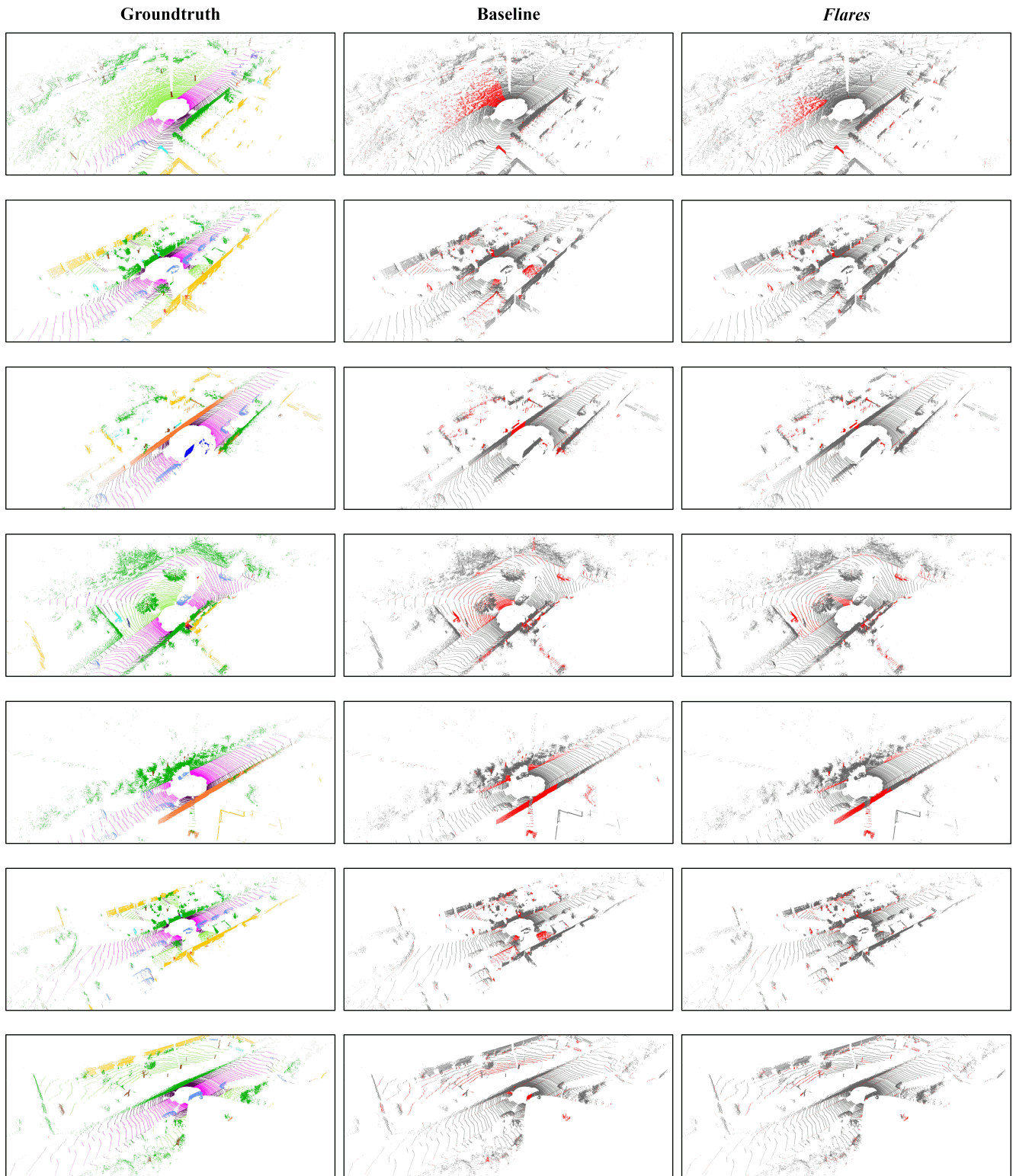


Figure 8. Additional qualitative results of **CENet** [4]: points in **red** and **gray** represent incorrect and correct predictions, respectively. The groundtruth image is color-coded to differentiate between classes.

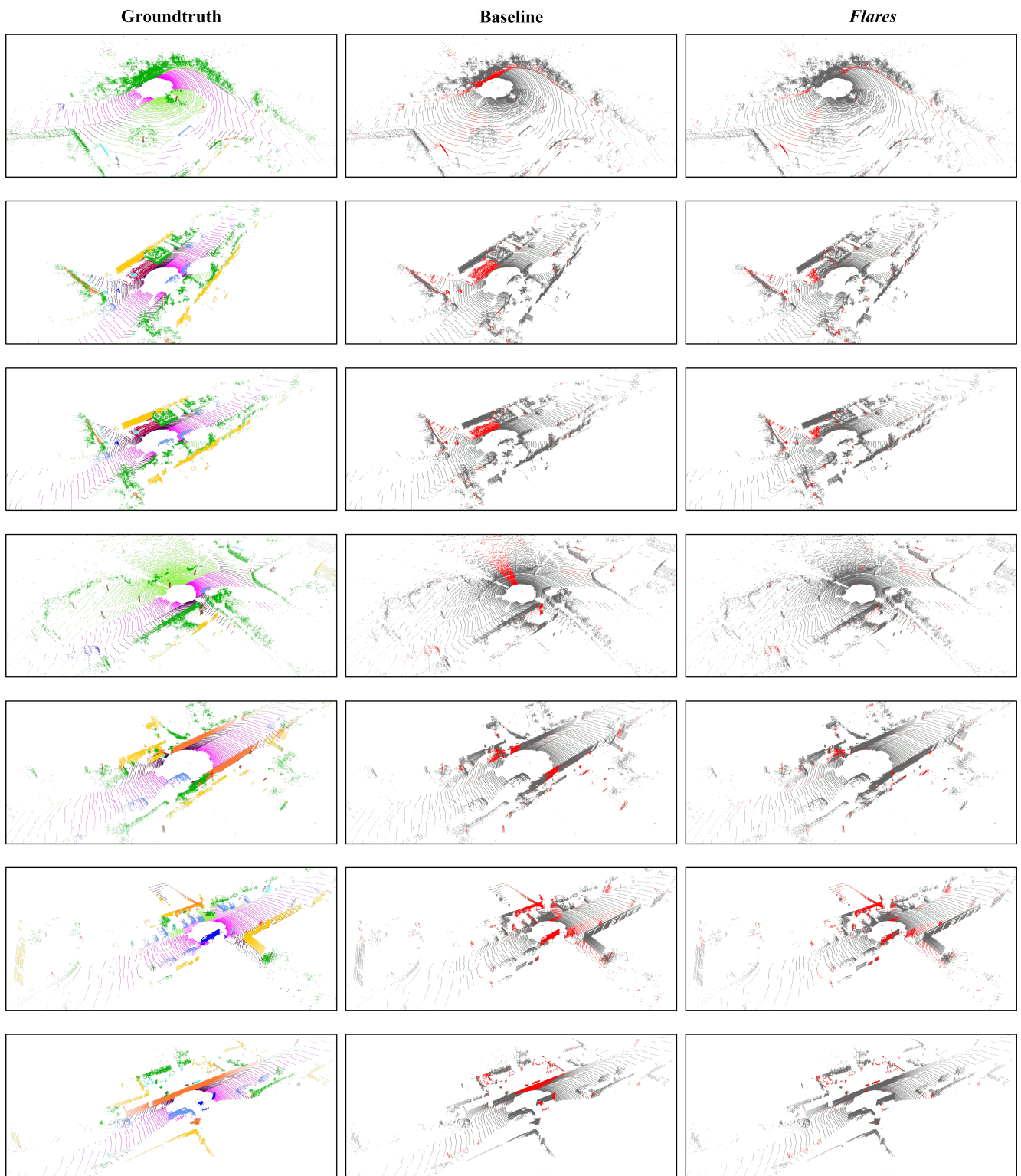


Figure 9. Additional qualitative results of **RangeViT** [2]: points in **red** and **gray** represent incorrect and correct predictions, respectively. The groundtruth image is color-coded to differentiate between classes.

## References

- [1] Yara Ali Alnaggar, Mohamed Afifi, Karim Amer, and Mohamed ElHelw. Multi projection fusion for real-time semantic segmentation of 3d lidar point clouds. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1800–1809, 2021. 7
- [2] 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, 4, 11
- [3] 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. 1, 4, 5, 7
- [4] 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, 4, 7, 10
- [5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 4
- [6] 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. 3, 7, 8
- [7] Ben Ding. Lenet: Lightweight and efficient lidar semantic segmentation using multi-scale convolution attention. *arXiv preprint arXiv:2301.04275*, 2023. 7
- [8] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 4, 5
- [9] 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. 1
- [10] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nusenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7(2):3795–3802, 2022. 1, 4, 5, 7
- [11] 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. 1, 7
- [12] 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. 5
- [13] 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. 7
- [14] Deyvid Kochanov, Fatemeh Karimi Nejadasl, and Olaf Booij. Kprnet: Improving projection-based lidar semantic segmentation. *arXiv preprint arXiv:2007.12668*, 2020. 7
- [15] 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, 5, 7
- [16] 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. 5
- [17] Rong Li, Shijie Li, Xieyuanli Chen, Teli Ma, Juergen Gall, and Junwei Liang. Tfnet: Exploiting temporal cues for fast and accurate lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4547–4556, 2024. 5, 7
- [18] Venice Erin Liong, Thi Ngoc Tho Nguyen, Sergi Widjaja, Dhananjai Sharma, and Zhuang Jie Chong. Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. *arXiv preprint arXiv:2012.04934*, 2020. 7
- [19] 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. 3
- [20] Alexey Nekrasov, Jonas Schult, Or Litany, Bastian Leibe, and Francis Engelmann. Mix3d: Out-of-context data augmentation for 3d scenes. In *2021 international conference on 3d vision (3dv)*, pages 116–125. IEEE, 2021. 5
- [21] Jaehyun Park, Chansoo Kim, Soyeong Kim, and Kichun Jo. Pescnet: Fast 3d semantic segmentation of lidar point cloud for autonomous car using point convolution and sparse convolution network. *Expert Systems with Applications*, 212: 118815, 2023. 7
- [22] Haibo Qiu, Baosheng Yu, and Dacheng Tao. GFNet: Geometric flow network for 3d point cloud semantic segmentation. *Transactions on Machine Learning Research*, 2022. 7
- [23] Tran Minh Quan, David Grant Colburn Hildebrand, and Won-Ki Jeong. Fusionnet: A deep fully residual convolutional neural network for image segmentation in connectomics. *Frontiers in Computer Science*, 3:613981, 2021. 7
- [24] 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. 7



- [25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. [4](#)
- [26] 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. [7](#)
- [27] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. [4](#)
- [28] 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](#)
- [29] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeeze-segv3: 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](#)
- [30] 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. [5](#)
- [31] et al. Xu X. Frnet: Frustum-range networks for scalable lidar segmentation. *arXiv preprint arXiv:2312.04484*, 2023. [5](#)
- [32] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3101–3109, 2021. [7](#)
- [33] Maosheng Ye, Rui Wan, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Efficient point cloud segmentation with geometry-aware sparse networks. In *Computer Vision – ECCV 2022*, pages 196–212, Cham, 2022. Springer Nature Switzerland. [7](#)
- [34] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zelong 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. [7](#)
- [35] 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. [4](#), [7](#), [9](#)
- [36] 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. [5](#), [7](#)