— Supplementary Material —
Calib3D: Calibrating Model Preferences for Reliable 3D Scene Understanding

Lingdong Kong[1,2,*], Xiang Xu[3,*], Jun Cen[4], Wenwei Zhang[1], Liang Pan[1], Kai Chen[1], Ziwei Liu[5,✉]

[1]Shanghai AI Laboratory [2]National University of Singapore [3]Nanjing University of Aeronautics and Astronautics
[4]The Hong Kong University of Science and Technology [5]S-Lab, Nanyang Technological University

Table of Contents

A. Calib3D Benchmark

In this section, we elaborate on additional details about the proposed Calib3D benchmark, including basic configurations regarding the datasets (Sec. A.1), models (Sec. A.2), evaluation protocols (Sec. A.3), and license (Sec. A.4).

A.1. 3D Datasets

The Calib3D benchmark encompasses a total of **10** popular datasets in the area of 3D scene understanding, with a diverse spectrum of dataset configurations regarding data collections, label mappings, and annotation protocols. Tab. A provides an overview of the datasets used in our benchmark. The key features of each dataset are summarized as follows.

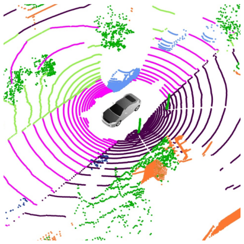
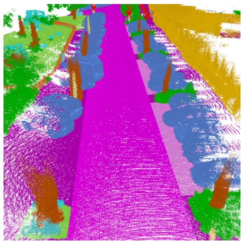
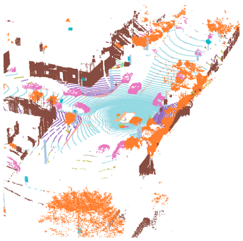
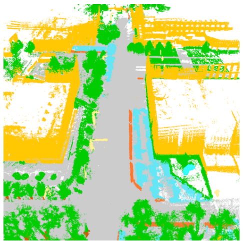
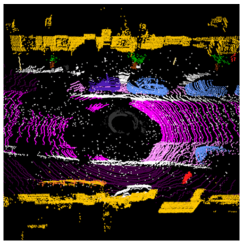
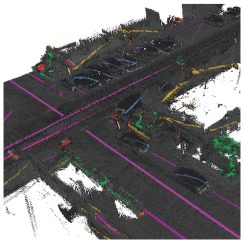
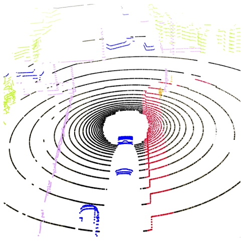
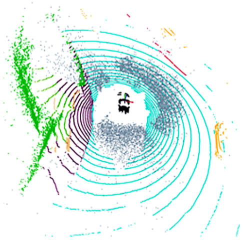
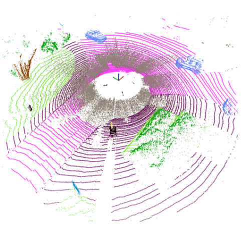
- **nuScenes** [13] is one of the most popular driving datasets in autonomous vehicle research, featuring multimodal data from Boston and Singapore. It contains 1000 scenes with 1.1 billion annotated LiDAR points acquired by a Velodyne HDL32E LiDAR sensor. In this work, we use the *lidarseg* subset of the Panoptic-nuScenes dataset, which provides point-wise class and instance labels across 16 merged semantic categories. For more information: https://www.nuscenes.org/nuscenes.

- **SemanticKITTI** [3] offers 22 densely labeled LiDAR sequences of urban street scenes, making it one of the most prevailing benchmarks for LiDAR-based semantic scene understanding. The point clouds are acquired by a Velodyne HDL-64E LiDAR sensor and are annotated with a total of 19 semantic categories. For more information: http://semantic-kitti.org.

- **Waymo Open Dataset (WOD)** [36] is a large-scale dataset for autonomous driving. The 3D semantic segmentation subset of WOD comprises 1150 scenes, which are further split into 798 training, 202 validation, and 150 testing scenes, corresponding to 23691 training scans, 5976 validation scans, and 2982 testing scans, respectively. The LiDAR scans are annotated across 22 semantic categories. For more information: https://waymo.com/open.

- **SemanticPOSS** [31] is constructed with a special focus on dynamic scenes. It includes 2988 scans from

Table A. Summary of 3D datasets encompassed in the **Calib3D** benchmark. A total of **ten** 3D datasets have been used in our benchmark, including ^{1}*nuScenes* [13], ^{2}*SemanticKITTI* [3], ^{3}*Waymo Open* [36], ^{4}*SemanticPOSS* [31], ^{5}*SemanticSTF* [43], ^{6}*ScribbleKITTI* [39], ^{7}*Synth4D* [35], ^{8}*S3DIS* [2], and ^{9}*nuScenes-C* and ^{10}*SemanticKITTI-C* from the Robo3D benchmark [18]. Each dataset sheds light on a specific data acquisition and annotation protocol, such as different LiDAR sensor setups, adverse weather conditions, weak annotations, synthetic data, indoor scenes, and out-of-domain corruptions. The images shown here are adopted from the original dataset papers.

nuScenes	SemanticKITTI	Waymo Open	SemanticPOSS	SemanticSTF
				
ScribbleKITTI	Synth4D	S3DIS	nuScenes-C	SemanticKITTI-C
				

a 40-beam Hesai Pandora LiDAR sensor, offering insights into scene dynamics at Peking University's campus. For more information: https://www.poss.pku.edu.cn/semanticposs.

- **SemanticSTF** [43] is built on the STF dataset [4]. It features 2076 scans under various weather conditions in the real world, serving as a testbed for assessing model robustness. The point clouds are acquired by a Velodyne HDL64 S3D LiDAR sensor under snowy, foggy, and rainy scenarios. For more information: https://github.com/xiaoaoran/SemanticSTF.

- **ScribbleKITTI** [39] is an extension of the SemanticKITTI [3] dataset. It introduces weakly-supervised annotations through line scribbles, offering a cost-effective labeling approach for 19130 LiDAR scans, which are under the same data splits and semantic annotations of citebehley2019semanticKITTI. For more information: https://github.com/ouenal/scribblekitti.

- **Synth4D** [35] was collected utilizing CARLA simulations [12]. The Synth4D-nuScenes subset contains about 20000 labeled point clouds for testing model performance in virtual urban and rural scenes, where the label mappings are aligned with that of the nuScenes [13] dataset. For more information: https://github.com/saltoricristiano/gipso-sfouda.

- **S3DIS** [2] is a comprehensive collection of point clouds for indoor spaces. It encompasses detailed scans from six large-scale indoor areas that include over 215 million points and covers more than 6,000 square meters. Each point in the dataset is annotated with one of several semantic labels corresponding to different object categories like walls, floors, chairs, tables, *etc*. For more information: http://buildingparser.stanford.edu/dataset.html.

- **nuScenes-C** [18] is part of the 3D robustness benchmarks in Robo3D [18] and is built based on the nuScenes [13] dataset. It focuses on the 3D model's out-of-distribution robustness against eight types of common corruptions, offering a platform for testing under diverse adverse conditions. For more information: https://github.com/ldkong1205/Robo3D.

- **SemanticKITTI-C** [18] shares the same common corruption types with nuScenes-C and is built based on the SemanticKITTI [3] dataset. For more information: https://github.com/ldkong1205/Robo3D.

A.2. 3D Models

The Calib3D benchmark encompasses a total of **28** state-of-the-art models in the area of 3D scene understanding, with a diverse spectrum of LiDAR representations, network architectures, and pre-/post-processing. Tab. B provides a summary of the models used, including their LiDAR modalities and key features.

A.3. Benchmark Protocols

In this work, to ensure fairness in comparisons, we adopt the following protocols in model evaluations:

- All 3D scene understanding models are trained on the official *training* set of each 3D dataset, and evaluated on data from the official *validation* set. There is no overlap between training and evaluation data.
- To reflect the original behavior of each 3D scene understanding model, we directly use public checkpoints whenever applicable, or re-train the model using its default configuration. The acknowledgments of public checkpoints and implementations are included in Sec. F.
- We notice that some models (and their public checkpoints) are enhanced using extra "tricks" on the validation/testing sets, such as test time augmentation, model ensembling, *etc*. To ensure fairness, we re-train such models to reflect their "clean" performance.

A.4. License

The Calib3D benchmark is released under the *CC BY-NC-SA 4.0* license[1]. For licenses regarding the codebase used in the Calib3D benchmark, kindly refer to Appendix F.1. For licenses regarding the 3D datasets used in the Calib3D benchmark, kindly refer to Appendix F.2. For licenses regarding the model implementations used in the Calib3D benchmark, kindly refer to Appendix F.3.

B. Additional Implementation Detail

In this section, we provide additional implementation details to help reproduce the key results shown in this work.

B.1. 3D Model Training

Our Calib3D benchmark is constructed based on the popular MMDetection3D [9] and OpenPCSeg [27] codebase, as well as several standalone implementations that have not been integrated into MMDetection3D [9] and/or OpenPCSeg [27]. Most 3D models adopt a unified training configuration, including the number of training epochs, optimizer, and learning rate scheduler. We apply common 3D data augmentations in Cartesian space, including random flipping, rotation, scaling, and jittering. The 3D models are trained using eight GPUs with a batch size of 2. The number of epochs are set as 80 for *nuScenes* and 50 for *SemanticKITTI*, *Waymo Open*, *SemanticPOSS*, *SemanticSTF*, *ScribbleKITTI*, and *Synth4D*. For *S3DIS*, we follow the default setups as MMDetection3D [9]. For additional details, please refer to the corresponding codebase.

B.2. 3D Model Evaluation

We evaluate the 3D models by following the conventional evaluation setups. As mentioned in Sec. A.3, we do not use any extra "tricks" on the validation/testing sets, such as test time augmentation, model ensembling, *etc*.

B.3. PyTorch-Style ECE Calculation

To facilitate reproduction, we provide a PyTorch-style code snippet for calculating the expected calibration error (ECE) on point clouds in Listing 1.

```python
import torch
import torch.nn.functional as F

def calculate_ece(logits, labels, ignore_index, n_bins=10):
    valid_index = labels != ignore_index
    logits, labels = logits[valid_index], labels[valid_index]

    bin_bound = torch.linspace(0, 1, n_bins + 1)
    lowers, uppers = bin_bound[:-1], bin_bound[1:]

    softmaxes = F.softmax(logits, dim=1)
    confs, preds = torch.max(softmaxes, 1)
    accs = preds.eq(labels)

    ece = torch.zeros(1)
    for l, u in zip(lowers, uppers):
        in_bin = confs.gt(l.item()) * confs.le(u.item())
        prop_in_bin = in_bin.float().mean()
        if prop_in_bin.item() > 0:
            acc_in_bin = accs[in_bin].float().mean()
            avg_conf_in_bin = confs[in_bin].mean()
            ece += torch.abs(avg_conf_in_bin - acc_in_bin) * prop_in_bin

    return ece.item()
```

Listing 1. PyTorch-style code snippet for calculating ECE scores on point clouds.

B.4. PyTorch-Style Implementation of DeptS

To facilitate reproduction, we provide a PyTorch-style code snippet of the proposed depth-aware scaling (DeptS) method in Listing 2.

```python
import numpy as np
import torch
import torch.nn as nn

class Depth_Aware_Scaling(nn.Module):

    def __init__(self, threshold):
        super(Depth_Aware_Scaling, self).__init__()
        self.T1 = nn.Parameter(torch.ones(1))
        self.T2 = nn.Parameter(torch.ones(1) * 0.9)
        self.k  = nn.Parameter(torch.ones(1) * 0.1)
```

[1] https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode.en.

Table B. Summary of 3D models encompassed in the **Calib3D** benchmark. We categorize models into five distinct groups, based on their LiDAR representations, *i.e.*, [1]*range view*, [2]*bird's eye view*, [3]*sparse voxel*, [4]*multi-view fusion*, and [5]*raw point*. Each model sheds light on a specific network structure and model configuration.

Model	Modality	Key Feature	Ref
RangeNet^{++}	● Range View	The first range view LiDAR segmentation model with a FCN structure	[30]
SalsaNext	● Range View	Uncertainty-aware range view segmentation with dilation modules	[11]
FIDNet	● Range View	Fully interpolation encoding for better range view post processing	[51]
CENet	● Range View	Concise and efficient range view learning with unified model structure	[7]
RangeViT	● Range View	Replace ResNet backbone with ViT for enhancing range view learning	[1]
RangeFormer	● Range View	Combine RangeAug, RangePost, and RangeFormer for better results	[17]
FRNet	● Range View	Frustum-range fusion & interpolation for scalable LiDAR segmentation	[47]
PolarNet	● Bird's Eye View	Point cloud embedding using polar coordinates for real-time processing	[50]
MinkUNet$_{18}$	● Sparse Voxel	Highly efficient sparse convolution operators with cubic voxel grids	[8]
MinkUNet$_{34}$	● Sparse Voxel	Enhanced MinkUNet structure with a larger segmentation backbone	[8]
Cylinder3D	● Sparse Voxel	Cylindrical voxel representation for balanced LiDAR points encoding	[52]
SpUNet$_{18}$	● Sparse Voxel	MinkUNet structure with SpConv operators for efficient 3D learning	[10]
SpUNet$_{34}$	● Sparse Voxel	Enhanced SpUNet structure with a larger segmentation backbone	[10]
RPVNet	● Multi-View Fusion	Multi-view fusion of range, point, and voxel for modality interactions	[45]
2DPASS	● Multi-View Fusion	Distillation from images to enhance point cloud feature learning	[48]
SPVCNN$_{18}$	● Multi-View Fusion	Efficient sparse point-voxel convolutions & a lightweight architecture	[37]
SPVCNN$_{34}$	● Multi-View Fusion	Enhanced SPVCNN structure with a larger segmentation backbone	[37]
CPGNet	● Multi-View Fusion	Cascade point-grid fusion & transformation consistency regularization	[25]
GFNet	● Multi-View Fusion	Complementary geometric flow fusion of range and bird's eye views	[34]
UniSeg	● Multi-View Fusion	Unified multi-view representation learning and cross-view distillation	[28]
KPConv	● Raw Point Input	Deformable convolutions for adaptive kernel-based geometry learning	[38]
PIDS$_{1.25\times}$	● Raw Point Input	Joint point interaction-dimension search with varying point densities	[49]
PIDS$_{2.0\times}$	● Raw Point Input	Enhanced PIDS structure with a larger segmentation backbone	[49]
PTv2	● Raw Point Input	Grouped vector attention & partition-based pooling using Transformers	[42]
WaffleIron	● Raw Point Input	Update point features by combining multi-MLPs and dense 2D CNNs	[32]
PointNet^{++}	● Raw Point Input	The first hierarchical network to direct operate on point clouds	[33]
DGCNN	● Raw Point Input	Use graph convolution to dynamically update graph in feature space	[40]
PAConv	● Raw Point Input	Dynamic kernel assembling to adjust convolutions with point positions	[46]

```
        self.b  = nn.Parameter(torch.zeros(1))
        self.alpha = 0.05
        self.threshold = threshold
        self.softmax = nn.Softmax(dim=-1)

    def forward(self, logits, gt, xyz):
        if self.training:
            ind = torch.argmax(logits, axis=1) == gt
            logits_pos, gt_pos = logits[ind], gt[ind]
            logits_neg, gt_neg = logits[~ind], gt[~ind]

            depth = torch.norm(xyz, p=2, dim=1)
            depth_pos, depth_neg = depth[ind], depth[~ind]

            s = np.random.randint(int(logits_pos.shape[0] * 1 / 3)) + 1
            logits = torch.cat((
                logits_neg, logits_pos[s:int(logits_pos.shape[0] / 2) + s]
            ), 0)
            gt = torch.cat((
                gt_neg, gt_pos[s:int(logits_pos.shape[0] / 2) + s]
            ), 0)
            depth = torch.cat((
                depth_neg, depth_pos[s:int(depth_pos.shape[0] / 2) + s]
            ), 0)

            prob = self.softmax(logits)

            score = torch.sum(-prob * torch.log(prob), dim=-1)
            cond_ind = score < self.threshold

            cal_logits_1, cal_gt_1 = logits[cond_ind], gt[cond_ind]
            cal_logits_2, cal_gt_2 = logits[~cond_ind], gt[~cond_ind]

            depth_coff = self.k * depth + self.b
            T1 = self.T1 * depth_coff[cond_ind].unsqueeze(dim=-1)
            T2 = self.T2 * depth_coff[~cond_ind].unsqueeze(dim=-1)

            cal_logits_1 = cal_logits_1 / T1
            cal_logits_2 = cal_logits_2 / T2

            cal_logits = torch.cat((cal_logits_1, cal_logits_2), 0)
            cal_gt = torch.cat((cal_gt_1, cal_gt_2), 0)

        else:
            prob = self.softmax(logits)
```

```python

        score = torch.sum(-prob * torch.log(prob), dim=-1)
        cond_ind = score < self.threshold

        scaled_logits, scaled_gt = logits[cond_ind], gt[cond_ind]
        inference_logits, inference_gt = logits[~cond_ind], gt[~cond_ind]

        depth = torch.norm(xyz, p=2, dim=1).to(logits.device)
        depth_coff = self.k * depth + self.b

        T1 = self.T1 * depth_coff[cond_ind].unsqueeze(dim=-1)
        T2 = self.T2 * depth_coff[~cond_ind].unsqueeze(dim=-1)

        scaled_logits = scaled_logits / T1
        inference_logits = inference_logits / T2

        cal_logits = torch.cat((scaled_logits, inference_logits), 0)
        cal_gt = torch.cat((scaled_gt, inference_gt), 0)

    return cal_logits, cal_gt
```

Listing 2. PyTorch-style code snippet of the proposed depth-aware scaling (DeptS).

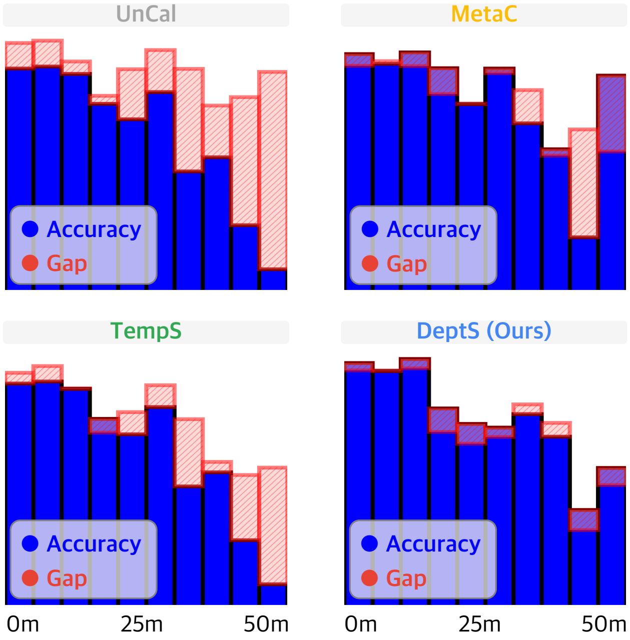


Figure A. Depth-wise confidence and accuracy statistics of uncalibrated (UnCal), temperature scaling (TempS), meta-calibration (MetaC), and our proposed depth-aware scaling (DeptS) methods.

C. Additional Quantitative Result

In this section, we supplement additional quantitative results to better support the findings and conclusions drawn in the main body of this paper.

C.1. Depth Correlations in LiDAR Data

As discussed in Sec. 3.3 of the main body of this paper, the motivation behind the depth-aware scaling method, DeptS, stems from some interesting observations from our experiments. We observe that traditional calibration techniques, effective in 2D image-based tasks, struggle with 3D data due to the unique characteristics of point clouds, such as being unordered and lacking texture. Through our analysis, we identified a clear correlation between calibration errors, prediction entropy, and depth. Specifically, as shown in Fig. A, LiDAR points at greater distances from the ego vehicle often exhibit lower accuracy, yet uncalibrated models maintain high confidence in these areas, leading to substantial calibration errors [20, 22]. This overconfidence in distant regions prompted the need for a tailored approach to address the depth-related calibration issue.

To tackle this, we propose DeptS, a method that adjusts model confidence based on depth information. By introducing a depth-correlation coefficient that reweights the temperature scaling parameters, DeptS reduces confidence for LiDAR points at larger depths, effectively mitigating the overconfidence problem. This method allows for better calibration in 3D scene understanding models, particularly in middle-to-far regions where predictions are less reliable, leading to improved calibration performance across diverse 3D datasets.

C.2. Reliability Diagrams

We provide additional reliability diagrams in Fig. B for a more comprehensive validation of the effectiveness of our method. As can be seen, the 3D models without proper calibration (UnCal) tend to suffer from huge confidence-accuracy gaps. This inevitably leads to potential impediments to the safe operation of 3D scene understanding systems in the real world. Our compared calibration methods show effectiveness in mitigating such issues. Compared to the previous calibration methods, our DeptS exhibits superior performance across a wide spectrum of scenarios. This can be credited to the depth-aware scaling operation which encourages a more consistent prediction in depth-correlated areas.

C.3. Domain-Shift Uncertainty Estimation

Enhancing the uncertainty estimation capability of handling challenging scenarios is crucial for the practical usage of 3D scene understanding systems [5, 6, 15, 18, 19, 21, 26, 44]. We supplement the domain-shift uncertainty estimation results of FRNet [47] and SPVCNN [37] in Tab. D and Tab. E, respectively. Similar to the observations drawn in the main body of this paper, we find that 3D models are vulnerable under adverse conditions. The expected calibration errors are extremely high under “fog”, “motion blur”, Table C. Comparisons between the proposed DeptS and state-of-the-art network calibration methods on the validation set of *SemanticKITTI* [3]. All ECE (the lower the better) and mIoU (the higher the better) scores reported are in percentage (%).

Method	**Venue**	**MinkUNet** [8]		**CENet** [7]	
		ECE	mIoU	ECE	mIoU
UnCal	-	3.04%	63.05%	5.95%	60.87%
TempS [14]	ICML'17	3.01%	63.05%	3.93%	60.87%
LogiS [14]	ICML'17	3.08%	63.11%	3.79%	60.95%
MetaC [29]	ICML'21	2.69%	62.93%	3.31%	60.81%
DeepEnsemble [23]	NeurIPS'17	2.99%	64.95%	5.61%	61.70%
BatchEnsemble [41]	ICLR'20	2.77%	64.70%	5.40%	62.13%
MIMO [16]	ICLR'21	3.21%	63.60%	6.97%	61.62%
PackedEnsemble [24]	ICLR'23	2.82%	63.88%	6.00%	59.81%
DeptS	**Ours**	**2.63%**	**63.47%**	**3.09%**	**61.20%**

"crosstalk", and "cross sensor" corruptions, which are commonly occurring scenarios in the real world. Compared to previous calibration methods like temperature scaling and meta-calibration, our DeptS shows a more stable performance across different domain-shift scenarios. We believe such an ability will become more and more important in the future development of 3D scene understanding systems.

C.4. Comparisons to Recent Calibration Methods

In the main body of this paper, we provide a comprehensive benchmark study of classical network calibration methods, such as TempS, LogiS, DiriS, and MetaC, across a range of ten different 3D datasets. The benchmark results verify that the proposed DepthS exhibits stronger performance compared to these classical approaches.

To provide a more holistic evaluation of DepthS compared to more recent network calibration methods, we conduct experiments with more recent network calibration methods, including DeepEnsemble [23], BatchEnsemble [41], MIMO [16], and PackedEnsemble [24], on the validation set of the SemanticKITTI [3] dataset. As shown in Tab. C, the results demonstrate that our proposed DeptS is consistently better than both the classical and recent network calibration methods.

D. Additional Qualitative Result

In this section, we supplement additional qualitative examples to better support the findings and conclusions drawn in the main body of this paper.

D.1. Visualized Calibration Results

We provide additional visualizations to help verify the effectiveness of the proposed model calibration model in enhancing the model's ability for uncertainty estimation. As can be seen from Fig. C and Fig. D, existing 3D scene understanding models often fail to deliver accurate uncertainty estimates, resulting in potential safety-related issues. Our proposed DeptS is capable of tackling these problems in a holistic manner. After calibration, models can generate more accurate uncertainty estimates, leading to a more reliable 3D scene understanding.

E. Limitation and Discussion

In this section, we elaborate on the limitations and potential negative societal impact of this work.

E.1. Potential Limitations

In this work, we established the first benchmark of 3D scene understanding from an uncertainty estimation viewpoint. We also proposed DeptS to effectively calibrate 3D models, achieving more reliable 3D scene understanding. We foresee the following limitations that could be promising future directions.

Data Dependence. Effective model calibration heavily relies on the quality and diversity of the data used. If the dataset is not representative of real-world scenarios or lacks diversity, the calibrated model may not generalize well across different environments or conditions.

Evaluation Challenges. Assessing the effectiveness of calibration can be challenging, as it requires comprehensive metrics that capture the model's performance across a broad range of scenarios. Standard evaluation metrics may not fully reflect the improvements in reliability and confidence achieved through calibration. It is enlightening to design new metrics for a more holistic evaluation.

E.2. Potential Societal Impact

3D scene understanding often involves capturing and analyzing detailed spatial data about environments which might include private spaces. Additionally, calibrated models might still inherit biases present in the data or algorithmic design, leading to unfair or discriminatory outcomes in certain scenarios. Addressing these issues requires more than technical solutions; it demands careful consideration of the ethical and societal implications of model deployment.

Table D. The expected calibration error (ECE, the lower the better) of FRNet [47] under eight domain-shift scenarios from *nuScenes-C* and *SemanticKITTI-C* in the *Robo3D* benchmark [18]. UnCal, TempS, LogiS, DiriS, MetaC, and DeptS denote the uncalibrated, temperature, logistic, Dirichlet, meta, and our depth-aware scaling calibration methods, respectively.

Type	**nuScenes-C**						**SemanticKITTI-C**					
	UnCal	TempS	LogiS	DiriS	MetaC	DeptS	UnCal	TempS	LogiS	DiriS	MetaC	DeptS
Clean ●	2.27%	2.24%	2.22%	2.28%	2.22%	2.17%	3.46%	3.53%	3.54%	3.49%	2.83%	2.75%
Fog ○	3.07%	3.06%	3.07%	3.03%	3.06%	2.98%	13.48%	13.57%	13.66%	13.47%	12.68%	12.42%
Wet Ground ○	2.46%	2.44%	2.43%	2.50%	2.56%	2.41%	4.01%	4.09%	4.11%	3.96%	3.32%	3.28%
Snow ○	3.50%	3.42%	3.53%	3.60%	2.93%	2.78%	7.28%	7.39%	7.49%	7.51%	6.65%	6.63%
Motion Blur ○	33.74%	33.48%	33.15%	32.15%	30.62%	28.43%	5.93%	6.03%	6.08%	6.55%	5.04%	4.92%
Beam Missing ○	2.52%	2.51%	2.50%	2.58%	2.91%	2.48%	2.71%	2.71%	2.72%	2.71%	2.40%	2.36%
Crosstalk ○	2.40%	2.39%	2.36%	2.38%	2.72%	2.35%	20.87%	21.16%	21.03%	19.84%	15.36%	14.79%
Incomplete Echo ○	2.36%	2.30%	2.32%	2.34%	2.28%	2.21%	3.77%	3.86%	3.88%	3.82%	3.13%	3.02%
Cross Sensor ○	5.24%	5.20%	5.29%	5.88%	5.34%	5.11%	5.08%	5.11%	5.17%	4.64%	3.91%	3.74%
Average ●	6.91%	6.85%	6.83%	6.81%	6.55%	6.09%	7.89%	7.99%	8.02%	7.81%	6.56%	6.40%

Table E. The expected calibration error (ECE, the lower the better) of SPVCNN [37] under eight domain-shift scenarios from *nuScenes-C* and *SemanticKITTI-C* in the *Robo3D* benchmark [18]. UnCal, TempS, LogiS, DiriS, MetaC, and DeptS denote the uncalibrated, temperature, logistic, Dirichlet, meta, and our depth-aware scaling calibration methods, respectively.

Type	**nuScenes-C**						**SemanticKITTI-C**					
	UnCal	TempS	LogiS	DiriS	MetaC	DeptS	UnCal	TempS	LogiS	DiriS	MetaC	DeptS
Clean ●	2.57%	2.44%	2.49%	2.54%	2.40%	2.31%	3.46%	2.90%	3.07%	3.41%	2.36%	2.32%
Fog ○	8.53%	8.12%	8.23%	8.54%	7.38%	7.34%	13.06%	12.33%	12.57%	13.23%	11.15%	11.10%
Wet Ground ○	2.80%	2.63%	2.68%	2.72%	2.63%	2.58%	3.52%	3.02%	3.19%	3.49%	2.76%	2.63%
Snow ○	8.49%	7.76%	7.97%	8.35%	6.87%	6.61%	8.50%	7.70%	7.94%	8.41%	6.31%	6.26%
Motion Blur ○	9.18%	8.80%	9.00%	9.33%	8.11%	7.98%	21.01%	19.92%	20.28%	20.41%	17.86%	17.22%
Beam Missing ○	2.88%	2.70%	2.74%	2.79%	2.72%	2.67%	3.01%	2.64%	2.73%	3.04%	2.48%	2.45%
Crosstalk ○	11.76%	11.09%	11.33%	12.01%	9.82%	9.48%	4.66%	4.00%	4.17%	4.49%	3.58%	3.31%
Incomplete Echo ○	2.40%	2.28%	2.33%	2.39%	2.30%	2.24%	3.54%	3.08%	3.24%	3.58%	2.56%	2.52%
Cross Sensor ○	4.80%	4.43%	4.52%	4.57%	4.22%	4.20%	3.27%	2.83%	2.96%	3.36%	2.81%	2.78%
Average ●	6.36%	5.98%	6.10%	6.34%	5.51%	5.39%	7.57%	6.94%	7.14%	7.50%	6.19%	6.03%

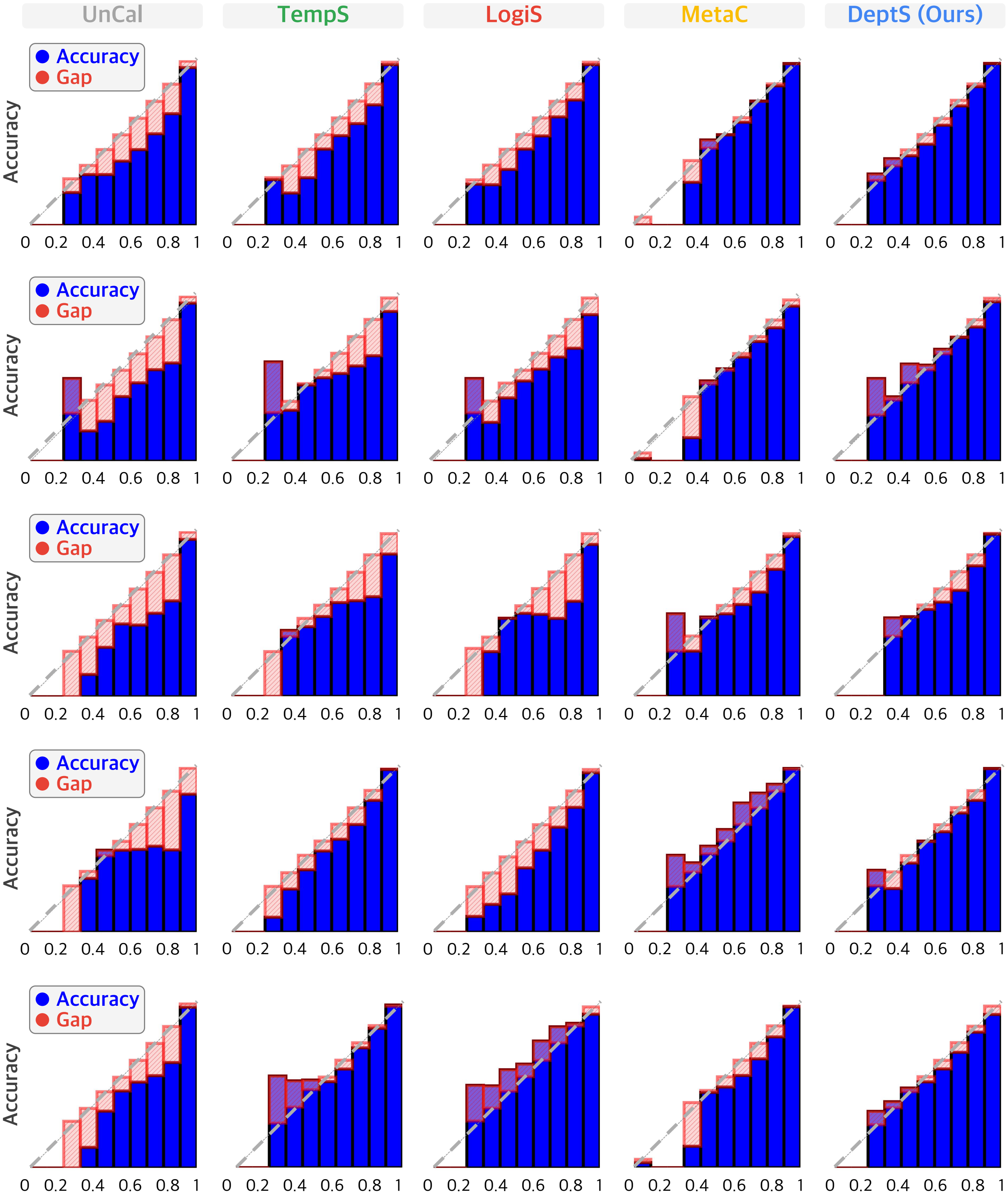


Figure B. The reliability diagrams of randomly sampled model predictions generated by the CENet [7] model on the validation set of the *SemanticKITTI* [3] dataset. UnCal, TempS, LogiS, MetaC, and DeptS denote the uncalibrated, temperature, logistic, meta, and our proposed depth-aware scaling calibration methods, respectively.

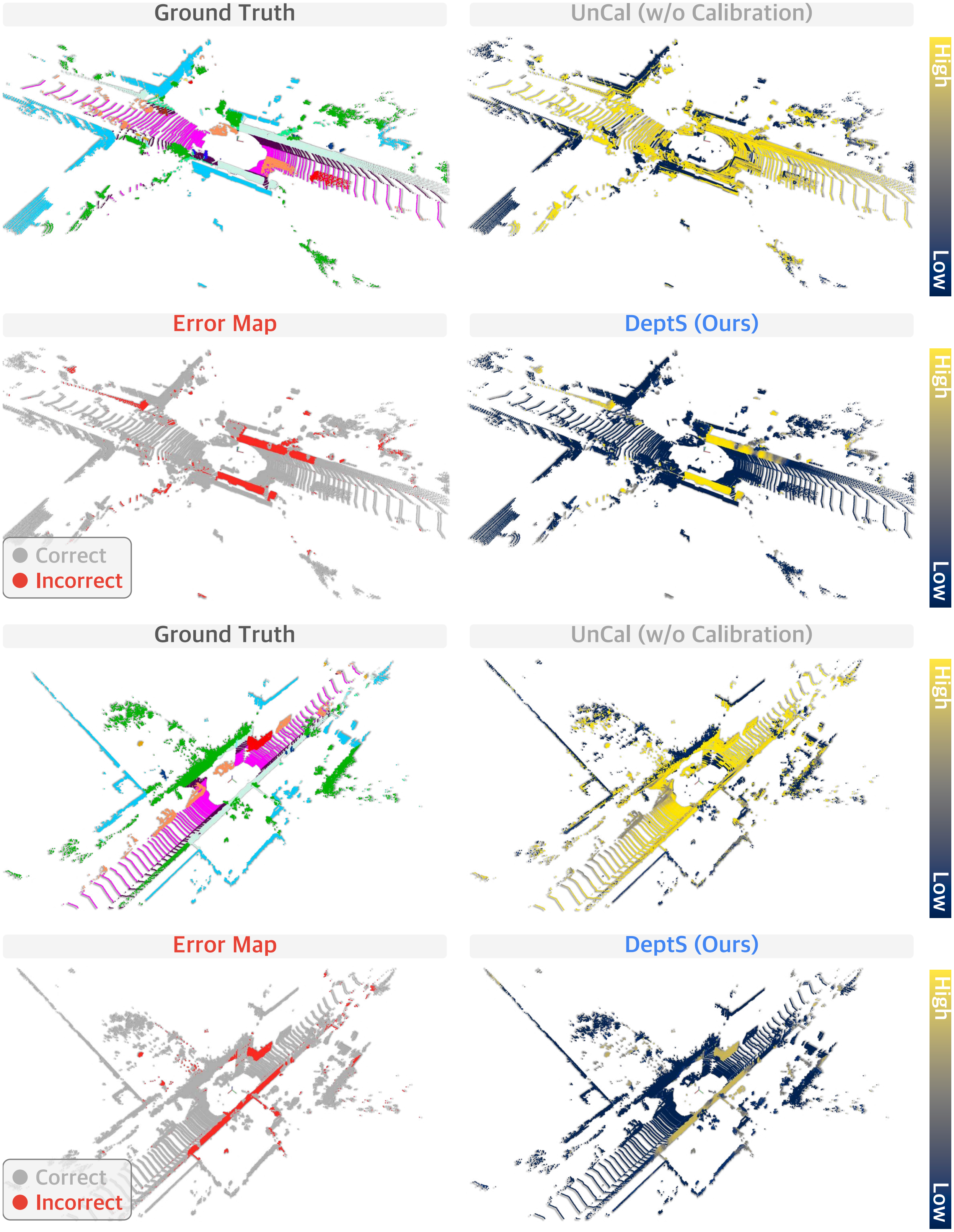


Figure C. The point-wise expected calibration error (ECE) of existing 3D semantic segmentation models without calibration (UnCal) and with our depth-aware scaling (DeptS). Our approach is capable of delivering accurate uncertainty estimates. The colormap goes from *dark* to *light* denotes *low* and *high* error rates, respectively.

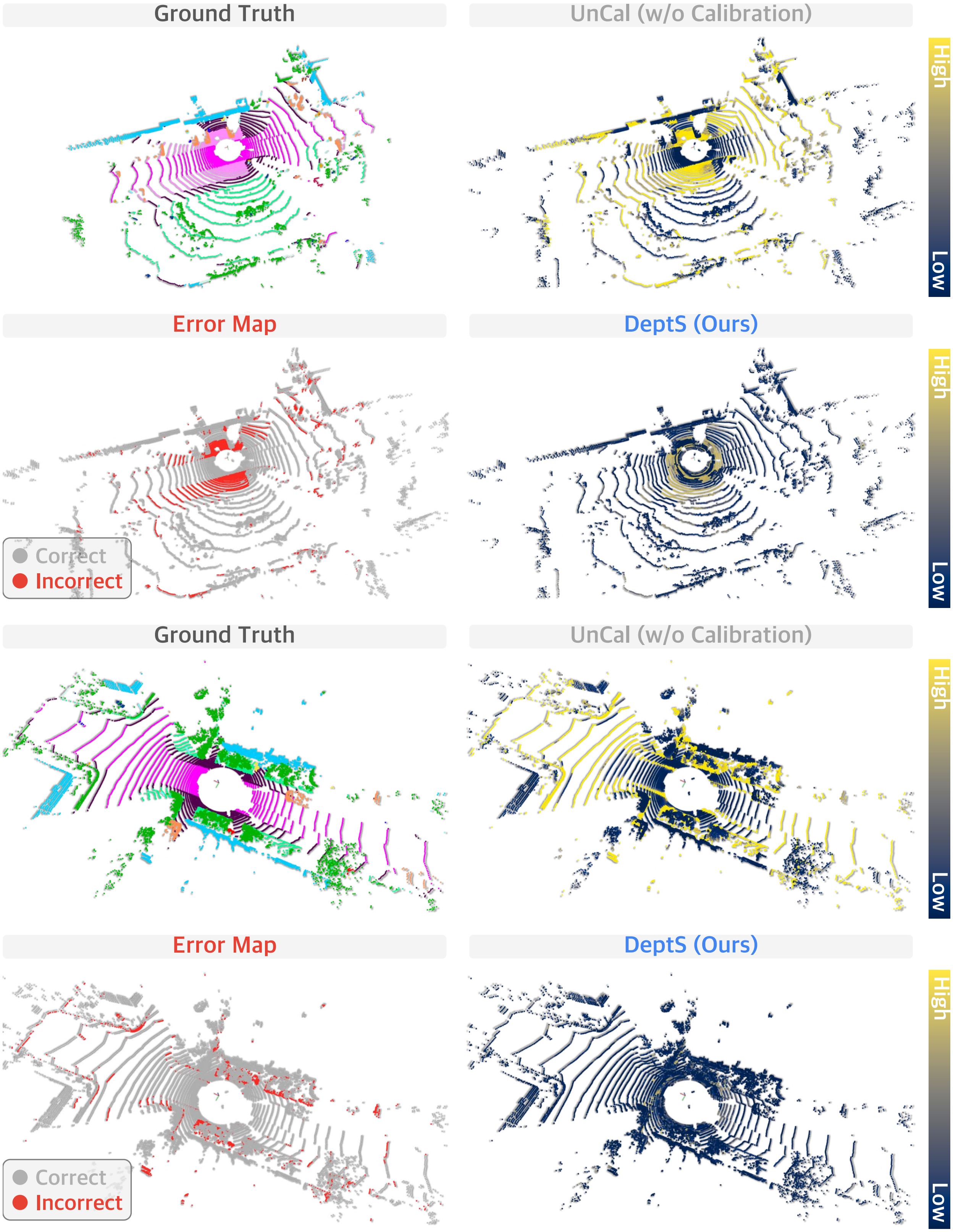


Figure D. The point-wise expected calibration error (ECE) of existing 3D semantic segmentation models without calibration (UnCal) and with our depth-aware scaling (DeptS). Our approach is capable of delivering accurate uncertainty estimates. The colormap goes from *dark* to *light* denotes *low* and *high* error rates, respectively.

F. Public Resources Used

In this section, we acknowledge the use of the following public resources, during the course of this work.

F.1. Public Codebase Used

We acknowledge the use of the following public codebase during this work:

MMCV[2]	Apache License 2.0
MMDetection[3]	Apache License 2.0
MMDetection3D[4]	Apache License 2.0
MMEngine[5]	Apache License 2.0
OpenPCSeg[6]	Apache License 2.0
Pointcept[7]	MIT License

F.2. Public Datasets Used

We acknowledge the use of the following public datasets during this work:

nuScenes[8]	CC BY-NC-SA 4.0
nuScenes-devkit[9]	Apache License 2.0
SemanticKITTI[10]	CC BY-NC-SA 4.0
SemanticKITTI-API[11]	MIT License
WaymoOpenDataset[12]	Waymo Dataset License
SemanticPOSS[13]	CC BY-NC-SA 3.0
Synth4D[14]	GPL-3.0 License
SemanticSTF[15]	CC BY-NC-SA 4.0
ScribbleKITTI[16]	Unknown
S3DIS[17]	Unknown
Robo3D[18]	CC BY-NC-SA 4.0

F.3. Public Implementations Used

We acknowledge the use of the following implementations during this work:

lidar-bonnetal[19]	MIT License
SalsaNext[20]	MIT License
FIDNet[21]	Unknown
CENet[22]	MIT License
rangevit[23]	Apache License 2.0
FRNet[24]	Apache License 2.0
PolarSeg[25]	BSD 3-Clause License
MinkowskiEngine[26]	MIT License
TorchSparse[27]	MIT License
SPVNAS[28]	MIT License
Cylinder3D[29]	Apache License 2.0
spconv[30]	Apache License 2.0
2DPASS[31]	MIT License
CPGNet[32]	Unknown
GFNet[33]	Unknown
KPConv[34]	MIT License
PIDS[35]	MIT License
PointTransformerV2[36]	Unknown

[2] https://github.com/open-mmlab/mmcv.

[3] https://github.com/open-mmlab/mmdetection.

[4] https://github.com/open-mmlab/mmdetection3d.

[5] https://github.com/open-mmlab/mmengine.

[6] https://github.com/PJLab-ADG/OpenPCSeg.

[7] https://github.com/Pointcept/Pointcept.

[8] https://www.nuscenes.org/nuscenes.

[9] https://github.com/nutonomy/nuscenes-devkit.

[10] http://semantic-kitti.org.

[11] https://github.com/PRBonn/semantic-kitti-api.

[12] https://waymo.com/open.

[13] http://www.poss.pku.edu.cn/semanticposs.html.

[14] https://github.com/saltoricristiano/gipso-sfouda.

[15] https://github.com/xiaoaoran/SemanticSTF.

[16] https://github.com/ouenal/scribblekitti.

[17] http://buildingparser.stanford.edu/dataset.html.

[18] https://github.com/ldkong1205/Robo3D.

[19] https://github.com/PRBonn/lidar-bonnetal.

[20] https://github.com/TiagoCortinhal/SalsaNext.

[21] https://github.com/placeforyiming/IROS21-FIDNet-SemanticKITTI.

[22] https://github.com/huixiancheng/CENet.

[23] https://github.com/valeoai/rangevit.

[24] https://github.com/Xiangxu-0103/FRNet.

[25] https://github.com/edwardzhou130/PolarSeg.

[26] https://github.com/NVIDIA/MinkowskiEngine.

[27] https://github.com/mit-han-lab/torchsparse.

[28] https://github.com/mit-han-lab/spvnas.

[29] https://github.com/xinge008/Cylinder3D.

[30] https://github.com/traveller59/spconv.

[31] https://github.com/yanx27/2DPASS.

[32] https://github.com/GangZhang842/CPGNet.

[33] https://github.com/haibo-qiu/GFNet.

[34] https://github.com/HuguesTHOMAS/KPConv.

[35] https://github.com/lordzth666/WACV23_PIDS-Joint-Point-Interaction-Dimension-Search-for-3D-Point-Cloud.

[36] https://github.com/Pointcept/PointTransformerV2.

- WaffleIron[37] .... Apache License 2.0
- selectivecal[38] .... Unknown
- LaserMix[39] .... CC BY-NC-SA 4.0
- PolarMix[40] .... MIT License

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5240–5250, 2023.

[2] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1534–1543, 2016.

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019.

[4] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020.

[5] Runnan Chen, Youquan Liu, Lingdong Kong, Nenglun Chen, Xinge Zhu, Yuexin Ma, Tongliang Liu, and Wenping Wang. Towards label-free scene understanding by vision foundation models. In *Advances in Neural Information Processing Systems*, volume 36, pages 75896–75910, 2023.

[6] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030, 2023.

[7] Huixian Cheng, Xianfeng Han, and Guoqiang Xiao. Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving. In *IEEE International Conference on Multimedia and Expo*, pages 1–6, 2022.

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[9] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020.

[10] Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022.

[11] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *International Symposium on Visual Computing*, pages 207–222, 2020.

[12] 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, 2017.

[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*, 7:3795–3802, 2022.

[14] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017.

[15] Xiaoshuai Hao, Mengchuan Wei, Yifan Yang, Haimei Zhao, Hui Zhang, Yi Zhou, Qiang Wang, Weiming Li, Lingdong Kong, and Jing Zhang. Is your hd map constructor reliable under sensor corruptions? In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[16] Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M. Dai, and Dustin Tran. Training independent subnetworks for robust prediction. In *International Conference on Learning Representations*, 2021.

[17] 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 *IEEE/CVF International Conference on Computer Vision*, pages 228–240, 2023.

[18] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robo3d: Towards robust and reliable 3d perception against corruptions. In *IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023.

[19] Lingdong Kong, Yaru Niu, Shaoyuan Xie, Hanjiang Hu, Lai Xing Ng, Benoit Cottereau, Ding Zhao, Liangjun Zhang, Hesheng Wang, Wei Tsang Ooi, Ruijie Zhu, Ziyang Song, Li Liu, Tianzhu Zhang, Jun Yu, Mohan Jing, Pengwei Li, Xiaohua Qi, Cheng Jin, Yingfeng Chen, Jie Hou, Jie Zhang, Zhen Kan, Qiang Lin, Liang Peng, Minglei Li, Di Xu, Changpeng Yang, Yuanqi Yao, Gang Wu, Jian Kuai, Xianming Liu, Junjun Jiang, Jiamian Huang, Baojun Li, Jiale Chen, Shuang Zhang, Sun Ao, Zhenyu Li, Runze Chen, Haiyong Luo, Fang Zhao, and Jingze Yu. The robodepth challenge: Methods and advancements towards robust depth estimation. *arXiv preprint arXiv:2307.15061*, 2023.

[20] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. Lasermix for semi-supervised lidar semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21705–21715, 2023.

[21] Lingdong Kong, Shaoyuan Xie, Hanjiang Hu, Lai Xing Ng, Benoit R. Cottereau, and Wei Tsang Ooi. Robodepth: Robust out-of-distribution depth estimation under corruptions.

[37] https://github.com/valeoai/WaffleIron.

[38] https://github.com/dwang181/selectivecal.

[39] https://github.com/ldkong1205/LaserMix.

[40] https://github.com/xiaoaoran/polarmix.

In *Advances in Neural Information Processing Systems*, volume 36, pages 21298–21342, 2023.

[22] Lingdong Kong, Xiang Xu, Jiawei Ren, Wenwei Zhang, Liang Pan, Kai Chen, Wei Tsang Ooi, and Ziwei Liu. Multimodal data-efficient 3d scene understanding for autonomous driving. *arXiv preprint arXiv:2405.05258*, 2024.

[23] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[24] Olivier Laurent, Adrien Lafage, Enzo Tartaglione, Geoffrey Daniel, Jean-Marc Martinez, Andrei Bursuc, and Gianni Franchi. Packed-ensembles for efficient uncertainty estimation. In *International Conference on Learning Representations*, 2023.

[25] Xiaoyan Li, Gang Zhang, Hongyu Pan, and Zhenhua Wang. Cpgnet: Cascade point-grid fusion network for real-time lidar semantic segmentation. In *IEEE International Conference on Robotics and Automation*, pages 11117–11123, 2022.

[26] Ye Li, Lingdong Kong, Hanjiang Hu, Xiaohao Xu, and Xiaonan Huang. Is your lidar placement optimized for 3d scene understanding? In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[27] Youquan Liu, Yeqi Bai, Lingdong Kong, Runnan Chen, Yuenan Hou, Botian Shi, and Yikang Li. Pcseg: An open source point cloud segmentation codebase. https://github.com/PJLab-ADG/PCSeg, 2023.

[28] Youquan Liu, Runnan Chen, Xin Li, Lingdong Kong, Yuchen Yang, Zhaoyang Xia, Yeqi Bai, Xinge Zhu, Yuexin Ma, Yikang Li, Yu Qiao, and Yuenan Hou. Uniseg: A unified multi-modal lidar segmentation network and the openpcseg codebase. In *IEEE/CVF International Conference on Computer Vision*, pages 21662–21673, 2023.

[29] Xingchen Ma and Matthew B. Blaschko. Meta-cal: Well-controlled post-hoc calibration by ranking. In *International Conference on Machine Learning*, pages 7235–7245, 2021.

[30] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4213–4220, 2019.

[31] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IEEE Intelligent Vehicles Symposium*, pages 687–693, 2020.

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

[33] Charles Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++ deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, pages 5105–5114, 2017.

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

[35] Cristiano Saltori, Evgeny Krivosheev, Stéphane Lathuiliére, Nicu Sebe, Fabio Galasso, Giuseppe Fiameni, Elisa Ricci, and Fabio Poiesi. Gipso: Geometrically informed propagation for online adaptation in 3d lidar segmentation. In *European Conference on Computer Vision*, pages 567–585, 2022.

[36] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[37] 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, 2020.

[38] 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 *IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019.

[39] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised lidar semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2697–2707, 2022.

[40] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019.

[41] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2020.

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

[43] Aoran Xiao, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, and Eric Xing. 3d semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9382–9392, 2023.

[44] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Benchmarking and improving bird's eye view perception robustness in autonomous driving. *arXiv preprint arXiv:2405.17426*, 2024.

[45] 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 *IEEE/CVF International Conference on Computer Vision*, pages 16024–16033, 2021.

[46] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3173–3182, 2021.

[47] Xiang Xu, Lingdong Kong, Hui Shuai, and Qingshan Liu. Frnet: Frustum-range networks for scalable lidar segmentation. *arXiv preprint arXiv:2312.04484*, 2023.

[48] Xu Yan, Jiantao Gao, Chao Zheng Chaoda 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, 2022.

[49] Tunhou Zhang, Mingyuan Ma, Feng Yan, Hai Li, and Yiran Chen. Pids: Joint point interaction-dimension search for 3d point cloud. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1298–1307, 2023.

[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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020.

[51] Yiming Zhao, Lin Bai, and Xinming Huang. Fidnet: Lidar point cloud semantic segmentation with fully interpolation decoding. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4453–4458, 2021.

[52] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9939–9948, 2021.