

# No Thing, Nothing: Highlighting Safety-Critical Classes for Robust LiDAR Semantic Segmentation in Adverse Weather

## Supplementary Material

In this supplementary material, we provide additional details and results not included in the main paper due to space constraints. The content is organized in the following order:

- Sec. [S1](#). summarizes the rationale of our methods, Feature Binding and Beam-wise Feature Distillation.
- Sec. [S2](#). provides results across various weather conditions.
- Sec. [S3](#). offers qualitative results on the SemanticPOSS-to-SemanticSTF benchmark.
- Sec. [S4](#). provides experimental results of the SynLiDAR-to-SemanticSTF benchmark.
- Sec. [S5](#). provides discussions about the effectiveness and limitations of our methods.
- Sec. [S6](#) analyzes the mIoU drop for the `bicyclist` and `fence` class.
- Sec. [S7](#). provides discussions about the performance degradation of `car` class on SemanticPOSS-to-SemanticSTF benchmark.
- Sec. [S8](#). provides related works on subclass- or prototype-based methods.
- Sec. [S9](#). provides additional experiments on different weather simulations.
- Sec. [S10](#). provides additional experiments on various datasets.
- Sec. [S11](#). provides additional experiments for different superclass criterions.
- Sec. [S12](#). provides additional examples illustrating the comparison between clean and corrupted data.
- Sec. [S13](#). provides failure cases of our methods.

### S1. Rationale

In this section, we provide a detailed explanation of the rationale behind the two proposed methods, as introduced in the main paper.

**Feature Binding.** As outlined in the main paper, Feature Binding (FB) aims to prevent *things* objects from being mispredicted as *stuff* classes. *Things* objects are often misclassified as *stuff* due to semantic-level corruption caused by weather perturbations. In such conditions, accurately predicting fine-grained classes (*e.g.* `person`, `motorcyclist`) with LiDAR Semantic Segmentation models becomes highly challenging. FB mitigates this issue by constraining features to visually similar superclasses, reducing *things*-to-*stuff* mispredictions.

**Beam-wise Feature Distillation.** As discussed in the main

paper, the goal of Beam-wise Feature Distillation (BFD) is to recover information lost due to missing points. BFD specifically addresses severe information loss in *things* objects by ensuring features from the augmented branch effectively capture *things* information from the clean branch. This approach aligns intact point patterns before the drop with collapsed point patterns after the drop, enabling efficient utilization of point pattern information.

**Why Divide Corruptions into Semantic and Local Levels?** This study categorizes corruptions based on the extent of point loss induced by weather corruption, which indicates potential degradation in data quality. A significant point loss can severely degrade semantic information. Thus, compensation strategies, such as Feature Binding, must be devised to mitigate this degradation at the semantic level. Even minimal point corruption can cause information loss. It disproportionately affects small-scale *things* objects. Beam-wise Feature Distillation is introduced to mitigate these corruptions.

**Why FB Benefits *things* Classes More?** Before applying our methodology, many misclassifications occurred between *things* and *stuff* classes. Thus, providing an additional discriminative signal via Feature Binding (FB) reduces misclassification between *things* and *stuff*. Second, we train the model to exploit the common point pattern in *things* classes through FB. This approach helps distinguish *things* from *stuff* because *things* share similar visual patterns, while *stuff* varies widely. Furthermore, as shown in Table 4, classwise prototypes do not improve performance. Therefore, our method’s effectiveness does not arise from alleviating class imbalance between *things* and *stuff*.

### S2. Results on Specific Weather Conditions

SemanticSTF consists of four adverse weather conditions: dense fog, light fog, rain, and snow. We examine the performance under each weather condition. As shown in Table S1, on the SemanticKITTI-to-SemanticSTF benchmark, we observe that our model produces robust results regardless of the weather condition. Additionally, Table S2 shows the results on the SemanticPOSS-to-SemanticSTF benchmark. Our proposed model achieves the highest mIoU in most weather conditions, including significant improvements in rain and snow scenarios. Therefore, our model demonstrates consistent performance in most adverse weather scenarios across both benchmarks.

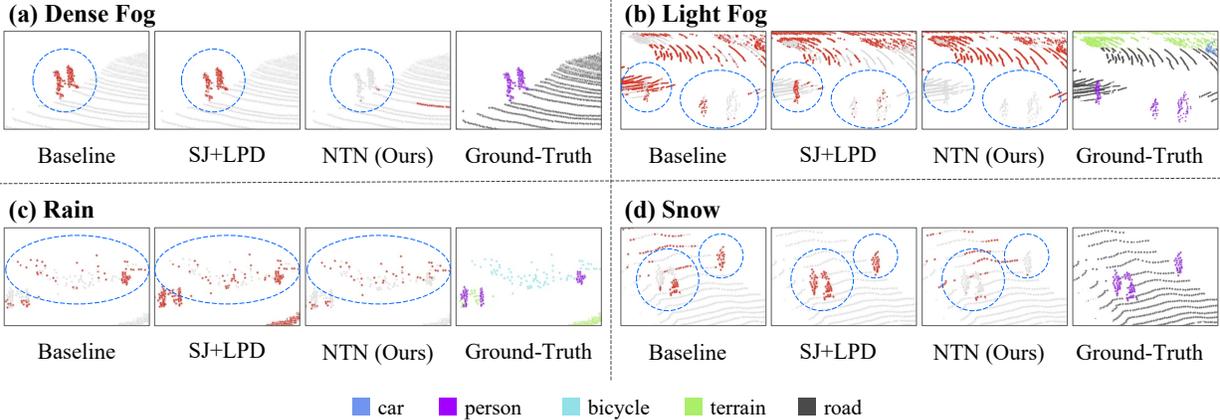


Figure S1. Qualitative results of our method on *validation set* of SemanticSTF. All models are trained on the *train set* of SemanticPOSS. Gray points indicate correct predictions, red points highlight errors, and ground-truth is shown with color-coded labels. Dashed circles highlight the predictions of *things* classes. Best viewed in color.

Method	D-fog	L-fog	Rain	Snow	mIoU
PolarMix [11]	29.7	25.0	28.6	25.6	27.2
PCL [13]	28.9	27.6	30.1	24.6	27.8
MMD [7]	30.4	28.1	32.8	25.2	29.1
PointDR [12]	31.3	29.7	31.9	26.2	29.8
DGLSS [6]	34.2	34.8	<b>36.2</b>	32.1	34.3
UniMix [14]	34.8	30.2	34.9	30.9	31.4
DGUIL [5]	<b>36.3</b>	34.5	35.5	<b>33.3</b>	34.8
SJ+LPD [9]	33.9	<b>35.5</b>	<u>35.8</u>	32.1	<u>36.3</u>
<b>NTN (Ours)</b>	<u>35.3</u>	<u>35.1</u>	35.7	<u>32.4</u>	<b>38.9</b>

Table S1. Performance comparison of different methods under varying weather conditions on SemanticKITTI-to-SemanticSTF benchmark. **Bold** indicates the best mIoU and underlined indicates the second-best performance.

Method	D-fog	L-fog	Rain	Snow	mIoU
PointDR [12]	26.2	30.1	50.1	<u>43.2</u>	34.7
DGLSS [6]	<b>32.8</b>	<u>34.8</u>	<u>54.9</u>	39.8	<u>39.2</u>
SJ+LPD [9]	25.4	30.6	38.2	40.6	38.3
<b>NTN (Ours)</b>	<u>31.8</u>	<b>38.7</b>	<b>58.4</b>	<b>50.4</b>	<b>46.2</b>

Table S2. Performance comparison of different methods under varying weather conditions on SemanticPOSS-to-SemanticSTF benchmark. **Bold** indicates the best mIoU and underlined indicates the second-best performance.

### S3. Qualitative Results on the SemanticPOSS-to-SemanticSTF benchmark

Fig. S1 shows qualitative results of our method and baselines on the SemanticSTF validation set, using SemanticPOSS as the source domain. Correct predictions are marked in gray, errors are in red, and ground-truth classes are color-coded. Our method demonstrates significant improvements over the original MinkowskiNet and SJ+LPD in segmenting the *things* category, particularly for classes such as *person* and *bicycle*. In scenarios (a), (b), and (d), our method accurately segments the *person* class, while others fail to recognize it or generate incomplete segmentation results. These results align with our quantitative analysis, which reveals a 15.5 IoU improvement for the *person* class. Additionally, in the rain scenario (c), while the segmentation performance of *bicycle* is not yet optimal, our method still surpasses existing methods, providing better segmentation results. Accurate segmentation of these *things* classes is highly significant, as they directly impact the safety and reliability of autonomous driving.

### S4. Experiment on SynLiDAR-to-SemanticSTF

We conduct additional experiments on the SynLiDAR-to-SemanticSTF benchmark. As shown in Table S3, our method improves the overall performance by 1.5 mIoU compared to the SJ+LPD model [9]. Although our proposed methods provide slight performance benefits, the improvements are not as significant as those observed in other benchmarks. SynLiDAR-to-SemanticSTF entangles both (1) the domain gap between synthetic and real data and (2) weather corruption. This is why our proposed meth-

Method	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	othe.g.	build.	fence	veget.	trunk	terra.	pole	traf.	mIoU
Oracle	89.4	42.1	0.0	59.9	61.2	69.6	39.0	0.0	82.2	21.5	58.2	45.6	86.1	63.6	80.2	52.0	77.6	50.1	61.7	54.7
Source-only	27.1	3.0	0.6	15.8	0.1	25.2	1.8	5.6	23.9	0.3	14.6	0.6	36.3	19.9	37.9	17.9	41.8	9.5	2.3	15.0
Dropout [10]	28.0	3.0	1.4	9.6	0.0	17.1	0.8	0.7	34.2	6.8	30.5	1.1	35.5	19.1	42.3	17.6	36.0	14.0	2.8	15.2
Perturbation	27.1	2.3	2.3	16.0	0.1	23.7	1.2	4.0	27.0	3.6	16.2	0.8	29.2	16.7	35.3	18.3	17.9	5.1	2.4	15.2
PolarMix [11]	39.2	1.1	2.2	8.3	1.5	17.8	0.8	0.7	23.3	1.3	17.5	0.4	45.2	24.8	46.2	20.1	38.7	10.9	0.6	15.7
MMD [7]	25.5	2.3	2.1	13.2	0.7	22.1	1.4	7.5	30.8	0.4	17.6	0.4	30.9	19.7	37.6	19.3	43.5	9.9	2.6	15.1
PCL [13]	30.9	0.8	1.4	10.0	0.4	23.3	4.0	7.9	28.5	1.3	17.7	1.2	39.4	18.5	40.0	18.0	38.6	12.1	2.3	15.5
PointDR [12]	37.8	2.5	2.4	23.6	0.1	26.3	2.2	7.7	27.9	7.7	17.5	0.5	47.6	25.3	45.7	21.0	37.5	17.9	5.5	18.5
DGLSS [6]	47.9	2.9	3.4	17.4	1.1	28.0	2.4	7.3	28.8	10.2	18.1	0.2	48.9	25.3	46.5	21.4	45.2	17.9	4.9	19.8
UniMix [14]	65.4	0.1	3.9	16.9	5.3	32.3	2.0	19.3	52.1	5.0	27.3	3.0	49.4	20.3	58.5	22.7	23.2	26.1	20.9	23.4
DGUIL [5]	43.3	2.8	2.6	23.2	3.2	31.3	2.5	4.4	34.3	9.2	17.9	0.3	57.1	27.6	50.0	24.2	41.5	19.0	6.1	21.1
SJ+LPD [9]	39.0	2.5	2.5	22.3	0.3	27.0	1.8	4.0	36.1	10.3	19.0	1.0	50.6	24.5	45.1	23.2	34.1	21.9	7.2	19.6
<b>NTN (Ours)</b>	48.4	1.5	2.4	19.4	0.2	29.1	3.2	8.9	43.5	6.7	20.5	0.0	52.2	30.1	49.8	20.0	32.9	24.7	7.5	21.1
<b>↑ to SJ+LPD</b>	<b>(+9.4)</b>	<b>(-1.0)</b>	<b>(-0.1)</b>	<b>(-2.9)</b>	<b>(-0.1)</b>	<b>(+2.1)</b>	<b>(+1.4)</b>	<b>(+4.9)</b>	<b>(+7.4)</b>	<b>(-3.6)</b>	<b>(+1.5)</b>	<b>(-1.0)</b>	<b>(+1.6)</b>	<b>(+5.6)</b>	<b>(+4.7)</b>	<b>(-3.2)</b>	<b>(-1.2)</b>	<b>(+2.8)</b>	<b>(+0.3)</b>	<b>(+1.5)</b>

Table S3. Comparison of methods on the SynLiDAR-to-SemanticSTF benchmark. Performance improvements of our method over SJ+LPD [9] are shown in the last row, with red text indicating *things* classes increments and green text for *stuff*.

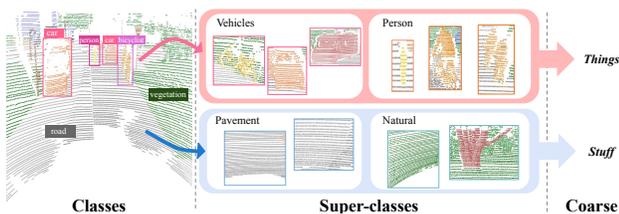


Figure S2. Examples of the three levels of category classification: classes, superclasses, and coarse, in increasing order of granularity.

ods provide marginal performance gain. Nevertheless, Table S3 shows FB and BFD outperform the previous method, SJ+LPD [9], implying its effectiveness in representing real adverse weather. Note that UniMix [14] and DGUIL [5] have not been reproduced due to the lack of publicly available code.

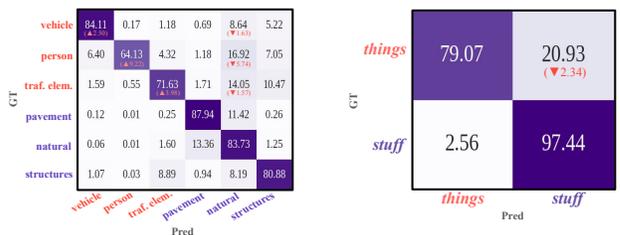
## S5. Discussions about Methods

In this section, we discuss both the effectiveness and limitations of our proposed modules on the SemanticKITTI-to-SemanticSTF benchmark.

### S5.1. Analysis on Feature Binding

**Multi-Level Categories.** We divide categories into three levels, as shown in Fig. S2. Classes are the smallest units, superclasses group semantically similar classes, and coarse categories consist of *things* and *stuff*. We perform a detailed analysis of these three levels using confusion matrices and qualitative results.

**Effect of FB for Relieving Confusion.** As mentioned in Sec. S1, Feature Binding is a module designed to prevent



(a) Confusion Matrix at the Superclass Level (b) Confusion Matrix at the Coarse Level

Figure S3. Confusion matrices for (a) superclasses and (b) *things* and *stuff* after adding Feature Binding. The numbers in brackets indicate the increase or decrease compared to SJ+LPD [9]. FB effectively reduces misclassification between *things* and *stuff*.

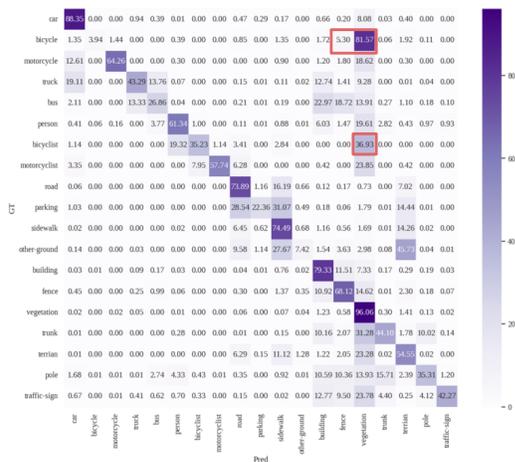
*things* from being misclassified as *stuff* when their semantic information is partially degraded due to corruption caused by adverse weather. To examine the effect of FB, we conduct an analysis at the superclass level and the coarse level, as shown in Fig. S3. When we compare the performance with the previous method [9], we find that misclassification ratio of *things* as *stuff* decreases by 2.34%. At the superclass level, we observe that the tendency to misclassify *things* as *natural* categories such as vegetation is greatly reduced. In particular, for the *person* category, true positive prediction ratio increases by 9.22% compared to before, and misclassification ratio as the *natural* category decreases by 5.74%. Thus, we confirm that FB helps to learn the semantic differences between *things* and *stuff* by continuously providing hierarchical semantic information.

### S5.2. Analysis on Beam-wise Feature Distillation.

**Challenges of BFD with Disrupted Point Patterns.** As in Fig. S4, misprediction ratio from things-to-stuff slightly increases in bicycle and bicyclist classes. This is



(a) Class-wise Confusion Matrix of SJ+LPD



(b) Class-wise Confusion Matrix of NTN (Ours)

Figure S4. Class-wise confusion matrices of (a) SJ+LPD [9] and (b) Ours on SemanticKITTI-to-SemanticSTF benchmark.

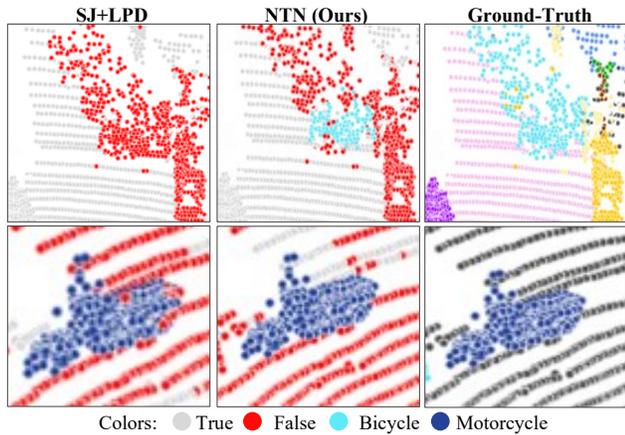


Figure S5. Qualitative results of NTN (FB + BFD). Gray is true prediction, and Red is false prediction. Other colors mean true predictions for specific classes.

Feature Distillation Types		Performance		
$\mathcal{B}$ -all.	$\mathcal{B}$ -wise. (Ours)	Things	Stuff	mIoU
✓	✗	35.3	43.7	38.9
✗	✓	<b>36.2</b>	<b>42.5</b>	<b>38.9</b>

Table S4. Comparison for feature distillation settings between average on all beams and separate beam. Experiments are done on SemanticKITTI-to-SemanticSTF benchmark.

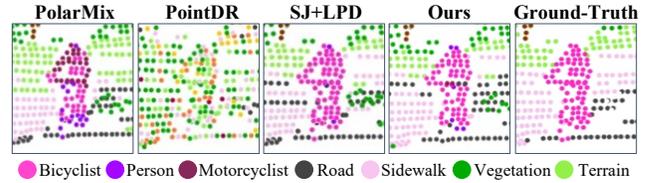


Figure S6. Qualitative results on bicyclist class for PolarMix [11], PointDR [12], SJ+LPD [9] and Ours.

because BFD tends to effectively utilize the point patterns before point drop, as mentioned in Sec. S1.

The learning mechanism of BFD enables the LSS model to perform better on objects where the clean branch’s point pattern is well preserved during test time. This explains why our method struggles with classes like bicycle and bicyclist, which inherently have few points and are prone to severe point pattern corruptions. In such cases, BFD struggles to utilize informations of local point pattern from clean data branch. Conversely, significant performance improvements are observed for classes like motorcycle and motorcyclist, which have clearer shapes. As shown in Fig. S5, the method with BFD demonstrates superior predictions for bicycles with reasonably preserved point patterns, as observed in row 1. Also, the motorcycle in row 2, which maintains its point pattern, achieves improved performance. In summary, confusion on classes like bicycle arises as a side effect of improving performance by preserving things object information without bias toward weather-corrupted data.

**Ablation Study on Types of Feature Distillation.** As shown in Table S4, distillation after averaging across all beams ( $\mathcal{B}$ -all.) achieves the same mIoU as our method, which performs beam-wise averaging before distillation ( $\mathcal{B}$ -wise.). However, our method outperforms  $\mathcal{B}$ -all. in things objects. This demonstrates that defining beam-wise local regions and performing distillation accordingly better compensates for information loss caused by point missing in things objects.

Dataset	Sem.KITTI; w. Other Sim.			Dataset	nuScenes; real rain			nuScenes-C; snow, fog, wet ground					
Method	Things	Stuff	All	Method	Things	Stuff	All	Heavy	Moderate	Light	Things	Stuff	All
Source Only	17.7	38.9	26.6	Source Only	34.2	67.0	48.6	47.5	52.5	55.1	39.0	68.0	51.7
[3] + [4]	22.9	42.8	31.3	SJ+LPD	33.2	66.6	47.8	49.0	53.6	55.7	40.7	68.2	52.7
+ NTN (Ours)	<b>23.6</b>	<b>43.9</b>	<b>32.2</b>	+ NTN (Ours)	<b>35.4</b>	<b>67.8</b>	<b>49.6</b>	<b>48.6</b>	<b>54.0</b>	<b>56.6</b>	<b>53.1</b>	<b>68.5</b>	<b>53.1</b>
↑ to [3] + [4]	(+0.7)	(+1.1)	(+0.9)	↑ to SJ+LPD	(+2.2)	(+1.2)	(+1.8)	(+2.2)	(+1.9)	(+1.7)	(+12.4)	(+0.3)	(+0.4)

Table S5. Comparison on various benchmarks and scenarios.

Method	Superclass	mIoU	Total
Coarse	<b>Things:</b> person, vehicle, traffic element <b>Stuff:</b> pavement, natural, structure	33.7 (↑2.4) 44.0 (↑0.9)	<b>38.1</b> (↑1.8)
GPT-o1	<b>Person:</b> person, bi.clst, mt.clst <b>Vehicle:</b> car, bi.cle, mt.cle, truck, oth-v. <b>Ground:</b> road, parki., sidew., othe.g. <b>Nature:</b> trunk, veget., terra. <b>Construction:</b> build., fence <b>Traffic Object:</b> pole, traf.	39.1 (↑13.4) 33.5 (↑3.4) 30.0 (↓0.8) 48.6 (↑0.4) 57.3 (↓6.8) 31.5 (↓2.0)	<b>38.3</b> (↑2.0)
DeepSeek-V3	<b>Person:</b> person, bi.clst, mt.clst <b>Vehicle:</b> car, bi.cle, mt.cle, truck, oth-v. <b>Road-infra.:</b> road, parki., sidew., othe.g. <b>Nature:</b> veget., terra. <b>Man-made:</b> build., fence, pole, traf. <b>Miscell.:</b> trunk	30.0 (↑4.4) 37.3 (↑7.2) 32.1 (↑1.3) 56.6 (↑2.9) 49.1 (↑0.4) 35.1 (↓2.1)	<b>39.4</b> (↑3.1)

Table S6. Comparison of different superclasses on SemanticKITTI-to-SemanticSTF (improvements over SJ+LPD in brackets).

## S6. Analysis on mIoU Drop Bicyclist & Fence Classes.

**Bicyclist.** We found out that both the predictions from SJ+LPD [9] and our method accurately detected the bicyclist within a single scan. As shown in the third and fourth visualizations of Fig. S6, the performance difference between SJ+LPD and our method is minimal, amounting to only a few points. Furthermore, the mispredicted points were all classified as a person, indicating that the slight performance drop does not significantly impact safety-critical predictions.

We verified the results for this object using PointDR and PolarMix, which are reproducible due to publicly available codes. As illustrated in the first and second visualizations of Fig. S6, both PolarMix [11] and PointDR [12] failed to provide accurate predictions for most bicycle points. In PolarMix, some points are predicted as bicyclist, but many are misclassified as motorcyclist. As motorcyclist generally moves faster than bicyclist or person, such mispredictions could lead to significant risks in ensuring safe driving. For PointDR, none of the points are predicted as person, bicyclist, or motorcyclist. This demonstrates that previous methods fail to predict even within the person superclass, highlighting the effectiveness of our method in safety-critical driving scenarios.

**Fence.** Unlike other thin traffic elements, fence varies from thin to large structures. Due to this structural di-

versity, large fences receive confusing signals from the distinct-shaped superclass by FB. As a result, fence had lower IoU and increased confusion with building and vegetation as in Fig. S4.

## S7. Performance Degradation of Car Class on SemanticPOSS→SemanticSTF.

The performance gap arises from differences in car class annotation between SemanticKITTI and SemanticPOSS. SemanticKITTI-trained model benefits from FB by learning diverse vehicle classes separately, thus effective in SemanticSTF. In contrast, SemanticPOSS groups car, bus, and truck into one class, limiting feature diversity and lowering performance. For optimal performance of FB, pre-defined classes should be as fine-grained as possible.

## S8. Related Works on Subclass- or Prototype-based Methods

Subclass- or Prototype-based regularization has frequently been employed to tackle label-efficiency problems, where limited or imbalanced annotations lead to suboptimal segmentation. Pixel-to-Prototype Contrast [1] addresses weakly supervised semantic segmentation by aligning pixel embeddings with class prototypes, refining noisy pseudo masks generated from image-level labels. Similarly, Prototypical Contrastive Network [8] focuses on highly imbalanced aerial segmentation by learning a single foreground

prototype and pushing away hard-negative background features, thus emphasizing minority classes. Unbiased Subclass Regularization [2] aims to mitigate semi-supervised class imbalance by splitting overrepresented classes into smaller clusters, forming class-balanced subclasses. Despite sharing the general goal of improving segmentation via prototype or class-based groupings, our work diverges in both methodology and application. Instead of subdividing a single class or maintaining per-class prototypes, we merge multiple classes into higher-level *superclasses* (e.g., *things* vs. *stuff*), which better addresses the broad semantic gap under adverse weather in LiDAR data. Moreover, prior methods largely concentrate on 2D image tasks or semi-/weakly supervised settings, whereas we focus on single-domain generalization for 3D LiDAR segmentation, emphasizing robustness against severe weather-induced corruptions.

## S9. Additional Experiments on Different Weather Simulations.

We used simulation methods [3, 4] to generate occlusion-induced points, improving performance (Table S5) and confirming robustness to other weather simulations. Similar to Table 2, the performance gain was higher in *things* (+0.7 mIoU), highlighting our method’s safety benefits.

## S10. Additional Experiments on Various Datasets

We tested our method on two settings: (1) nuScenes *clean train set-to-rainy splits*, and (2) nuScenes *train set-to-nuScenes-C*. As shown in Table S5, our method consistently surpasses SJ+LPD [9] across diverse datasets, particularly for *things*.

## S11. Additional Experiments for Different Superclass.

We evaluated the impact of superclass selection on generalizability by: (1) separating superclasses into *things* and *stuff*, (2) using GPT-o1, and (3) Deepseek-V3 recommendations. For (2) and (3), we provide COCO and ImageNet supercategory examples. Table S6 shows that even (1) helps FB reduce *things*→*stuff* mispredictions and improves performance, while finer superclasses in (2) and (3) achieve higher gains. This demonstrates well-chosen superclasses can boost performance and highlight the scalability of our method. While our method consistently improves *things*, future work is needed for automatic superclass selection.

## S12. Additional Visual Comparisons Between Clean and Corrupted Data

We provide additional comparisons between clean and corrupted data, as shown in Fig. S7. It illustrates point patterns of *things* objects in clean and weather-induced corrupted data. As mentioned in Sec. 1 and Sec. 3 in the main paper, *things* objects in clean weather have well-defined shapes and smooth boundaries, whereas in adverse weather, they have blurred shapes and irregular boundaries. This proves that *things* classes are more vulnerable to such noise or point loss, making accurate predictions more difficult.

## S13. Failure Cases

Fig. S8 illustrates cases where our method fails in prediction. Errors remain, particularly at very close distances to vehicles, as shown in (a) and (e). As demonstrated in (a), performance improvements were limited for the *car* class due to extreme sparsity from occlusion by droplets. For the *person* class, (c) shows yet many incorrect predictions remain in extremely noisy conditions. Thin objects like *pole* and *traffic-sign* are often misclassified as vegetation, as seen in cases (b), (d), (e), (f) and (h). Errors also occur in classes with varying object sizes, such as fences in examples (g) and (h). As discussed in Sec. S6, this issue arises from defining FB superclasses manually.

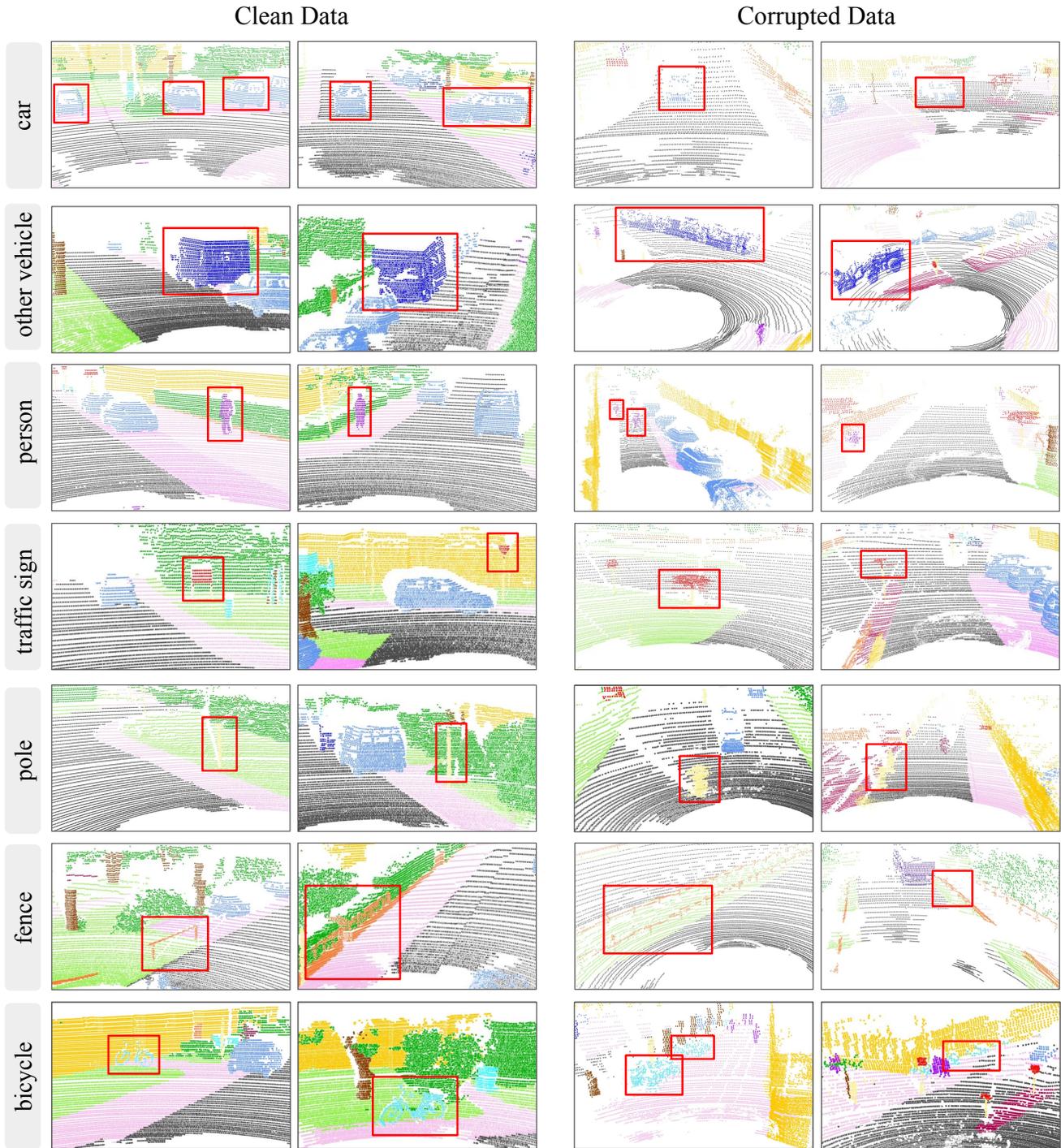


Figure S7. Examples of clean and corrupted data for various *things* classes. In clean weather, objects have well-constructed shapes with dense point clouds. In contrast, objects in adverse weather have blurred shapes with significant point loss.

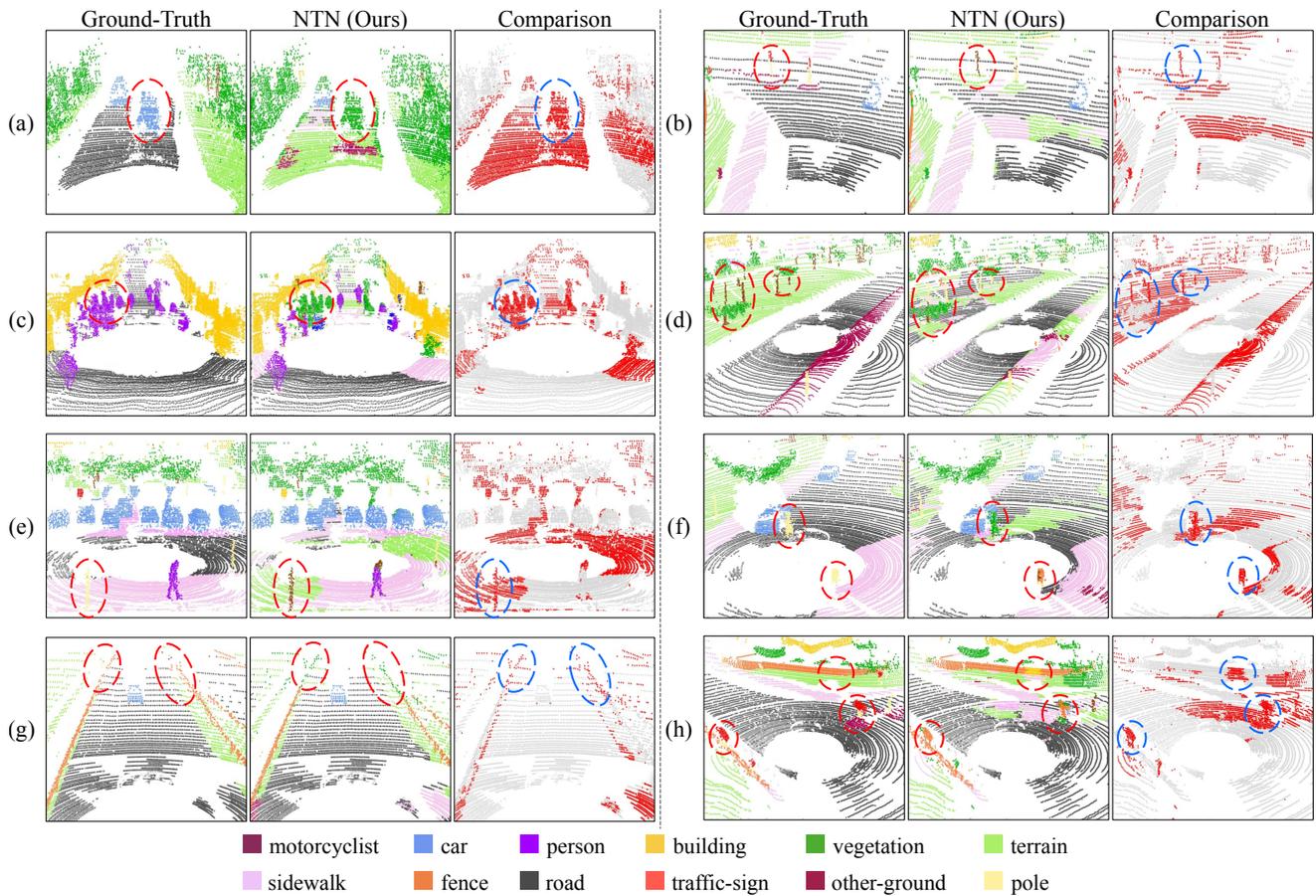


Figure S8. Qualitative results of failure cases on the *validation set* of SemanticSTF. Ground-truth and our prediction results are shown with color-coded labels. For comparison, gray points indicate correct predictions and red points highlight errors.

## References

- [1] Ye Du, Zehua Fu, Qingjie Liu, and Yunhong Wang. Weakly supervised semantic segmentation by pixel-to-prototype contrast, 2022. 5
- [2] Dayan Guan, Jiaxing Huang, Aoran Xiao, and Shijian Lu. Unbiased subclass regularization for semi-supervised semantic segmentation, 2022. 6
- [3] Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real lidar point clouds for 3d object detection in adverse weather. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15283–15292, 2021. 5, 6
- [4] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. Lidar snowfall simulation for robust 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16364–16374, 2022. 5, 6
- [5] Pei He, Licheng Jiao, Lingling Li, Xu Liu, Fang Liu, Wenping Ma, Shuyuan Yang, and Ronghua Shang. Domain generalization-aware uncertainty introspective learning for 3d point clouds segmentation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 651–660, 2024. 2, 3
- [6] Hyeonseong Kim, Yoonsu Kang, Changgyoon Oh, and Kuk-Jin Yoon. Single domain generalization for lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17587–17598, 2023. 2, 3
- [7] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018. 2, 3
- [8] Keiller Nogueira, Mayara Maezano Faima-Pinheiro, Ana Paula Marques Ramos, Wesley Nunes Gonçalves, José Marcato Junior, and Jefersson A Dos Santos. Prototypical contrastive network for imbalanced aerial image segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 8366–8376, 2024. 5
- [9] Junsung Park, Kyungmin Kim, and Hyunjung Shim. Re-thinking data augmentation for robust lidar semantic segmentation in adverse weather. In *European Conference on Computer Vision*. Springer, 2025. 2, 3, 4, 5, 6
- [10] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 3
- [11] Aoran Xiao, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, and Ling Shao. Polarmix: A general data augmentation technique for lidar point clouds. *Advances in Neural Information Processing Systems*, 35:11035–11048, 2022. 2, 3, 4, 5
- [12] Aoran Xiao, Jiaxing Huang, Weihao Xuan, Ruijie Ren, Kangcheng Liu, Dayan Guan, Abdulmotaleb El Saddik, Shijian Lu, and Eric P Xing. 3d semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9382–9392, 2023. 2, 3, 4, 5
- [13] Xufeng Yao, Yang Bai, Xinyun Zhang, Yuechen Zhang, Qi Sun, Ran Chen, Ruiyu Li, and Bei Yu. Pcl: Proxy-based contrastive learning for domain generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7097–7107, 2022. 2, 3
- [14] Haimei Zhao, Jing Zhang, Zhuo Chen, Shanshan Zhao, and Dacheng Tao. Unimix: Towards domain adaptive and generalizable lidar semantic segmentation in adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14781–14791, 2024. 2, 3