3D Semantic Segmentation in the Wild: Learning Generalized Models for Adverse-Condition Point Clouds –Supplementary Material–

Aoran Xiao[1], Jiaxing Huang[1], Weihao Xuan[2], Ruijie Ren[3], Kangcheng Liu[1]
Dayan Guan[4], Abdulmotaleb El Saddik[4,6], Shijian Lu[1], Eric Xing[4,5]
[1]Nanyang Technological University [2]Waseda University [3]Technical University of Denmark
[4]Mohamed bin Zayed University of Artificial Intelligence
[5]Carnegie Mellon University [6]University of Ottawa

We provide more experiment details of domain adaptation and domain generalization in Section A and Section B, respectively, supervised learning on adverse conditions in Section C and additional details on SemanticSTF dataset in Section D.

A. Domain generalization

A.1. Implementation details

We provide the detailed training configurations for semantic segmentation of LiDAR point clouds that have been adopted as described in Sec. 5.1 of the submitted paper. Specifically, we implement the backbone model MinkowskiNet [2] with the TorchSparse library [11]. For training, we use SGD optimizer. The learning rate, momentum and weight decays are set as $0.24, 0.9$, and $1.4e-4$, respectively. τ in Eq. 1 in the paper is set as 0.07 [4, 15] and λ_{ct} in Eq. 2 is set as 0.1. The momentum coefficient m is set at 0.99. We train 50 epochs with one NVIDIA 2080Ti with 11GB GPU memory and set the batch size at 4. The augmentations of training data in the source-domain are implemented as follows: For *rotation*, LiDAR points are rotated with the range of $[0, 360°]$ along Z axis. For *scale*, the coordinates of LiDAR points are randomly scaled within $[0.95, 1.05]$. For *drop-out*, we randomly drop-out 0-20% points of input LiDAR scans with a probability of 0.5. As for *noise perturbation*, $0-2,000$ random points are added into the 3D space of each LiDAR scan with a probability of 0.5. When using *flipping*, we randomly flip coordinates of LiDAR point clouds along x or y axis with a probability of 0.5. As for *jittering*, random coordinate shifts with a range of $[-0.05, 0.05]$ meters are added into LiDAR points with a probability of 0.5.

In training the oracle model, we employ the SGD optimizer with the hyperparameters including initial learning rate at 0.1, momentum at 0.9, weight decay at $1.0e-4$, and dampening at 0.1. We train the segmentation model with 500 epochs using a single NVIDIA 2080Ti with 11GB GPU memory. The batch size is set as 4. We use Poly learning rate policy with power$= 0.9$. As for data augmentations, we follow [12] and adoptes random rotation ($[-\pi, \pi]$) and scaling ($[0.95, 1.05]$); We also adopts PolarMix [16] with following parameter settings: Rotation angles along the Z-axis, denoted as Ω, are randomly scaled within normal distributions with a mean of $\mu = 0$ and standard deviation of $\sigma = \frac{2}{3}\pi$. We keep the original instance classes for rotate-pasting in PolarMix.

A.2. Evaluation of individual adverse weather conditions

We noticed that for certain individual adverse weather conditions, some class has no data captured in the validation set of SemanticSTF. Specifically, there are no points of *bicycle* and *motorcycle* in the validation set of *dense fog*; no points of *bicyclist* and *motorcyclist* in the validation set of *snow*, and no *bicycle* and *motorcyclist* in the validation set of *rain*. This is reasonable as the LiDAR data of SemanticSTF is collected in European countries including Germany, Sweden, Denmark, and Finland where motorcycles are not widely used for the reason of environmental protection. In addition, people usually do not ride bicycles or motorcycles in adverse weather conditions. As a result, classes *motor*, *motorcyclist*, and *bicyclist* have extremely lower occurrence frequency, leading to an absence of these classes in the validation set of SemanticSTF under relevant weather conditions. Tables 1, 2, 3, and 4 present corresponding class-level IoU performance for each adverse weather in Table 3 of the submitted paper.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.
SemanticKITTI→SemanticSTF(*dense fog*)																			
Baseline	74.7	-	-	7.8	0.0	6.4	8.9	0.0	72.2	0.6	33.8	0.0	59.6	48.7	56.9	27.4	56.4	27.2	21.1
Dropout [10]	67.5	-	-	1.9	0.0	8.9	2.8	0.0	70.9	5.6	29.0	0.8	64.6	44.0	60.0	31.6	60.6	28.1	21.3
Perturbation	68.6	-	-	8.8	0.0	6.0	0.0	0.0	66.6	14.8	24.3	0.1	52.2	43.5	60.1	19.4	54.1	16.3	11.5
PolarMix [16]	52.3	-	-	17.2	0.0	3.6	0.0	19.3	75.2	0.0	28.7	0.6	62.4	49.5	60.5	29.0	55.4	20.8	30.7
MMD [6]	75.5	-	-	0.3	0.0	4.2	0.0	0.0	75.4	11.2	33.6	0.5	64.8	51.7	64.7	26.1	62.3	23.0	23.0
PCL [18]	64.3	-	-	11.7	0.0	0.6	0.0	0.0	72.4	3.8	31.3	0.8	63.1	46.5	65.7	19.4	64.3	18.5	28.9
PointDR (Ours)	69.2	-	-	7.1	0.0	2.4	6.7	0.0	73.5	8.5	33.6	0.2	65.6	47.6	63.6	31.0	60.7	24.4	38.8
SynLiDAR→SemanticSTF(*dense fog*)																			
Baseline	21.6	-	-	6.4	0.0	3.7	2.9	18.9	25.7	0.0	7.7	1.0	41.2	22.5	52.3	15.4	55.5	9.3	2.4
Dropout [10]	12.7	-	-	7.7	0.0	1.9	0.4	2.5	38.3	0.1	10.2	0.3	37.3	21.8	57.4	13.1	44.5	10.1	1.0
Perturbation	13.3	-	-	10.4	0.0	4.3	2.8	19.1	30.0	0.7	8.8	1.2	30.5	17.5	48.9	18.4	50.3	16.3	5.2
PolarMix [16]	15.8	-	-	10.6	0.0	1.5	1.7	3.5	27.7	0.0	9.9	0.3	46.2	28.9	59.2	13.5	49.5	4.4	1.7
MMD [6]	26.5	-	-	12.7	0.0	2.7	4.0	22.3	30.6	0.0	9.4	0.0	31.6	21.7	52.6	13.9	54.3	8.9	2.5
PCL [18]	22.9	-	-	20.1	0.0	2.2	6.2	28.3	29.0	0.0	9.2	2.6	37.9	22.9	54.5	11.4	45.9	8.5	1.1
PointDR (Ours)	42.5	-	-	16.6	0.0	2.4	3.2	12.2	31.9	0.2	9.0	0.8	42.8	27.1	59.8	18.3	44.0	15.4	5.7

Table 1. Class-wise IoU on domain generalization with SemanticKITTI or SynLiDAR as the source and validation set of *dense fog* in SemanticSTF as the target. '-' represents no samples captured in *dense fog* in the validation set of SemanticSTF.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.
SemanticKITTI→SemanticSTF(*light fog*)																			
Baseline	60.0	0.0	0.0	1.3	10.9	12.3	0.0	0.0	68.6	4.5	36.0	0.0	61.5	53.1	55.6	38.0	44.7	29.2	18.2
Dropout [10]	63.2	0.0	0.0	3.2	10.2	5.5	0.0	0.0	63.8	4.9	29.4	0.1	62.5	53.1	58.6	42.5	46.6	27.8	14.3
Perturbation	76.6	0.0	0.0	38.2	0.0	21.9	0.0	0.0	66.6	8.8	34.6	0.1	62.4	56.1	63.2	25.3	46.2	22.4	6.5
PolarMix [16]	42.6	0.2	0.0	29.4	3.3	17.0	0.0	0.2	69.8	0.7	33.1	0.1	56.2	56.3	54.9	24.7	44.8	24.1	16.6
MMD [6]	63.6	0.0	0.0	0.1	13.3	25.9	0.0	0.0	73.9	5.6	42.8	0.1	64.1	55.3	61.9	36.6	50.7	29.2	9.9
PCL [18]	66.3	0.0	0.0	26.7	0.2	8.7	0.0	0.0	67.8	5.0	36.7	0.4	64.3	58.0	66.1	21.2	53.1	25.5	24.6
PointDR (Ours)	65.9	0.0	0.0	29.7	4.4	11.4	0.9	0.0	70.9	8.8	43.3	0.0	66.5	55.1	61.3	43.0	49.1	29.1	24.3
SynLiDAR→SemanticSTF(*light fog*)																			
Baseline	32.0	4.2	0.5	27.3	0.2	14.0	6.2	0.0	31.0	0.0	12.6	0.9	38.7	24.8	51.5	26.7	46.4	8.5	1.3
Dropout [10]	22.5	3.0	0.9	16.0	0.1	10.0	5.2	0.2	40.3	1.3	18.1	0.0	38.9	22.1	57.6	23.5	38.5	13.8	3.7
Perturbation	31.1	1.9	1.6	21.5	0.0	12.5	2.6	0.0	33.2	1.6	14.3	1.1	34.3	20.1	48.7	29.8	42.0	16.7	4.5
PolarMix [16]	27.3	0.3	0.4	8.9	1.4	8.2	1.2	0.0	29.0	0.2	15.5	0.7	39.9	27.4	57.3	28.8	40.9	5.8	1.5
MMD [6]	31.0	2.1	0.5	16.0	0.0	10.5	1.7	0.0	37.7	0.3	16.3	0.6	29.2	24.9	51.8	29.6	47.8	8.3	1.8
PCL [18]	31.7	0.7	0.8	10.1	0.1	10.2	21.6	0.0	33.9	0.6	16.1	0.1	37.8	22.2	52.5	23.8	42.6	11.3	2.2
PointDR (Ours)	44.7	1.7	1.0	33.9	0.3	12.9	4.7	0.0	36.0	0.9	15.8	0.7	44.4	30.3	60.0	28.3	42.4	15.1	5.7

Table 2. Class-wise IoU on domain generalization with SemanticKITTI or SynLiDAR as the source and validation set of *light fog* in SemanticSTF as the target.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	buid.	fence	veget.	trunk	terra.	pole	traf.
SemanticKITTI→SemanticSTF(*rain*)																			
Baseline	72.4	0.0	-	0.0	16.3	6.9	0.0	-	71.6	12.7	58.1	0.0	70.0	33.0	51.8	9.9	24.2	33.3	22.9
Dropout [10]	81.3	0.0	-	0.0	21.2	5.6	0.0	-	62.2	11.8	44.8	0.6	76.8	44.7	56.0	16.3	23.3	32.8	22.2
Perturbation	83.9	0.0	-	2.4	0.0	20.9	0.0	-	73.2	12.6	54.7	7.0	71.7	43.2	58.3	5.9	29.4	29.4	16.9
PolarMix [16]	56.7	4.0	-	9.1	1.5	29.8	0.0	-	68.2	10.9	50.2	0.5	73.2	47.2	48.3	17.8	22.3	32.3	14.1
MMD [6]	83.9	0.0	-	0.0	8.9	31.6	0.0	-	77.9	17.9	60.2	0.3	69.6	39.3	58.4	14.1	32.5	34.0	30.0
PCL [18]	84.2	0.0	-	0.0	0.1	4.3	0.0	-	68.1	10.9	55.5	4.6	74.7	43.9	59.6	5.8	27.3	34.2	38.8
PointDR (Ours)	78.0	0.0	-	0.0	13.8	20.0	0.0	-	72.1	14.7	60.0	1.2	76.1	36.9	58.0	18.3	24.7	36.1	32.5
SynLiDAR→SemanticSTF(*rain*)																			
Baseline	45.8	4.5	-	6.8	0.4	38.9	0.0	-	32.0	0.0	24.3	0.0	43.0	8.0	33.8	11.3	23.9	11.5	7.7
Dropout [10]	47.0	7.6	-	7.7	0.0	34.0	0.0	-	47.3	6.9	34.6	0.0	39.8	11.5	37.5	13.8	29.6	21.6	8.6
Perturbation	57.5	5.3	-	18.2	0.0	36.3	0.1	-	37.1	1.5	26.9	0.3	34.9	10.4	32.6	12.2	20.5	23.2	10.4
PolarMix [16]	59.6	1.5	-	6.0	5.2	24.6	1.0	-	31.4	0.1	30.4	0.0	55.5	12.2	44.6	13.1	25.0	11.0	4.7
MMD [6]	49.5	4.8	-	20.0	4.7	37.6	0.0	-	43.7	0.0	32.4	0.0	42.1	11.3	34.4	12.3	25.1	13.4	8.1
PCL [18]	51.3	0.9	-	4.3	2.1	35.6	0.0	-	41.4	0.0	32.0	0.0	54.8	9.7	37.1	11.4	24.2	16.6	6.3
PointDR (Ours)	42.2	3.3	-	21.9	0.0	30.4	1.7	-	35.8	3.2	31.9	0.0	54.0	14.4	40.7	12.5	31.9	23.6	11.8

Table 3. Class-wise IoU on domain generalization with SemanticKITTI or SynLiDAR as the source and validation set of *rain* in SemanticSTF as the target. '-' represents no samples captured in *rain* in the validation set of SemanticSTF.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.
SemanticKITTI→SemanticSTF(*snow*)																			
Baseline	49.5	0.0	0.3	0.5	11.6	10.8	-	-	42.1	14.9	23.9	0.0	71.5	26.7	29.3	24.0	17.8	30.8	10.1
Dropout [10]	58.5	0.0	30.5	5.4	13.2	5.2	-	-	41.9	18.0	20.4	2.5	76.4	30.5	31.8	32.7	19.8	28.2	7.0
Perturbation	73.6	0.0	0.0	5.5	1.1	19.8	-	-	45.7	10.9	34.4	0.1	80.6	32.8	45.2	12.8	20.0	24.4	9.5
PolarMix [16]	66.5	3.4	9.3	3.5	5.8	32.4	-	-	55.3	3.6	30.1	0.1	77.8	36.1	34.2	12.6	25.1	29.8	10.1
MMD [6]	59.4	0.0	4.7	0.0	14.7	30.5	-	-	50.8	16.9	32.8	0.2	68.4	24.4	36.6	24.1	24.1	30.0	11.4
PCL [18]	64.0	0.0	0.0	8.2	0.7	9.2	-	-	38.9	15.2	31.6	2.3	79.6	35.1	41.3	11.2	23.1	30.1	26.8
PointDR (Ours)	66.2	0.0	10.4	0.0	16.7	21.3	-	-	43.0	15.2	33.0	1.7	76.8	30.3	36.1	27.6	22.2	30.0	14.1
SynLiDAR→SemanticSTF(*snow*)																			
Baseline	24.6	2.7	1.5	2.4	0.0	32.2	-	-	12.9	0.4	18.3	0.0	33.3	13.8	15.7	14.9	18.1	10.1	1.9
Dropout [10]	35.9	2.8	3.7	3.0	0.0	21.9	-	-	20.9	10.0	22.8	0.0	33.2	14.8	17.1	16.8	16.5	15.7	2.6
Perturbation	27.1	2.4	6.8	6.8	0.2	31.0	-	-	15.4	4.8	19.7	0.0	26.3	12.4	14.0	22.0	16.4	19.0	4.1
PolarMix [16]	53.4	2.3	4.1	6.0	1.2	27.9	-	-	11.7	1.9	21.5	0.3	45.2	20.8	21.7	18.8	16.5	10.5	1.7
MMD [6]	20.8	2.7	6.0	4.8	0.2	31.3	-	-	20.1	0.5	21.0	0.1	29.6	12.2	15.0	16.6	21.8	11.3	2.4
PCL [18]	30.7	1.1	4.4	6.2	0.3	34.6	-	-	19.1	1.7	22.0	0.3	37.8	12.6	16.4	14.2	19.9	14.7	3.0
PointDR (Ours)	34.2	4.0	7.4	7.5	0.1	36.2	-	-	13.8	12.0	22.7	0.0	48.8	19.9	19.9	18.9	17.0	20.7	3.4

Table 4. Class-wise IoU on domain generalization with SemanticKITTI or SynLiDAR as the source and validation set of *snow* in SemanticSTF as the target. '-' represents no samples captured in *snow* in the validation set of SemanticSTF.

A.3. Ablation study

Data augmentation. We study how data augmentation techniques affect generalized semantic segmentation of point clouds (3DSS) and compare them with the proposed PointDR. As Table 5 shows, we report seven models over the benchmark "SemanticKITTI→SemanticSTF": 1) The *Baseline* is a source-only model that is trained by using the training data of SemanticKITTI; 2) The *drop-out, noise perturbation, flipping*, and *jittering* are segmentation models with different augmentation techniques over input data; *All* is the model that combines of all these augmentation techniques; 3) Our proposed *PointDR*. We can see that implementing each of these augmentation techniques improves the generalization capability of the segmentation model clearly and consistently. However, the combination of them all did not yield the best segmentation performance, largely because the combination brings too many distortions to the input point clouds. On the contrary, the proposed PointDR achieves the best segmentation performance, indicating its superior ability to learn universal representations for all-weather 3DSS.

Method	Baseline	drop-out	perturbation	flipping	jittering	All	PointDR
mIoU	24.4	25.7	25.9	25.2	26.9	26.1	28.6

Table 5. Comparison of data augmentation techniques and the proposed PointDR. PointDR performs clearly the best over domain generalized segmentation task SemanticKITTI→SemanticSTF.

Parameter analysis. We examine the parameter λ_{cl} in Eq. 2 in the paper that balances the cross entropy loss and the contrastive loss. As Table 6 shows, optimizing the proposed contrastive loss is able to improve segmentation performance consistently while different λ_{cl} produce quite different mIoUs. The best mIoU is obtained when $\lambda_{cl} = 0.10$.

λ_{cl}	0.0	0.05	0.10	0.15	1.0
mIoU	24.4	28.2	28.6	27.3	25.1

Table 6. Performance of PointDR models with different contrastive loss weight λ_{cl} in Eq. 2 in the paper.

Table 7 below shows segmentation performance with different momentum values (m) used for updating the memory bank $\mathcal{B}$. It performs reasonably well when m is 0.98 or 0.99, showing that a slowly progressing memory bank is beneficial. However, when m is too large (at 0.999), the memory bank updates too slowly to capture the latest and representative feature embeddings, which fails to serve as the class-wise proxy and ultimately leads to a clear segmentation performance drop.

m	0.98	0.99	0.999
mIoU	28.1	28.6	26.1

Table 7. Performance of PointDR models with different momentum updated weight m for the memory bank $\mathcal{B}$.

B. Domain adaptation

B.1. Implementation details

In Tables 4 and 5 of the submitted paper, we examine state-of-the-art UDA methods over the proposed *normal-to-adverse* UDA scenario. Specifically, we selected typical UDA methods from the popular *synthetic-to-real* UDA benchmark [9, 17] as the baseline methods as described in Section 5.2 of the paper. We adopt MinkowskiNet [2] as the segmentation model as in *synthetic-to-real* UDA. When implementing ADDA [13], entropy minimization [14], and self-training [20], we follow the same implementation and training configurations as the *synthetic-to-real* UDA [17] and leverage TorchSparse library [11]] (with version 1.1.0) based on PyTorch [7] library. While for CoSMix [9], we use the officially released codes based on MinkowskiEngine with default training parameters for the adaptation. We report mIoU of the covered classes for individual adverse weather conditions in Table 5.

B.2. Detailed class-level results

In Tables 8, 9, 10, and 11 below, we present the class-level IoU performance for the UDA methods that are examined in the setting of adaptation to individual conditions in Table 5 of the paper.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.
Source-only	56.4	-	-	10.1	0.0	0.6	15.4	0.0	68.0	0.6	22.8	0.0	63.6	36.6	62.8	29.4	53.5	17.7	19.5
ADDA [13]	63.4	-	-	14.3	0.0	2.1	8.0	38.7	68.0	0.1	25.6	0.0	60.6	45.4	64.8	30.4	52.6	20.4	41.9
Ent-Min [14]	68.0	-	-	4.9	0.0	1.9	7.6	0.0	74.8	0.0	39.4	0.0	68.8	50.5	61.0	28.3	63.3	22.7	43.2
Self-training [20]	68.2	-	-	24.4	0.0	5.4	4.8	0.0	70.9	0.3	31.3	0.0	65.9	46.7	59.2	31.6	55.4	22.5	43.7
CoSMix [9]	76.5	-	-	27.0	0.0	4.7	0.0	0.0	74.2	0.5	29.9	1.8	62.1	48.0	62.6	37.3	59.6	23.4	28.8

Table 8. Comparison of state-of-the-art domain adaptation methods on SemanticKITTI→SemanticSTF adaptation for *dense fog*. '-' represents no samples captured in *dense fog* in the validation set of SemanticSTF.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.
Source-only	55.1	0.0	0.0	16.3	4.3	0.7	0.8	0.0	68.3	5.3	33.2	0.0	66.0	44.1	62.0	40.3	48.2	23.4	10.3
ADDA [13]	61.4	0.0	0.0	40.0	10.8	1.4	2.3	0.0	69.4	2.5	36.3	0.0	62.0	52.0	60.4	43.2	48.9	22.7	16.9
Ent-Min [14]	67.1	0.0	0.0	46.7	12.0	0.0	0.0	0.0	73.4	0.2	38.8	0.0	67.1	56.6	56.7	38.2	46.8	25.0	15.7
Self-training [20]	69.3	0.0	0.0	47.5	19.4	0.9	0.1	0.0	73.2	0.8	40.7	0.0	67.4	56.5	58.5	41.3	47.1	26.6	19.8
CoSMix [9]	74.9	0.4	1.3	19.3	1.6	26.1	0.0	0.0	70.3	10.0	35.0	1.1	67.1	54.7	64.1	46.4	49.4	25.4	28.7

Table 9. Comparison of state-of-the-art domain adaptation methods on SemanticKITTI→SemanticSTF adaptation for *light fog*.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.
Source-only	69.4	0.0	-	0.1	0.1	12.1	0.0	-	72.9	9.7	54.5	0.0	73.7	31.2	55.2	16.2	21.4	33.9	18.8
ADDA [13]	71.8	0.0	-	0.1	0.7	3.8	0.0	-	71.9	9.2	51.5	0.0	67.8	35.6	53.6	17.8	25.7	32.0	24.2
Ent-Min [14]	78.4	0.0	-	0.4	2.9	0.1	0.0	-	80.3	10.1	57.9	0.0	78.0	47.1	53.8	13.0	24.1	35.8	33.8
Self-training [20]	69.4	0.0	-	0.1	0.1	12.1	0.0	-	72.9	9.7	54.5	0.0	73.7	31.2	55.2	16.2	21.4	33.9	18.8
CoSMix [9]	83.6	0.1	-	2.1	11.8	47.9	0.0	-	64.7	10.9	51.1	2.5	72.6	47.2	59.8	25.7	20.9	27.2	35.2

Table 10. Comparison of state-of-the-art domain adaptation methods on SemanticKITTI→SemanticSTF adaptation for *rain*. '-' represents no samples captured in *rain* in the validation set of SemanticSTF.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.
Source-only	70.7	0.0	0.0	15.4	1.6	5.1	-	-	49.8	8.9	36.6	0.0	67.1	26.3	30.7	28.1	22.1	26.8	9.6
ADDA [13]	69.3	0.0	0.0	14.0	0.8	2.9	-	-	55.3	1.3	35.7	0.0	67.2	26.7	37.5	30.1	21.2	25.4	11.0
Ent-Min [14]	73.8	0.0	15.4	19.8	1.4	2.9	-	-	53.6	1.6	32.9	0.0	73.4	28.5	34.1	28.8	21.7	26.6	8.8
Self-training [20]	73.9	0.0	6.1	16.9	5.2	7.7	-	-	53.9	6.2	34.3	0.0	69.3	27.7	33.7	29.8	19.5	26.9	16.0
CoSMix [9]	79.2	1.3	0.0	0.6	14.2	38.9	-	-	70.1	15.1	54.1	6.3	74.6	44.1	58.3	20.5	20.4	26.9	35.6

Table 11. Comparison of state-of-the-art domain adaptation methods on SemanticKITTI→SemanticSTF adaptation for *snow*. '-' represents no samples captured in *snow* in the validation set of SemanticSTF.

C. Supervised learning on adverse conditions

We use SemanticSTF to train five state-of-the-art 3DSS models in a supervised manner and report their segmentation performance in Table 12. Specifically, we use their officially released codes and default training configurations for model training. We can see that these state-of-the-art models achieve much lower segmentation performance over SemanticSTF as compared with their performance over SemanticKITTI. The results indicate that SemanticSTF is a more challenging benchmark for supervised methods due to the diverse data distribution and hard geometric domains. In addition, comparing Table 12 and Table 6 of the submitted paper, we notice that the rankings of the supervised and the pre-trained 3DSS models are not well aligned, indicating that the ability of supervised representation learning may not be highly correlated with the generalization ability. We also notice that the state-of-the-art network Cylinder3D [19] achieves much lower segmentation performance over SemanticSTF as compared with its performance over SemanticKITTI . This could be due to two major factors: 1) The design of Cylinder3D is sensitive to complicated and noisy geometries of point clouds as introduced by various adverse weather conditions; 2) Cylinder3D is sensitive to training parameters and the default training configurations for SemanticKITTI does not work well for SemanticSTF. The results further demonstrate the importance of studying universal 3DSS as well as the value of the proposed SemanticSTF dataset in steering the future endeavour along this very meaningful research direction.

Methods	car	bi.cle	mt.cle	truck	oth-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	oth-g.	build.	fence	veget.	trunk	terra.	pole	traf.	invalid	mIoU
RandLA-Net [5]	75.2	0.0	0.0	25.8	0.0	47.3	0.0	0.0	73.3	7.8	48.7	57.5	68.2	48.3	61.5	27.3	49.5	39.7	27.5	56.5	35.7
SalsaNext [3]	77.3	31.2	0.0	47.5	30.5	64.2	26.6	5.0	76.3	18.2	55.2	64.9	79.2	50.4	56.8	27.8	55.8	36.8	36.7	62.2	45.1
MinkowskiNet [2]	87.4	42.5	0.0	51.2	40.3	73.6	29.1	0.0	79.5	15.0	57.7	63.4	78.6	56.8	64.4	40.4	53.3	47.6	47.6	67.7	49.8
SPVCNN [12]	87.1	45.5	0.0	53.1	42.7	74.1	21.9	0.0	78.9	16.3	57.9	57.0	78.6	56.5	65.6	40.9	50.3	49.4	45.9	66.4	49.4
Cylinder3D [19]	77.7	31.7	2.7	43.4	23.8	67.8	18.4	0.0	78.5	10.0	51.8	48.7	81.2	56.0	63.4	38.3	52.1	48.0	43.0	63.9	45.0

Table 12. Comparison of state-of-the-art 3DSS methods (trained in a supervised manner) over the test set of SemanticSTF.

D. Additional Details on SemanticSTF Dataset

D.1. Annotation

In this section, we explain the implementation of our point cloud labeling in more detail. We leveraged a professional labeling program that has multiple annotating tools such as a brush, a polygon tool, a bounding volume tool, as well as different filtering methods for hiding labeled points or selected labels. Corresponding 2D images are displayed to assist labelling. The program also supports cross-checking and correction as illustrated in the main paper. Fig. 1 shows the interface of our point cloud annotation program.

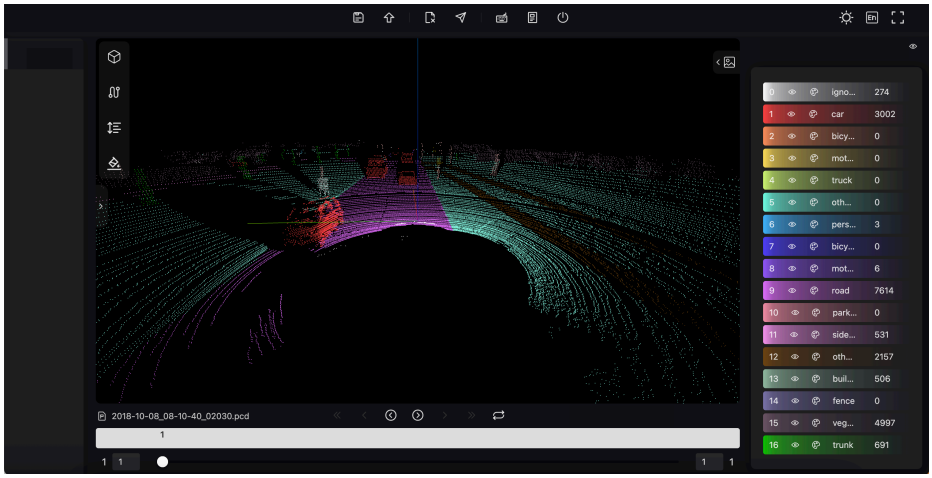

Figure 1. The interface of point cloud labeling program for annotating SemanticSTF.

D.2. Semantic class definition

In the process of labeling such challenging data, we had to decide which classes we wanted to annotate at some point in time. In general, we followed the class definitions of the SemanticKITTI dataset [1] and ACDC [8] dataset, but did some simplifications and adjustments for the data source used. The annotated classes with their respective definitions are listed in Table 13 below.

cat.	class	definition
flat	road	Drivable areas where cars could drive on including main road, bike lanes, and crossed areas on the street. Road curb is excluded.
	sidewalk	Paths along sides of the road, used for pedestrians and bicycles, but cars are not allowed to drive on. Also include private driveways.
	parking	Areas for parking and are clearly different from sidewalk and road. If unclear then other-ground or sidewalk can be selected. Garages are labeled as building instead of parking.
	other-ground	Ground that excludes sidewalk, terrain, road, and parking. It includes (paved/plastered) traffic islands that are not meant for walking.
construction	building	All building parts including walls, doors, windows, stairs, and garages, etc.
	fence	Separators including wood or metal fences, small walls and crash barriers.
vehicle	car	Different types of cars, including cars, jeeps, SUVs, and vans.
	truck	Trucks, vans with a body that is separate from the driver cabin, pickup trucks, as well as their attached trailers.
	bicycle	Including different types of bicycles, without any riders or pedestrians nearby.
	motorcycle	Including different types of motorcycles, without any riders or pedestrians nearby.
	other-vehicle	Other types of vehicles that do not belong to previously defined vehicle classes, such as various trailers, excavators, forklifts, and fallbacks.
nature	vegetation	Including bushes, shrubs, foliage, treetop except for trunks, and other clearly identifiable vegetation.
	trunk	The tree trunk is labeled as *trunk* separately from the treetop.
	terrain	Mainly include grass and soil.
human	person	Humans that are standing, walking, sitting, or in any other pose, but not driving any vehicle. Trolley cases, strollers, and pets nearby are excluded.
	bicyclist	Humans driving a bicycle or standing in close range to a bicycle (within *arm reach*).
	motorcyclist	Humans driving a motorcycle or standing in close range to a motorcycle (within *arm reach*).
object	pole	Lamp posts, the poles of traffic signs and traffic lights.
	traffic sign	Traffic signs excluding their mounting.
	invalid	Indiscernible semantic contents caused by adverse weather, such as points of thick snow cover, falling snow or rain droplets, and the splash from the rear of the moving vehicles when driving on the road of snow or water.

Table 13. Definitions of semantic classes in SemanticSTF.

References

[1] 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. 6

[2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019. 1, 4, 5

[3] 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. Springer, 2020. 5

[4] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 1

[5] 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. 5

[6] 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

[7] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 4

[8] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10765–10775, 2021. 6

[9] Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Elisa Ricci, and Fabio Poiesi. Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. *ECCV*, 2022. 4

[10] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 2, 3

[11] Haotian Tang, Zhijian Liu, Xiuyu Li, Yujun Lin, and Song Han. TorchSparse: Efficient Point Cloud Inference Engine. In *Conference on Machine Learning and Systems (MLSys)*, 2022. 1, 4

[12] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *European conference on computer vision*, pages 685–702. Springer, 2020. 1, 5

[13] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017. 4

[14] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. 4

[15] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 1

[16] Aoran Xiao, Jiaxing Huang, Dayan Guan, Kaiwen Cui, Shijian Lu, and Ling Shao. Polarmix: A general data augmentation technique for lidar point clouds. *NeurIPS*, 2022. 1, 2, 3

[17] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from synthetic to real lidar point cloud for semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2795–2803, 2022. 4

[18] 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

[19] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9939–9948, 2021. 5

[20] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019. 4