Test-Time Training for LiDAR Semantic Segmentation under Corruption via Geometric Inlier Discrimination

Supplementary Material

1. Overview

This supplementary material provides additional details and results that complement the main paper. It consists of the following sections:

- Implementation details (Sec. 2).
- Evaluation beyond corruption datasets (Sec. 3).
- Correlation between GeoID and segmentation (Sec. 4).
- Sensitivity analysis (Sec. 5).
- Continual test-time training details (Sec. 6).
- Performance breakdown by corruption type (Sec. 7).
- Additional qualitative results (Sec. 8).

2. Implementation Details

We use two LiDAR semantic segmentation backbones in our experiments: MinkUNet34 [1] and Cylinder3D [4]. All baselines and our method share the same backbone architecture for fair comparison. For both backbones, the GeoID decoder adopts the same architecture as the segmentation decoder, except for the final output layer. We use the Adam optimizer [2] for all training and adaptation.

MinkUNet34. For MinkUNet34, we use a voxel size of 0.2 m on both source datasets. During joint training on the source domain, we use a learning rate of 0.01 with a batch size of 4. During test-time training, we use a learning rate of 0.001 with a mini-batch size of $B = 2$ (*i.e.* two synthetic GeoID augmentations per test sample).

Cylinder3D. For Cylinder3D, we follow the official training setting. Joint training uses a learning rate of 0.001 with a batch size of 2, while test-time training uses a learning rate of 0.0003 with a mini-batch size of $B = 1$.

Data augmentation and GeoID settings. During joint training, both the original point cloud and the augmented point cloud with synthetic off-manifold points are fed into the model. We apply standard data augmentations: random rotation around the z-axis, random translation, scaling, and random flipping along the x, y, or x–y plane. At test time, we use the same off-manifold construction as in joint training, with the maximum spatial displacement set to $\delta = 0.6$ m (as in Eq. (2) in the main paper). For bidirectional unreliable point filtering, the reliability threshold τ_r is set to 0.4 on SemanticKITTI-C and 0.2 on nuScenes-C.

Computational cost. Under the default setting using the MinkUNet34 backbone on SemanticKITTI-C, GeoID requires 24 GFLOPs and 2.8GB of peak memory on a single NVIDIA A6000 GPU. A runtime analysis is provided in Fig. 5 of the main paper, showing that increasing the stride improves FPS with only a modest performance trade-off.

Table 1. Test-time training results on clean target datasets under both in-dataset and cross-dataset evaluation settings. For brevity, SemanticKITTI and nuScenes are denoted by K and N.

Method	K→K	K→N	N→N	N→K
Source-only	54.06	35.32	71.35	28.30
GeoID	55.51	38.60	71.76	36.97

3. Evaluation Beyond Corruption Datasets

To further validate GeoID beyond corruption benchmarks, we additionally evaluate test-time adaptation on clean target datasets under both in-dataset (SemanticKITTI → SemanticKITTI, nuScenes → nuScenes) and cross-dataset (SemanticKITTI → nuScenes, nuScenes → SemanticKITTI) settings, as summarized in Tab. 1. Overall, GeoID consistently improves performance across all settings, showing that it remains effective even in non-corrupted scenarios without introducing negative transfer. These results suggest that the geometric cues used for inlier discrimination, such as planar roads and vertical facades, are not limited to corruption-specific cases but remain informative across domains and under broader distribution shifts.

4. Correlation Between GeoID and Semantic Segmentation

We further examine whether GeoID provides a task-relevant self-supervised signal for semantic segmentation during test-time adaptation by analyzing the correlation between the two tasks. Although GeoID does not explicitly constrain the semantic decision boundary, it may still support the target task if its optimization is positively correlated with the semantic segmentation objective. To verify this, we evaluate intermediate test-time adaptation checkpoints and track both the GeoID and semantic segmentation losses. As shown in Fig. 1, the two losses decrease together throughout adaptation, indicating a positive correlation between the GeoID objective and the semantic segmentation objective. This observation suggests that GeoID provides an implicit yet task-relevant test-time training signal.

5. Sensitivity Analysis

We analyze the sensitivity of GeoID to key design choices used in test-time adaptation, including the reliability thresh-

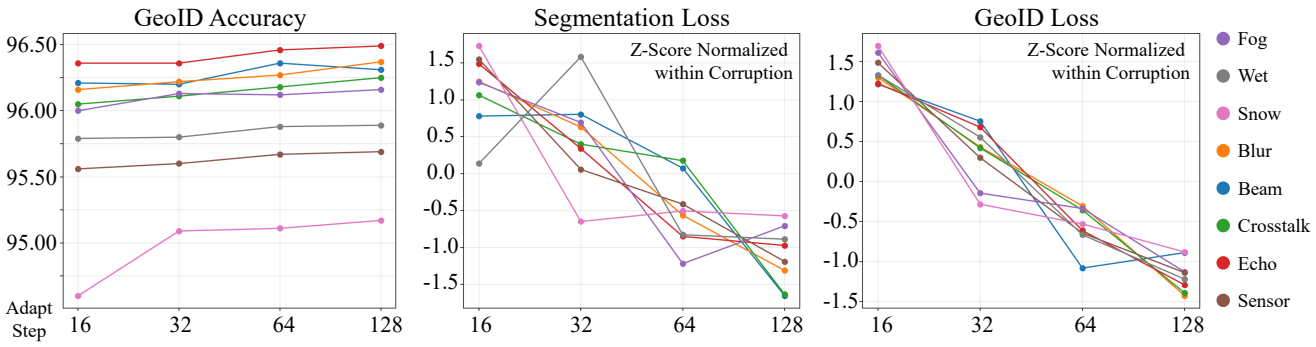

Figure 1. Correlation between GeoID and semantic segmentation during test-time adaptation. As adaptation progresses, GeoID accuracy increases (left), while both the semantic segmentation loss (middle) and the GeoID loss (right) decrease, suggesting a positive correlation between the two tasks. The loss is Z-score normalized within each corruption for clearer visualization.

Table 2. Sensitivity analysis of the reliability threshold τ_r on SemanticKITTI-C.

τ_r	Fog	Wet Ground	Snow	Motion Blur	Beam Missing	Crosstalk	Incomplete Echo	Cross Sensor	Light	Moderate	Heavy	Avg.
0.2	40.40	47.42	40.33	54.62	51.24	43.10	50.28	47.90	49.76	47.60	43.38	46.91
0.4	40.14	47.66	40.63	54.59	52.10	40.82	50.79	48.97	50.01	47.66	43.22	46.96
0.6	39.24	47.42	40.26	54.60	52.27	36.75	51.38	48.83	49.77	46.65	42.60	46.34

Table 3. Sensitivity analysis of the reliability threshold τ_r on nuScenes-C.

τ_r	Fog	Wet Ground	Snow	Motion Blur	Beam Missing	Crosstalk	Incomplete Echo	Cross Sensor	Light	Moderate	Heavy	Avg.
0.2	57.40	69.27	53.91	69.12	63.49	30.84	59.72	50.11	63.21	56.81	50.19	56.73
0.4	56.60	69.27	51.70	69.08	63.67	26.27	59.77	49.64	62.10	55.97	49.27	55.78
0.6	55.23	68.43	48.86	69.25	63.70	24.95	59.80	48.41	61.47	55.19	48.19	54.95

Table 4. Sensitivity analysis of the maximum displacement magnitude δ on SemanticKITTI-C.

δ (m)	Fog	Wet Ground	Snow	Motion Blur	Beam Missing	Crosstalk	Incomplete Echo	Cross Sensor	Light	Moderate	Heavy	Avg.
0.2	34.22	37.30	53.61	54.67	50.63	36.45	50.58	48.76	48.38	45.42	40.91	44.90
0.6	40.14	47.66	40.63	54.59	52.10	40.82	50.79	48.97	50.01	47.66	43.22	46.96
1.0	42.38	49.50	40.74	53.30	51.25	38.79	50.67	49.61	48.87	46.15	46.08	47.03

Table 5. Effect of the perturbation direction on SemanticKITTI-C.

Direction	Fog	Wet Ground	Snow	Motion Blur	Beam Missing	Crosstalk	Incomplete Echo	Cross Sensor	Light	Moderate	Heavy	Avg.
Axis	40.14	47.66	40.63	54.59	52.10	40.82	50.79	48.97	50.01	47.66	43.22	46.96
Non-axis	39.08	49.96	41.51	54.16	53.13	37.48	51.40	51.43	50.02	47.83	43.40	47.27

old τ_r, the maximum displacement magnitude δ, and the perturbation direction.

Reliability threshold. Tables 2 and 3 show the sensitivity of GeoID to the reliability threshold τ_r. Overall, the performance remains stable across a broad range of threshold values. Although the optimal τ_r varies slightly depending on the corruption type, the best performance is typically obtained when τ_r is in the range of 0.2 to 0.4. We also observe that the optimal threshold is largely consistent across different severity levels. On average, the optimal values are 0.4 on SemanticKITTI-C and 0.2 on nuScenes-C.

Maximum displacement magnitude. Table 4 reports the sensitivity of GeoID to the maximum displacement magnitude δ. GeoID remains robust across different choices of δ and consistently outperforms the baselines reported in Tab. 1 of the main paper. In particular, better performance is obtained when the displacement is sufficiently large. We attribute the degradation at smaller displacements to the fact

Table 6. Corruption sequences used for continual test-time training on SemanticKITTI-C. "Seq." denotes the position in the sequence, and "Comb. 1-3" denote three continual test streams constructed by combining eight different LiDAR corruptions.

Seq.	Comb. 1	Comb. 2	Comb. 3
1	Crosstalk	Motion Blur	Beam Missing
2	Fog	Cross Sensor	Crosstalk
3	Motion Blur	Beam Missing	Cross Sensor
4	Wet Ground	Wet Ground	Incomplete Echo
5	Beam Missing	Snow	Snow
6	Cross Sensor	Incomplete Echo	Motion Blur
7	Snow	Crosstalk	Fog
8	Incomplete Echo	Fog	Wet Ground

that the injected points become more tightly entangled with inlier structures, making inlier discrimination more challenging.

Perturbation direction. Table 5 reports results obtained using non-axis-aligned outliers generated by displacements along random 3D directions. GeoID achieves comparable performance under these perturbations, with a slightly higher average performance in our evaluation. This result suggests that the behavior of GeoID is not specific to the axis-aligned injection pattern.

6. Continual Test-Time Training Details

In the main paper, we evaluate our method in a continual test-time training scenario where different corruption types appear sequentially in a single test stream. We partition the stream into eight equal segments and randomly assign to each segment one of eight LiDAR corruption types. This process is repeated three times to define three continual test streams, as summarized in Tab. 6.

7. Performance Breakdown by Corruption

Table 7 reports the per-corruption mIoU of Cylinder3D-based methods on SemanticKITTI-C. GeoID achieves the highest mean performance and shows clear gains on several corruptions (e.g., fog, snow, beam missing, and crosstalk), while NormFusion [3] performs better on others. Overall, these results indicate that our geometry-aware test-time adaptation is particularly effective for certain corruption types and competitive with state-of-the-art normalization-based adaptation on this benchmark.

8. Additional Qualitative Results

Additional qualitative results on SemanticKITTI-C and nuScenes-C are shown in Figs. 2 and 3, respectively. Compared to the baselines, our method better preserves object boundaries and maintains semantic consistency under various corruptions.

References

[1] 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

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[3] Yunxiang Zhou, Xin Lou, Ying He, Wei Zhang, and Indriyati Atmosukarto. Normfusion: Real-time test-time adaptation for lidar segmentation under corruption. In *2025 IEEE Conference on Artificial Intelligence (CAI)*, pages 1029–1035. IEEE, 2025. 3

[4] 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. 1, 4

Table 7. Comparison results against test-time adaptation methods using Cylinder3D [4] segmentation model, reporting mIoU (%) for each corruption type in SemanticKITTI-C. The best result is highlighted in **bold**.

Method	Fog	Wet Ground	Snow	Motion Blur	Beam Missing	Crosstalk	Incomplete Echo	Cross Sensor	Mean
Source-only	43.51	56.36	47.61	52.84	57.89	54.38	52.67	54.52	52.43
MixNorm	44.15	58.84	48.73	56.99	58.13	55.16	53.54	55.13	53.83
TENT	42.11	56.08	47.25	58.61	57.32	54.09	52.49	54.98	52.86
CoTTA	44.65	58.97	49.08	57.07	58.39	55.51	53.96	55.29	54.11
NormFusion	45.09	**60.06**	50.11	**60.18**	58.67	56.03	**54.92**	**56.94**	55.25
GeoID (Ours)	**51.73**	59.87	**54.32**	56.76	**59.37**	**63.12**	50.80	47.71	**55.46**

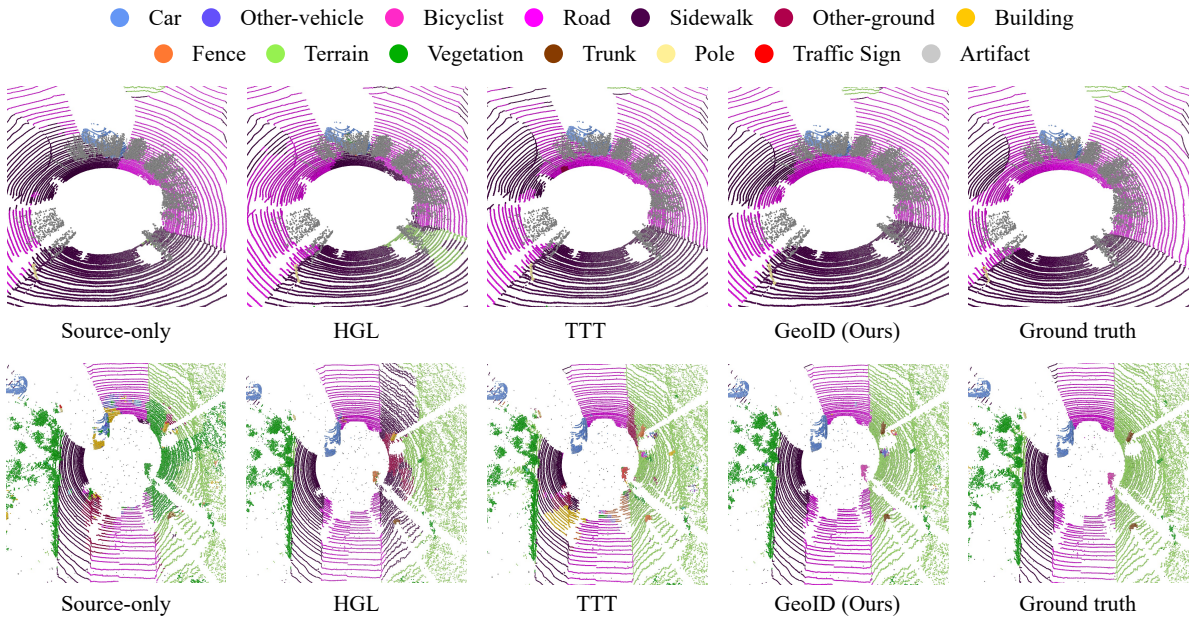

Figure 2. Qualitative results on SemanticKITTI-C under *fog* (top) and *crosstalk* (bottom) corruptions.

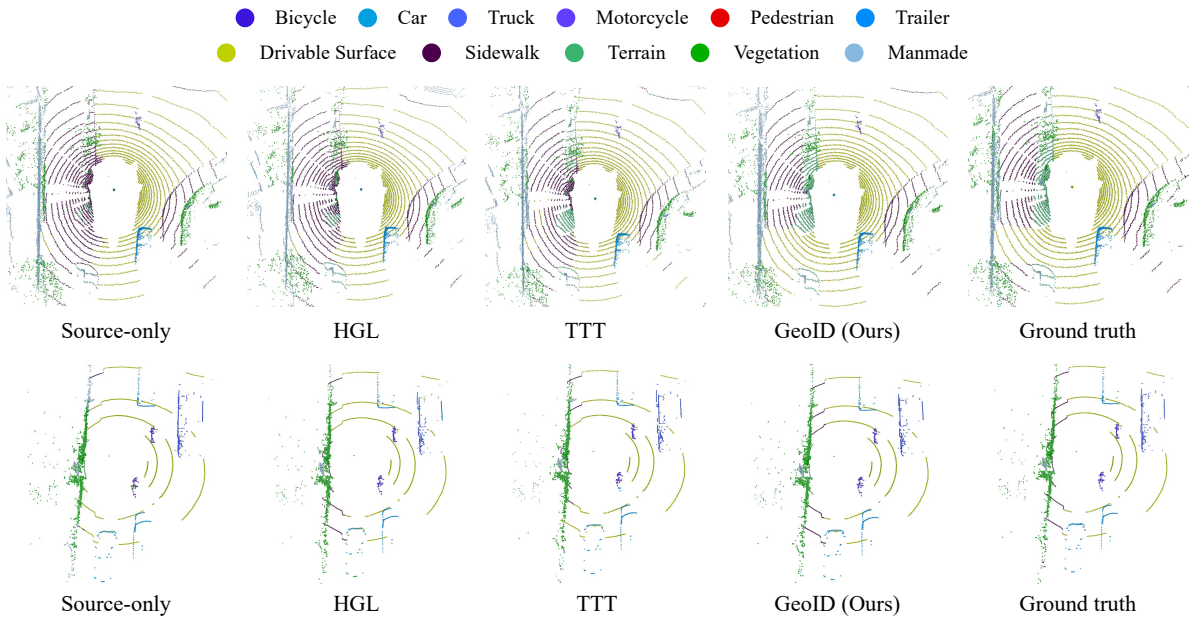

Figure 3. Qualitative results on nuScenes-C under *motion blur* (top) and *cross sensor* (bottom) corruptions.