

000 SCENE, CLASS, SIGNAL: TRI-LEVEL ADAPTATION 001 FOR SYNTHETIC-TO-REAL LiDAR SEGMENTATION 002

003 **Anonymous authors**
004

005 Paper under double-blind review
006

007 ABSTRACT

008 Synthetic LiDAR datasets offer a scalable alternative to costly real-world an-
009 notations, but still exhibit a significant domain gap when applied to real-world
010 data. Previous unsupervised domain adaptation (UDA) methods mainly rely on
011 general adaptation strategies, without directly addressing the LiDAR-specific fac-
012 tors causing this gap. In this work, we analyze the synthetic-to-real domain gap
013 from a root-cause-driven perspective. We decompose the components of this gap
014 into three distinct granularities: **scene-level**, **class-level**, and **signal-level**. At the
015 **scene-level**, we address the point structure distortions caused by real-world sensor
016 effects, such as motion blur and rolling shutter. At the **class-level**, we consider that
017 the domain gap varies depending on the structural complexity and dynamicity of
018 each object class. Finally, at the **signal-level**, we tackle the lack of direct, realistic
019 semantic information that corresponds to the synthetic input. To address these
020 challenges, we propose a multi-level adaptation framework. Motivated by our
021 finding that encoder feature statistics (mean and variance) capture point structural
022 domain gap, we propose to employ a *style embedding*. Built from the feature mean
023 and variance, this embedding serves as a domain cue for adversarial learning at
024 the **scene-level** and extends to the **class-level** for category-dependent shifts. At the
025 **signal-level**, we complement this with an *intensity-guided self-training* scheme
026 for handling non-structural gaps, leveraging real LiDAR intensity as weak super-
027 vision for synthetic inputs. On *SynLiDAR*→*SemanticKITTI*, our method achieves
028 44.7 mIoU, and on *SynLiDAR*→*SemanticPOSS*, it reaches 51.2 mIoU, setting a
029 new state of the art on both benchmarks. Extensive ablation studies validate each
030 component, confirming our style embedding captures the structural domain gap
031 while our self-training scheme significantly improves adaptation.

032 1 INTRODUCTION

033 LiDAR semantic segmentation (LSS), which aims to assign a semantic label to each 3D point cap-
034 tured by LiDAR, is a fundamental task for scene understanding in autonomous driving. By providing
035 dense and geometrically accurate representations of the environment, LSS plays a key role in en-
036 abling high-level decision-making processes such as path planning and obstacle avoidance. Despite
037 its importance, the development of high-performing LSS models remains challenging due to the
038 significant annotation cost associated with acquiring large-scale, labeled real-world data.

039 Unlike other data domain, collecting LiDAR data for outdoor driving scenes requires extensive
040 physical driving across diverse environments while recording high-frequency 3D point clouds. This
041 process is inherently time-consuming and must be conducted under real traffic conditions, where
042 safety risks are persistently present due to the unpredictable behavior of surrounding agents. Fur-
043 thermore, the annotation of LiDAR point clouds is labor-intensive, as it often involves manual cross-
044 referencing with RGB images to determine point-wise semantic labels (Behley et al., 2019).
045

046 To alleviate the need for physical data acquisition and human annotation, synthetic LiDAR datasets
047 have gained popularity. These datasets enable scalable supervision through simulation, allowing
048 models to be trained on fully annotated synthetic data and then adapted to real domains. However,
049 bridging the synthetic-to-real domain gap is uniquely difficult for LiDAR, as the discrepancies are
050 not limited to texture or appearance (as in images). In particular, the domain shift in LiDAR stems
051 from physical signal formation and sensor characteristics. This makes LiDAR adaptation a different
052 and more complex challenge compared to other modalities.

053 Specifically, real-world LiDAR data is affected by phenomena such as beam attenuation, scattering,
054 and occlusion (Manivasagam et al., 2023; Hahner et al., 2021; 2022), leading to sparse, noisy, and

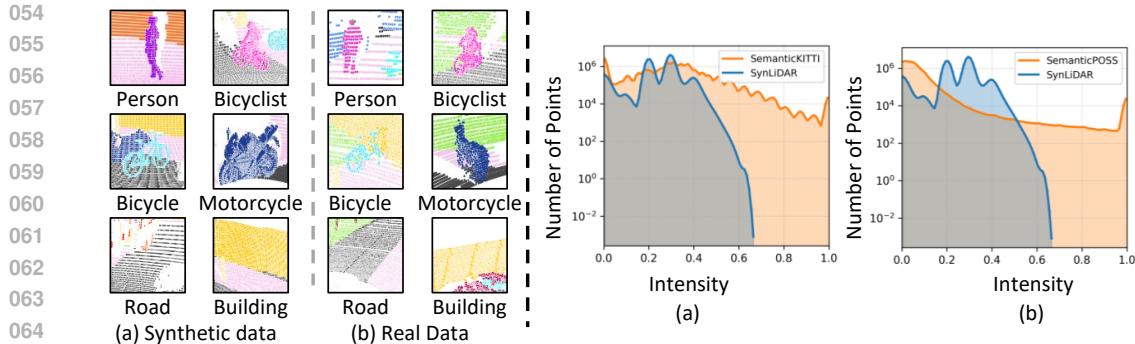


Figure 1: **(Left)** Point structure difference between (a) SynLiDAR and (b) SemanticKITTI. Objects in the first and second row are inherently more dynamic and structurally complex than objects in the third row. As a result, they experience greater point structure distortion, leading to class-dependent variations in the domain gap. **(Right)** Intensity distribution on (a) SynLiDAR & SemanticKITTI and (b) SynLiDAR & SemanticPOSS. A significant difference in intensity distribution can be observed between synthetic and real data.

irregular point distributions. In contrast, synthetic data, typically generated via uniform ray casting, tends to exhibit overly dense and artifact-free geometry. Moreover, real LiDAR sensors introduce temporal distortions such as rolling shutter effects and motion blur (Manivasagam et al., 2020; 2023; Hess et al., 2025), which are rarely modeled in simulation. Intensity distributions also diverge: real beams decay over distance and vary by surface reflectance (Vacek et al., 2021), whereas synthetic intensities are often simplified for efficiency. These observations indicate that the LiDAR domain gap has deeper structural and semantic roots than previously considered.

While recent efforts have proposed various unsupervised domain adaptation (UDA) techniques to bridge this gap, most existing approaches fall short in fully addressing the underlying challenges. Several methods augment inputs with noise (Li et al., 2023), others utilize entropy-based adversarial learning (Yuan et al., 2024), others regress intensity from the target data to create an additional learning signal (Yuan et al., 2023), and a few adopt teacher-student frameworks with self-training. However, these approaches share three key limitations. **(L1)** They rely on indirect domain cues, such as noise and entropy, that reflect consequences of the domain gap rather than its root cause. **(L2)** They apply domain adaptation uniformly across the entire scene, overlooking the fact that domain shifts are often class-dependent. As shown on the left of Fig. 1, the severity of the gap varies across object categories, making global adaptation insufficient. **(L3)** Although the intensity value of the target data implicitly contains the semantics of real data, this information is not utilized for learning from synthetic data inputs.

In this paper, we take a root-cause-driven perspective and propose a *coarse-to-fine LiDAR adaptation framework* that systematically addresses the domain gap along three levels of granularity: **scene-level**, **class-level**, and **signal-level**.

In **scene-level**, inspired by prior findings that feature statistics in encoder layers capture the “style” of the input (Huang & Belongie, 2017) and our findings, we hypothesize that such feature statistics of a LiDAR segmentation model can effectively represent point structure differences—such as variations in point density, motion blur, and rolling shutter artifacts—as a form of style.

In **class-level**, following prior work (Manivasagam et al., 2023), we observed that variations in object motion and geometric structure across semantic categories lead to class-dependent domain shifts (the left of Fig. 1). To address this, we extend the use of feature statistics from a scene-level representation to a class-wise formulation. This design allows the model to more precisely capture class-specific structural differences, thereby improving its ability to mitigate class-dependent domain shifts. Moreover, semantic similarity among classes naturally implies similarity in their complexity and dynamics. Thus, to effectively address class-dependent domain gaps, style representations should be aligned among visually similar classes. To this end, we adopt a class-hierarchical formulation that aligns styles within superclasses (e.g., *bicycle* and *motorcycle*), thereby capturing structural similarities among related classes. This formulation ensures more consistent adaptation and improves model generalization. Note that in row 1 and row 2 on the left of Fig. 1, we visualize examples belonging to the same superclass to illustrate this concept.

108 In **signal-level**, overfitting to synthetic data arises primarily from the absence of supervision in the
 109 real domain in UDA setting. To compensate for the lack of real labels, prior works (Yuan et al., 2023;
 110 Viswanath et al., 2025) leverage intensity measurements from real LiDAR scans as an auxiliary
 111 learning signal. However, given the fact that intensity contains implicit semantic information (Song
 112 et al., 2002; Naich & Carrión, 2024; Viswanath et al., 2025), utilizing this information for synthetic
 113 data supervision could be more effective in reducing the domain gap. We advance prior work in this
 114 direction. We construct a self-training mechanism that encourages the synthetic data input to follow
 115 the real LiDAR intensity distribution, which mitigates the synthetic-to-real gap.

116 We validate these hypotheses through extensive experiments. Our method achieves state-of-the-art
 117 performance, reaching 44.7 mIoU on the SynLiDAR-to-SemanticKITTI benchmark and 51.2 mIoU
 118 on the SynLiDAR-to-SemanticPOSS benchmark. Ablation studies further confirm the effectiveness
 119 of each component, and the method shows robustness to hyperparameter choices.

120 In summary:

121

- 122 • We define the domain gap using feature statistics extracted from the encoder of the seg-
 123 mentation model.
- 124 • We introduce class-wise style embeddings to account for class-dependent point structure
 125 distortions.
- 126 • Through our intensity-guided self-training strategy, we encouraged the synthetic data input
 127 to follow the real intensity distribution.
- 128 • Our one-stage framework achieves state-of-the-art performance on two major synthetic-to-
 129 real benchmarks.

130 2 RELATED WORKS

131 2.1 LiDAR SEMANTIC SEGMENTATION

132 LiDAR semantic segmentation (LSS) methods can be categorized into three groups: point-based,
 133 projection-based, and voxel-based approaches.

134 **Point-based methods** Qi et al. (2017); Thomas et al. (2019); Zhao et al. (2021); Thomas et al.
 135 (2024); Choe et al. (2022) directly process LiDAR points. Despite preserving geometric details,
 136 these methods are computationally intensive and slow at inference, limiting suitability for dynamic
 137 scenes.

138 **Projection-based methods** Milioto et al. (2019); Ando et al. (2023); Kong et al. (2023a); Xu et al.
 139 (2025) convert 3D LiDAR points into 2D representations, typically via spherical projection, allowing
 140 efficient use of 2D CNNs. Milioto et al. (2019) pioneered using spherical-projected range images
 141 for semantic segmentation. Ando et al. (2023) improved performance by fine-tuning a ViT pre-
 142 trained on images. Kong et al. (2023a) enhanced projection-based methods with specialized data
 143 augmentation. Xu et al. (2025) edited spherical projection with an MLP-based approach, preserving
 144 all input point information.

145 **Voxel-based methods** Choy et al. (2019); Zhou et al. (2020); Lai et al. (2023) voxelize 3D points
 146 for computational efficiency and parallel processing. Choy et al. (2019) introduced sparse 3D con-
 147 volution, and Tang et al. (2020) integrated point-wise features to reduce information loss. Zhou
 148 et al. (2020) proposed cylindrical voxelization to address varying point densities. Lai et al. (2023)
 149 employed radial window partitioning to capture long-range context. Voxel-based methods balance
 150 segmentation quality and inference speed effectively, making them popular for LiDAR segmenta-
 151 tion.

152 2.2 SYNTHETIC LiDAR SIMULATIONS

153 To address the difficulty of collecting large-scale driving scene LiDAR data in the real world, ex-
 154 isting studies have attempted to simulate realistic LiDAR data (Manivasagam et al., 2020; 2023;
 155 Xiao et al., 2022; Vacek et al., 2021). Manivasagam et al. (2020) proposed a method for generating
 156 realistic synthetic LiDAR data by pairing each sample with corresponding real LiDAR data. Mani-
 157 vasagam et al. (2023) generated realistic outdoor LiDAR data by transforming spherical-projected
 158 range images to match real data, incorporating effects such as motion blur and rolling shutter. Xiao
 159 et al. (2022) used Unreal Engine to synthesize LiDAR data and applied a generative model to sim-
 160 ulate point drops, producing more realistic patterns. Vacek et al. (2021) focused on the intensity

162 gap between synthetic and real data, and used a neural network to generate realistic intensity dis-
 163 tributions for ray-casted LiDAR points. Despite these efforts, simulating realistic LiDAR remains
 164 computationally expensive and requires modeling all relevant domain gap components. Therefore,
 165 synthetic-to-real unsupervised domain adaptation is essential for practical deployment.
 166

167 **2.3 SYNTHETIC-TO-REAL UNSUPERVISED DOMAIN ADAPTATION FOR LiDAR SEMANTIC
 168 SEGMENTATION**

169 Previous works have addressed domain adaptation by adding noise to the input data (Li et al., 2023),
 170 interpreting entropy as a cue for domain gap and minimizing it through adversarial learning, or en-
 171 hancing teacher-student architectures via self-training mechanisms (Yuan et al., 2024). Additionally,
 172 other studies have sought to learn implicit semantic information by training on the intensity values of
 173 the target dataset (Yuan et al., 2023). Rather than relying on noise or entropy as the primary source of
 174 the domain gap, we extract style embeddings from the segmentation model to directly characterize
 175 structural discrepancy and reduce the domain gap. Furthermore, unlike prior approaches, we incor-
 176 porate point intensity within a self-training strategy to be aware of implicit semantic information of
 177 the real domain from synthetic input and to prevent overfitting to the synthetic domain.

178 **3 MOTIVATION**

179 As discussed in Sec. 1, we take a root-cause-driven perspective and propose a novel LiDAR adap-
 180 tation framework that systematically addresses the domain gap along three levels of granularity:
 181 **scene-level**, **class-level**, and **signal-level**. We must address three key challenges to achieve this:
 182 **(scene-level)** discovering domain cues that explicitly capture point structure differences, the main
 183 cause of the synthetic-to-real domain gap. **(class-level)** modeling these domain cues from the scene
 184 level to the class level to capture class-dependent domain shifts. **(signal-level)** finding a way to
 185 utilize intensity, which contains implicit semantic information, for synthetic data input.

186 **Scene-level Domain Cues from “Style”.** Input data for UDA task in LSS is typically limited to
 187 xyz coordinates. This provides no appearance information. We decompose LiDAR input into two
 188 conceptual components: “content” and “style”. Content refers to the object’s identity, whereas style
 189 defines its structural representation, like point density, motion blur, and rolling shutter distortion.
 190 Inspired by Huang & Belongie (2017), we find that feature statistics effectively capture this style in-
 191 formation through toy experiments. We compute the mean and standard deviation at various feature
 192 levels to form a compact “style embedding”. This embedding serves as a transferable domain cue,
 193 reflecting the synthetic-to-real domain gap. We conducted toy experiments to validate the rationale,
 194 and details are provided in the supplementary material.

195 **Scene-level Domain Cues to Class-level Domain Cues.** As shown in the left Fig. 1, classes such
 196 as `person`, `bicyclist`, `bicycle`, and `motorcycle` exhibit greater point structure distortion
 197 than classes like `road` or `building`. This is because these classes are typically dynamic and
 198 geometrically complex, making them more susceptible to motion blur and rolling shutter artifacts
 199 compared to static objects, as mentioned in Manivasagam et al. (2023). This motivates the use
 200 of class-aware domain cues to better capture class-specific structural differences. Additionally,
 201 semantically similar classes tend to share similar levels of dynamicness and complexity. For instance,
 202 classes like `car` and `truck` share visual and structural properties, thus also share similar dynamics
 203 and complexity. Consequently, features of synthetic `car` points should align not only with real `car`
 204 points but also with real `truck` points, facilitating more effective domain adaptation. Motivated
 205 by this observation, we introduce class-hierarchical domain cues that leverage semantic similarity
 206 among classes, enabling more precise modeling of class-dependent point structure discrepancies
 207 beyond global scene-level domain cues. We conducted toy experiments to validate the rationale behind
 208 these two domain cues. Further details are provided in the supplementary material.

209 **Utilizing LiDAR Intensity for Synthetic Inputs.** In conventional UDA settings, label supervision
 210 is applied only to the synthetic domain. This imbalance in learning signal often causes the seg-
 211 mentation model to overfit to synthetic data, and addressing this issue is non-trivial due to the lack
 212 of annotations in real data. However, LiDAR sensors naturally capture point-wise intensity values,
 213 which can serve as auxiliary information without requiring manual annotation. Since intensity val-
 214 ues inherently reflect class-specific characteristics through their dependence on distance and object
 215 reflectivity, they implicitly encode semantic class information (Song et al., 2002; Naich & Carrión,
 216 2024; Viswanath et al., 2025). Accordingly, treating intensity from real scans as labels allows the

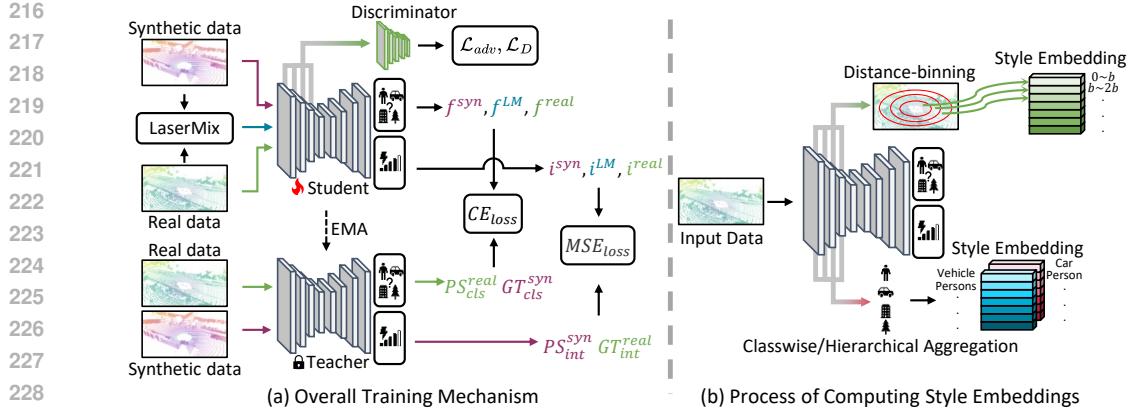


Figure 2: (a) Overview of our method. (b) Process of computing style embeddings. Our method extracts scene-wise, class-wise style embeddings (SSE , CSE) for adversarial learning. In parallel, we add an intensity prediction head to the feature extractor, enabling intensity self-training by predicting LiDAR point intensities. This head produces outputs i^{syn} , i^{LM} , and i^{real} , corresponding to synthetic, LaserMix-ed, and real inputs, respectively. f^{syn} , f^{LM} , and f^{real} are the original outputs obtained from the classifier head. PS_{cls}^{real} denotes the pseudo-labels obtained from the teacher model for real data, and PS_{int}^{syn} represents the pseudo-intensities generated by the teacher model for synthetic data. GT_{cls}^{syn} is the ground-truth class label from the synthetic dataset, and GT_{int}^{real} is the ground-truth intensity from real data.

model to learn implicit semantic information. Thus, prior works have utilized intensity to generate additional learning signals (Yuan et al., 2023; Viswanath et al., 2025). However, these studies did not attempt to adapt synthetic inputs using an intensity-based learning signal. This is because synthetic and real data have noticeably different intensity distributions, as shown in the right of Fig. 1. Consequently, using raw synthetic intensity of synthetic data as a learning signal can interfere with learning semantic cues from real data. To address this limitation, we extend prior work by proposing a self-training mechanism. This mechanism generates real-like pseudo-intensity for synthetic inputs and utilizes it during training. The strategy enables the model to learn semantic cues from real data and prevents overfitting to synthetic data.

4 METHODS

4.1 OVERVIEW

Fig. 2 shows the overview of the proposed framework. We build upon SAC-LM (Yuan et al., 2024), a recent self-training-based domain adaptation framework for LSS. SAC-LM, an extension of LaserMix (Kong et al., 2023b), is a method for 3D segmentation that reduces the domain gap. It achieves this by mixing target scans with synthetic source-like scans generated through point drop. A KL-based loss ensures consistency between predictions on raw and source-like target inputs. For a comprehensive methodology, please refer to Yuan et al. (2024). To enable intensity-guided adaptation, we augment the baseline with an additional MLP head dedicated to intensity prediction. This head produces outputs i^{syn} , i^{LM} , and i^{real} , corresponding to synthetic, LaserMix-ed, and real inputs, respectively. In parallel, f^{syn} , f^{LM} , and f^{real} are the original outputs obtained from the classifier head.

During self-training, we extract two types of style embeddings, scene-wise and class-wise from synthetic, real, and LaserMix-ed data. These embeddings are then used in an adversarial learning framework to align domain-specific styles across domains. The teacher model is updated as the exponential moving average (EMA) of the student model.

4.2 ADVERSARIAL LEARNING WITH STYLE EMBEDDINGS

4.2.1 SCENE-WISE STYLE EMBEDDING (SSE).

This embedding aims to capture domain gap factors that are class-agnostic and scene-wise, such as motion blur and rolling shutter distortions. As shown in Fig. 2(b), we divide the 3D points into radial distance bins, using a maximum distance threshold d and a binning interval b . Since point sparsity significantly varies with depth, we explicitly consider this variability by extracting style embeddings

separately for each distance bin. For each encoder block, we track the downsampled point features and retrieve their corresponding xyz coordinates. Within each distance bin b_i , we compute the mean and standard deviation of point features and concatenate them across all encoder blocks to form the final style embeddings $SSE_{\text{mean}} \in \mathbb{R}^{d/b, D}$ and $SSE_{\text{std}} \in \mathbb{R}^{d/b, D}$. These embeddings are then passed to a domain discriminator to facilitate adversarial learning. The formulation is expressed as follows:

$$SSE_{\text{mean}}^i = \bigoplus_{k=1}^B \mathbb{E}_p[f_k(x)], \quad SSE_{\text{std}}^i = \bigoplus_{k=1}^B \text{std}_p[f_k(x)]. \quad (1)$$

B is the number of encoder blocks. \bigoplus denotes the concatenation of the mean and standard deviation from each encoder block. $i \in \{\text{real, LM, syn}\}$ indicates data domains. D represents the total dimension after concatenation. $f_k(\cdot)$ refers to the k -th encoder block, and p represents an individual point.

4.2.2 CLASSWISE STYLE EMBEDDING (CSE).

As discussed in Sec. 3, point structure distortions differ across semantic classes. To capture such class-specific variations, we extend the scene-wise style embedding by also incorporating label information. For each encoder block, we downsample the labels consistently with the downampling of point features, thereby tracking labels corresponding to each point feature. Then the mean and standard deviation of point features are computed within each class. Finally, we concatenate these statistics across all classes to obtain the classwise style embeddings $CSE_{\text{mean}}^{\text{cls}} \in \mathbb{R}^{C, D}$ and $CSE_{\text{std}}^{\text{cls}} \in \mathbb{R}^{C, D}$. Fig. 2 (b) visualizes this process. The formulation is expressed as follows:

$$CSE_{\text{mean}}^{\text{cls}, i} = \bigoplus_{k=1}^B \mathbb{E}_{p \in c}[f_k(x)], \quad CSE_{\text{std}}^{\text{cls}, i} = \bigoplus_{k=1}^B \text{std}_{p \in c}[f_k(x)]. \quad (2)$$

C is number of classes and $p \in c$ indicates that point p belongs to a specific class c . In the real domain, encoder features are grouped by class using the pseudo-labels.

As mentioned in Sec. 3, semantically similar classes share dynamics and structural complexity. By leveraging this fact, we assume that classes within the same superclass exhibit similar domain gaps, which enables more effective compensation of class-dependent point structure differences during domain adaptation. In this reason, we further introduce class-hierarchical style embeddings to extend class-dependent feature alignment at the superclass level. To implement this, we define a class hierarchy automatically through GPT-4o. We filtered contradictory findings based on previous research that utilized class hierarchies (Kim et al., 2023; Park et al., 2025). The resulting hierarchy is presented in Table 1. Then, we aggregate features across superclass groups using the same mean and standard deviation computation used for CSE . This results in class-hierarchical style embeddings $CSE_{\text{mean}}^{\text{hier}} \in \mathbb{R}^{H, D}$ and $CSE_{\text{std}}^{\text{hier}} \in \mathbb{R}^{H, D}$. The formulation is expressed as follows:

$$CSE_{\text{mean}}^{\text{hier}, i} = \bigoplus_{k=1}^B \mathbb{E}_{p \in h}[f_k(x)], \quad CSE_{\text{std}}^{\text{hier}, i} = \bigoplus_{k=1}^B \text{std}_{p \in h}[f_k(x)]. \quad (3)$$

H is the number of superclasses, defined in class hierarchy. $p \in h$ indicates that point p belongs to a specific superclass h . Unlike Park et al. (2025), our method does not directly utilize the class as supervision. Instead, our approach extends the scene-wise domain gap to a class-wise domain gap.

4.2.3 STYLE EMBEDDING-BASED ADVERSARIAL LEARNING.

We perform adversarial learning using the previously described scene-wise and classwise, which are extracted from each synthetic, real, and LaserMix-ed data pipeline. By performing adversarial learning on these style embeddings, we obtain segmentation features that are invariant to domain gaps caused by differences in point structure. The formulation is expressed as follows:

$$\mathcal{L}_{\text{adv}} = -\frac{1}{N^{\text{real}}} \sum_i \|D(SE_i^{\text{real}})\|^2 - \frac{1}{N^{\text{LM}}} \sum_i \|D(SE_i^{\text{LM}})\|^2, \quad (4)$$

$$\mathcal{L}_{\text{D}} = -\frac{1}{N^{\text{syn}}} \sum_i \|D(SE_i^{\text{syn}})\|^2 - \frac{1}{N^{\text{real}}} \sum_i \|D(SE_i^{\text{real}}) - 1\|^2 - \frac{1}{N^{\text{LM}}} \sum_i \|D(SE_i^{\text{LM}}) - 1\|^2. \quad (5)$$

$SE \in \{SSE, CSE\}$ denotes the chosen style-embedding type, and $i \in \{\text{mean, std}\}$ selects either the mean or standard-deviation branch. For the discriminator architecture and adversarial loss, we adopt the same settings as in Yuan et al. (2024).

Superclass	SemanticKITTI Classes
Vehicle	car, bicycle, motorcycle, truck, other-vehicle
Person	pedestrian, bicyclist, motorcyclist
Traffic Element	pole, traffic-sign
Pavement	road, parking, sidewalk, other-ground
Natural	vegetation, trunk, terrain
Structure	building, fence
Superclass	SemanticPOSS Classes
Vehicle	car, bike
Person	rider, pedestrian
Traffic Element	traffic-sign, pole
Object	trash-can, cone-stone
Pavement	ground
Natural	trunk, plant
Structure	building, fence

Table 1: Superclass hierarchy for the SynLiDAR-to-SemanticKITTI (top) and SynLiDAR-to-SemanticPOSS (bottom) setups.

4.3 INTENSITY-GUIDED SELF-TRAINING

To mitigate overfitting to synthetic data and to enable the model to learn implicit semantic information of real data from synthetic inputs, we propose a self-training mechanism based on LiDAR intensity. An intensity prediction head is appended to the student and teacher models, respectively. Both networks adopt the same segmentation architecture, with the teacher model updated as the exponential moving average (EMA) of the student model. During training, the student model learns to predict real intensity values using ground-truth labels from real LiDAR data. For synthetic data, the teacher model predicts pseudo-intensity values, which the student model then learns to match. The overall training pipeline is illustrated in Fig. 2(a). The formulation is expressed as follows:

$$\mathcal{L}_{IS} = \sum_{k \in \{\text{real, LM, Syn}\}} \|\text{IH}(f(x)) - i^k\|^2. \quad (6)$$

IH denotes the intensity head, and f denotes the final output of the feature extractor. i denotes the intensity corresponding to the ground-truth intensity of real data, the pseudo-intensity of synthetic data, and the intensity values of LaserMix-ed points.

4.4 OPTIMIZATION AND INFERENCE

The total loss for student model combines classification, intensity, and adversarial terms mentioned above:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_{int} \mathcal{L}_{IS} + \lambda_{adv} \mathcal{L}_{adv}. \quad (7)$$

λ_{int} and λ_{adv} were 10 and 1e-4 for main experiments, respectively. The teacher updates by EMA of the student. At test time, we use the student alone without an intensity head.

5 EXPERIMENTS

5.1 EXPERIMENTS SETTINGS

We use the SynLiDAR dataset (Xiao et al., 2022) as our synthetic source domain. SynLiDAR is a large-scale synthetic point cloud dataset generated from diverse urban driving scenes built on the Unreal Engine 4 platform. The LiDAR sensor is simulated based on the Velodyne HDL-64E, producing high-resolution point clouds with up to 64 beams. For real target domains, we use the SemanticKITTI (Behley et al., 2019) and SemanticPOSS (Pan et al., 2020) datasets. SemanticKITTI is collected using the Velodyne HDL-64E sensor with 60 beams and contains real-world point cloud data collected across 22 driving sequences in urban environments. SemanticPOSS uses the Velodyne Alpha Prime sensor with 40 beams and contains point clouds captured within a university campus. Following the adaptation setup in previous work (Yuan et al., 2024), we downsample the SynLiDAR point clouds to 40 beams when training on the SemanticPOSS target. Consistent with Yuan et al. (2024), we adopt MinkUnet32 (Choy et al., 2019) as base segmentation architecture. We train our models for 200,000 iterations, consistent with DGT-ST. Each experiment is conducted on a single NVIDIA A6000 GPU.

	Mech.	car	bicycle	motorcycle	truck	oth.-v.	pers.	bi.clst	mt.clst	road	parki.	sidew.	otherrg.	build.	fence	veget.	trunk	ten.	pole	traf.	mIoU
Source only	-	35.9	7.5	10.7	0.6	2.9	13.3	44.7	3.4	21.8	6.9	29.6	0.0	34.1	7.4	62.9	26.0	35.5	30.3	14.1	20.4
AdaptSegNet Tsai et al. (2018)	A	52.1	10.8	11.2	2.6	9.6	15.1	35.9	2.6	62.2	10.4	41.3	0.1	58.1	17.1	68.0	38.4	38.7	35.9	20.4	27.9
CLAN Luo et al. (2019)	A	51.0	15.8	16.8	2.2	7.8	18.7	46.8	3.0	68.9	11.1	44.9	0.1	59.6	17.5	71.7	41.1	44.0	37.7	19.8	30.5
ADVENT Vu et al. (2019)	A	59.9	13.8	14.6	3.0	8.0	17.7	45.8	3.0	67.6	11.3	45.6	0.1	61.7	15.8	72.4	41.5	47.0	34.5	15.3	30.5
FADA Wang et al. (2020)	A	49.9	6.7	5.1	2.5	10.0	5.7	26.6	2.3	65.8	10.8	37.8	0.1	60.3	21.5	60.4	37.2	31.9	35.4	17.4	25.6
MRNet Zheng & Yang (2019)	A	49.5	11.0	12.2	2.2	8.6	16.0	46.4	2.7	60.0	10.5	41.9	0.1	55.1	16.5	68.1	38.0	40.7	36.5	20.8	28.3
PMAN Yuan et al. (2023)	A	71.0	14.9	24.8	1.6	6.6	23.6	61.1	5.5	75.3	10.5	54.1	0.1	47.9	17.4	69.6	38.6	61.5	37.0	18.6	33.7
PCAN Yuan et al. (2024)	A	85.0	<u>17.5</u>	27.4	10.4	<u>11.9</u>	27.5	<u>63.7</u>	2.6	78.1	13.5	50.1	0.1	68.5	<u>20.0</u>	76.2	41.3	45.7	41.0	21.8	37.0
CoSMix Saltori et al. (2022)	S	56.4	10.2	20.8	2.1	13.0	25.6	41.3	2.2	67.4	8.2	43.4	0.0	57.9	12.2	68.4	44.8	35.0	42.1	17.0	29.9
LaserMix Kong et al. (2023b)	S	90.3	7.8	37.2	<u>2.3</u>	2.4	40.6	49.1	<u>5.1</u>	80.5	9.9	57.4	0.0	57.6	3.4	77.6	46.6	<u>60.1</u>	42.0	13.6	36.0
DGT-ST Yuan et al. (2024)	S	92.9	<u>17.3</u>	43.4	15.0	6.1	49.2	54.2	4.2	<u>86.4</u>	<u>19.1</u>	<u>62.3</u>	0.0	<u>78.2</u>	9.2	<u>83.3</u>	<u>56.0</u>	59.1	<u>51.2</u>	<u>32.3</u>	<u>43.1</u>
Ours	A+S	90.6	14.4	51.9	<u>11.3</u>	4.3	59.9	66.7	3.2	88.8	27.4	64.7	0.0	69.9	11.7	79.5	60.1	55.0	51.3	39.2	44.7

Table 2: Comparison results on *SynLiDAR* → *SemanticKITTI* benchmark. **Bold**=highest, underline=second highest in each column. A/S denotes adversarial training (A) / self-training (S).

Methods	Mech.	bi.clst	car	trunk	veget.	traf.	pole	garb.	build.	cone.	fence	bi.cle	ground	pers.	mIoU
Source only	-	47.2	43.6	37.8	70.3	11.1	33.8	19.5	67.9	11.2	19.9	9.6	77.9	47.8	38.3
AdaptSegNet Tsai et al. (2018)	A	43.9	48.2	39.0	69.6	15.5	33.6	21.3	64.3	12.7	25.0	11.6	76.0	49.9	39.3
CLAN Luo et al. (2019)	A	43.9	46.6	41.3	71.0	15.1	34.3	20.4	69.6	9.5	23.2	12.0	75.1	51.3	39.5
ADVENT Vu et al. (2019)	A	44.6	47.6	40.3	71.2	15.6	35.6	22.0	68.4	10.6	25.9	10.4	76.7	52.3	40.1
FADA Wang et al. (2020)	A	39.6	41.2	38.8	69.2	16.3	32.1	18.1	67.9	11.5	22.0	13.0	71.4	47.9	37.6
MRNet Zheng & Yang (2019)	A	43.5	47.2	39.1	70.4	15.5	32.8	22.0	66.1	13.2	24.2	11.2	76.8	50.0	39.4
PMAN Yuan et al. (2023)	A	52.6	61.5	46.8	75.1	18.8	36.5	21.4	74.7	18.3	25.8	37.5	73.7	61.9	46.5
PCAN Yuan et al. (2024)	A	48.6	62.1	37.5	74.0	<u>23.9</u>	31.4	22.2	76.9	6.5	41.9	11.9	79.1	61.2	44.4
CoSMix Saltori et al. (2022)	S	53.6	47.6	44.8	75.1	16.8	<u>37.9</u>	25.3	72.7	19.9	39.7	10.8	80.0	56.5	44.6
LaserMix Kong et al. (2023b)	S	<u>58.4</u>	61.3	<u>47.7</u>	69.0	21.9	<u>39.5</u>	30.9	61.0	16.1	36.5	7.1	<u>79.5</u>	62.6	45.5
DGT-ST Yuan et al. (2024)	S	<u>55.1</u>	<u>70.7</u>	46.1	<u>74.2</u>	30.1	36.3	<u>44.1</u>	<u>81.0</u>	4.3	62.8	10.3	78.5	<u>67.2</u>	50.8
Ours	A+S	63.5	71.2	50.4	75.3	21.7	37.0	47.1	81.4	1.0	<u>60.6</u>	7.5	79.0	69.7	51.2

Table 3: Comparison results on *SynLiDAR* → *SemanticPOSS* benchmark. **Bold**=highest, underline=second highest in each column. A/S denotes adversarial training (A) / self-training (S).

5.2 MAIN EXPERIMENTS

SynLiDAR-to-SemanticKITTI. As shown in Table 2, our method achieves 44.7 mIoU on the SynLiDAR-to-SemanticKITTI benchmark, outperforming the previous state-of-the-art DGT-ST by +1.6 mIoU. Notably, we observe substantial gains in complex and dynamic object classes such as **person** (**+10.7**), **motorcycle** (**+8.5**), and **bicyclist** (**+12.5**) mIoU. These classes are characterized by irregular motion and geometrically complex shapes, which typically result in larger domain gaps. The consistent improvements on these classes indicate that our method effectively captures point structure differences, a key factor in the domain gap. This also confirms that the classwise style embeddings successfully reduce the additional domain shift caused by class-specific complex and dynamic natures. In addition, we observe gains of +9.7, +3.2, and +1.9 mIoU in classes such as **parking**, **sidewalk**, and **road**, respectively. These results indicate that our method leads to performance improvements across most semantic classes.

SynLiDAR-to-SemanticPOSS. As shown in Table 3, our method achieves 51.2 mIoU on the SynLiDAR-to-SemanticPOSS benchmark, surpassing the previous state-of-the-art by +0.4 mIoU. Compared to SemanticKITTI, the SemanticPOSS dataset uses a 40-beam LiDAR sensor, resulting in sparser point clouds during data acquisition. Consequently, reducing the domain gap is harder and yields modest gains, yet absolute scores stay high because the dataset contains fewer samples and classes. Moreover, SemanticPOSS (Pan et al., 2020) contains mostly static and few scenes and only a few classes, leading to a high baseline mIoU and leaving little room for further improvement. Despite this, our method still achieves substantial improvements in complex object classes, including **bicyclist** (**+8.4**), **garbage can** (**+3.0**), and **person** (**+2.5**) mIoU. We also observe gains in **trunk** (+4.3), **ground** (+0.5), and **vegetation** (+1.1), demonstrating consistent performance benefits across various semantic classes.

5.3 ABLATION STUDY

As shown in Table 4, we conduct an ablation study by incrementally adding our proposed components on baseline model, SAC-LM (Yuan et al., 2024). Applying scene-wise adversarial learning yields a +5.6 mIoU gain, demonstrating that the scene-wise style embedding effectively captures global domain gaps, such as those caused by noise, rolling shutter or motion blur. Adding classwise adversarial learning provides an additional +1.1 mIoU gain, indicating the importance of modeling class-specific domain gaps in addition to scene-level shifts and considering dynamicity similarity between classes during adaptation is beneficial, even across domains. Finally, applying intensity

ID	S-wise	C-wise	Int.	mIoU	Increments
0				37.0	-
1	✓			42.6	+5.6
2	✓	✓		43.7	+1.1
3	✓	✓	✓	44.7	+1.0

Table 4: Ablation study on the effect of each component. The base model, SAC-LM Yuan et al. (2024), gradually adds Scene-wise (S-wise), Class-wise(C-wise) style embedding-based adversarial learning, and Intensity self-training (Int.) losses.

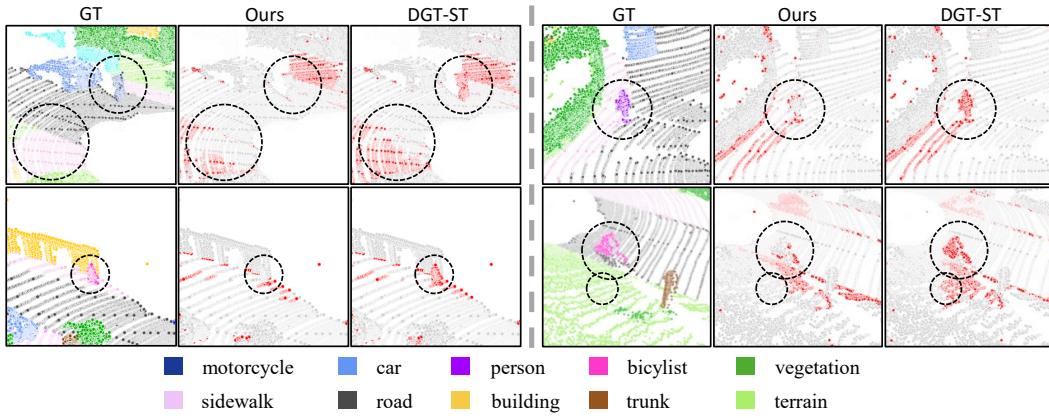


Figure 3: Qualitative results on *SynLiDAR* → *SemanticKITTI* benchmark. Red=incorrect, Gray=correct. Our method shows improved performance on structurally complex and dynamic objects such as motorcycle, person, and bicyclist.

self-training results in a +1.0 mIoU gain, showing that learning implicit semantic information from synthetic inputs, via pseudo-intensity, is beneficial for domain adaptation.

5.4 QUALITATIVE RESULTS

Qualitative results on the *SynLiDAR*-to-*SemanticKITTI* benchmark are presented in Fig. 3. As shown in the left of row 1, our method achieves better segmentation on the motorcycle class compared to previous methods, consistent with the analysis in Sec. 5.2. Additionally, improvements on the person class can be observed in the left of row 2 and the right of row 1. In the right of row 2, our method also shows better performance on the bicyclist class. These results support our findings in Sec. 5.2, where we highlighted that our method yields more accurate predictions for dynamic, and structurally complex objects compared to prior approaches.

6 CONCLUSION

We presented a unified framework to address the synthetic-to-real LiDAR domain gap, a key challenge for annotation-efficient semantic segmentation. Unlike general UDA methods, our approach directly targets sensor-induced structural distortions and class-dependent variability, while extending intensity learning to the synthetic data pipeline via self-training. We proposed scene-wise and class-wise style embeddings for adversarial learning and an intensity-guided self-training strategy. Our method achieved state-of-the-art results on two major adaptation benchmarks, highlighting its potential for label-efficient LiDAR segmentation.

REFERENCES

Angelika Ando, Spyros Gidaris, Andrei Bursuc, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5240–5250, 2023.

486 Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and
 487 Jürgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In
 488 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9297–9307, 2019.
 489

490 Jaesung Choe, Chunghyun Park, Francois Rameau, Jaesik Park, and In So Kweon. Pointmixer: Mlp-
 491 mixer for point cloud understanding. In *European Conference on Computer Vision*, pp. 620–640.
 492 Springer, 2022.

493 Christopher Choy, Jun Young Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski
 494 convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision
 495 and pattern recognition*, pp. 3075–3084, 2019.

496

497 Martin Hahner, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Fog simulation on real li-
 498 dar point clouds for 3d object detection in adverse weather. In *Proceedings of the IEEE/CVF
 499 international conference on computer vision*, pp. 15283–15292, 2021.

500

501 Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc
 502 Van Gool. Lidar snowfall simulation for robust 3d object detection. In *Proceedings of the
 503 IEEE/CVF conference on computer vision and pattern recognition*, pp. 16364–16374, 2022.

504

505 Georg Hess, Carl Lindström, Maryam Fatemi, Christoffer Petersson, and Lennart Svensson. Splatad:
 506 Real-time lidar and camera rendering with 3d gaussian splatting for autonomous driving. In *Pro-
 ceedings of the Computer Vision and Pattern Recognition Conference*, pp. 11982–11992, 2025.

507

508 Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance nor-
 509 malization. In *Proceedings of the IEEE international conference on computer vision*, pp. 1501–1510,
 510 2017.

511

512 Hyeonseong Kim, Yoonsu Kang, Changgyoon Oh, and Kuk-Jin Yoon. Single domain generalization
 513 for lidar semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision
 and pattern recognition*, pp. 17587–17598, 2023.

514

515 Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou,
 516 Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *Pro-
 ceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 228–240, 2023a.

517

518 Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. Lasermix for semi-supervised lidar semantic
 519 segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
 520 Recognition*, pp. 21705–21715, 2023b.

521

522 Xin Lai, Yukang Chen, Fanbin Lu, Jianhui Liu, and Jiaya Jia. Spherical transformer for lidar-based
 523 3d recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
 524 Recognition*, pp. 17545–17555, 2023.

525

526 Guangrui Li, Guoliang Kang, Xiaohan Wang, Yunchao Wei, and Yi Yang. Adversarially mask-
 527 ing synthetic to mimic real: Adaptive noise injection for point cloud segmentation adaptation.
 528 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.
 20464–20474, 2023.

529

530 Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain
 531 shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of
 532 the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2507–2516, 2019.

533

534 Sivabalan Manivasagam, Shenlong Wang, Kelvin Wong, Wenyuan Zeng, Mikita Sazanovich,
 535 Shuhan Tan, Bin Yang, Wei-Chiu Ma, and Raquel Urtasun. Lidarsim: Realistic lidar simula-
 536 tion by leveraging the real world. In *Proceedings of the IEEE/CVF Conference on Computer
 Vision and Pattern Recognition*, pp. 11167–11176, 2020.

537

538 Sivabalan Manivasagam, Ioan Andrei Bârsan, Jingkang Wang, Ze Yang, and Raquel Urtasun. To-
 539 wards zero domain gap: A comprehensive study of realistic lidar simulation for autonomy testing.
 In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8272–8282,
 2023.

540 Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate
 541 lidar semantic segmentation. In *2019 IEEE/RSJ international conference on intelligent robots*
 542 and systems (IROS), pp. 4213–4220. IEEE, 2019.

543 Ammar Yasir Naich and Jesús Requena Carrión. Lidar-based intensity-aware outdoor 3d object
 544 detection. *Sensors*, 24(9):2942, 2024.

545 Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A
 546 point cloud dataset with large quantity of dynamic instances. In *2020 IEEE intelligent vehicles*
 547 symposium (IV), pp. 687–693. IEEE, 2020.

548 Junsung Park, Hwijeong Lee, Inha Kang, and Hyunjung Shim. No thing, nothing: Highlighting
 549 safety-critical classes for robust lidar semantic segmentation in adverse weather. In *Proceedings*
 550 of the Computer Vision and Pattern Recognition Conference, pp. 6690–6699, 2025.

551 Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets
 552 for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision*
 553 and pattern recognition, pp. 652–660, 2017.

554 Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Elisa Ricci, and Fabio Poiesi. Cos-
 555 mix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. In *European*
 556 conference on computer vision, pp. 586–602. Springer, 2022.

557 Jeong-Heon Song, Soo-Hee Han, Ki Yun Yu, and Yong-Il Kim. Assessing the possibility of land-
 558 cover classification using lidar intensity data. *International archives of photogrammetry remote*
 559 *sensing and spatial information sciences*, 34(3/B):259–262, 2002.

560 Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Search-
 561 ing efficient 3d architectures with sparse point-voxel convolution. In *European conference on*
 562 *computer vision*, pp. 685–702. Springer, 2020.

563 Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette,
 564 and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In
 565 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6411–6420, 2019.

566 Hugues Thomas, Yao-Hung Hubert Tsai, Timothy D Barfoot, and Jian Zhang. Kpconvx: Moderniz-
 567 ing kernel point convolution with kernel attention. In *Proceedings of the IEEE/CVF Conference*
 568 *on Computer Vision and Pattern Recognition*, pp. 5525–5535, 2024.

569 Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan
 570 Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings*
 571 of the IEEE conference on computer vision and pattern recognition, pp. 7472–7481, 2018.

572 Patrik Vacek, Otakar Jašek, Karel Zimmermann, and Tomáš Svoboda. Learning to predict lidar
 573 intensities. *IEEE Transactions on Intelligent Transportation Systems*, 23(4):3556–3564, 2021.

574 Kasi Viswanath, Peng Jiang, and Srikanth Saripalli. Reflectivity is all you need!: Advancing lidar
 575 semantic segmentation. *IFAC-PapersOnLine*, 59(3):43–48, 2025.

576 Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adver-
 577 sarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of*
 578 the IEEE/CVF conference on computer vision and pattern recognition, pp. 2517–2526, 2019.

579 Haoran Wang, Tong Shen, Wei Zhang, Ling-Yu Duan, and Tao Mei. Classes matter: A fine-grained
 580 adversarial approach to cross-domain semantic segmentation. In *European conference on com-*
 581 *puter vision*, pp. 642–659. Springer, 2020.

582 Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Transfer learning from
 583 synthetic to real lidar point cloud for semantic segmentation. In *Proceedings of the AAAI confer-*
 584 *ence on artificial intelligence*, volume 36, pp. 2795–2803, 2022.

585 Xiang Xu, Lingdong Kong, Hui Shuai, and Qingshan Liu. Frnet: Frustum-range networks for
 586 scalable lidar segmentation. *IEEE Transactions on Image Processing*, 2025.

594 Zhimin Yuan, Ming Cheng, Wankang Zeng, Yanfei Su, Weiquan Liu, Shangshu Yu, and Cheng
595 Wang. Prototype-guided multitask adversarial network for cross-domain lidar point clouds se-
596 mantic segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–13, 2023.
597

598 Zhimin Yuan, Wankang Zeng, Yanfei Su, Weiquan Liu, Ming Cheng, Yulan Guo, and Cheng Wang.
599 Density-guided translator boosts synthetic-to-real unsupervised domain adaptive segmentation of
600 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
Recognition*, pp. 23303–23312, 2024.
601

602 Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In
603 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16259–16268,
604 2021.

605 Zhedong Zheng and Yi Yang. Unsupervised scene adaptation with memory regularization in vivo.
606 *International Joint Conference on Artificial Intelligence*, 2019.
607

608 Hui Zhou, Xinge Zhu, Xiao Song, Yuxin Ma, Zhe Wang, Hongsheng Li, and Dahua Lin. Cylin-
609 der3d: An effective 3d framework for driving-scene lidar semantic segmentation. *arXiv preprint
arXiv:2008.01550*, 2020.
610

611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647