remote sensing

MDPI

Article

Context-Aware Feature Adaptation for Mitigating Negative Transfer in 3D LiDAR Semantic Segmentation

Lamiae El Mendili *, Sylvie Daniel and Thierry Badard

Department of Geomatics Sciences, Université Laval, Québec, QC G1V 0A6, Canada; sylvie.daniel@scg.ulaval.ca (S.D.); thierry.badard@scg.ulaval.ca (T.B.)
* Correspondence: lamiae.el-mendili.1@ulaval.ca

Abstract

Semantic segmentation of 3D LiDAR point clouds is crucial for autonomous driving and urban modeling but requires extensive labeled data. Unsupervised domain adaptation from synthetic to real data offers a promising solution, yet faces the challenge of negative transfer, particularly due to context shifts between domains. This paper introduces Context-Aware Feature Adaptation, a novel approach to mitigate negative transfer in 3D unsupervised domain adaptation. The proposed approach disentangles object-specific and context-specific features, refines source context features through cross-attention with target information, and adaptively fuses the results. We evaluate our approach on challenging synthetic-to-real adaptation scenarios, demonstrating consistent improvements over state-of-the-art domain adaptation methods with up to 7.9% improvement in classes subject to context shift. Our comprehensive domain shift analysis reveals a positive correlation between context shift magnitude and performance improvement. Extensive ablation studies and visualizations further validate the efficacy in handling context shift for 3D semantic segmentation.

Keywords: UDA; 3D point clouds; domain shift; 3D semantic segmentation; negative transfer

check for updates

Academic Editor: Li Li

Received: 28 May 2025
Revised: 27 July 2025
Accepted: 8 August 2025
Published: 14 August 2025

Citation: El Mendili, L.; Daniel, S.; Badard, T. Context-Aware Feature Adaptation for Mitigating Negative Transfer in 3D LiDAR Semantic Segmentation. *Remote Sens.* **2025**, *17*, 2825. https://doi.org/10.3390/rs17162825

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Semantic segmentation of outdoor mobile mapping point clouds is vital for applications, like autonomous driving and digital twins of cities. Supervised deep learning methods achieve strong results but rely on large annotated datasets, which are costly and time-consuming to obtain. This has motivated the exploration of transfer learning using synthetic datasets with readily available labels [1,2]. However, the domain shift between synthetic and real data introduces challenges, including sensor variations, geographical differences, and class discrepancies in urban scenes, causing a generalization gap in real-world scenarios. Unsupervised domain adaptation (UDA) methods have been proposed to address this gap [3–7]. Despite progress, UDA methods face challenges, like negative transfer (NT) [8]. NT can be defined as the degradation in target domain performance that occurs when knowledge learned from the source domain is inappropriate or misleading for the target. A principal driver of NT in 3D semantic segmentation is cross-domain *context shift*, which is a mismatch in how semantic classes co-occur spatially between domains. Context shift denotes the variation in spatial co-occurrence patterns and scene-level relationships between semantic classes across source and target domains, such as differences in where or with what other objects a class typically appears. As an illustrative case, motorcycles in a synthetic dataset may frequently appear riding on the roadway, whereas in real-world

scans they are more often parked on sidewalks or terrain. This can mislead the model and produce class confusion in the target domain.

UDA methods for 3D semantic segmentation are commonly divided into self-training and adversarial. Self-training approaches may suffer from error propagation [9], while adversarial methods can misalign features, losing domain-specific context [10]. Current UDA efforts primarily focus on cross-domain transferability, often transferring domain-specific features from the source to the target, exacerbating NT [8]. They operate by aligning global or class-level feature distributions, often ignoring spatial dependencies and semantic co-occurrence patterns that vary significantly between domains. Therefore, modeling and mitigating context shift remains an underexplored but critical issue of UDA for 3D semantic segmentation.

This paper proposes Context-Aware Feature Adaptation (CAFA) to tackle NT caused by context shifts in 3D UDA. CAFA selectively adapts context-specific features while preserving domain-invariant object features. CAFA's three-step process first disentangles object and context features, refines source context features using target context via cross-attention, and fuses refined context features with object features. Our contributions are as follows:

- A novel feature disentanglement technique for separating object-specific and context-specific information in 3D point cloud data and a cross-attention mechanism for refining source context features using target domain information.
- A modular feature refinement framework that can be integrated into various UDA techniques to enhance their performance by addressing negative transfer arising from context shift.
- A method to quantify context shift to enable an analytical evaluation of performance improvements relative to context variability.
- Extensive experiments demonstrating our approach reduces NT and improves UDA performance on challenging 3D semantic segmentation tasks.

2. Related Work

Unsupervised domain adaptation for 3D semantic segmentation. UDA for 3D semantic segmentation enables knowledge transfer from labeled synthetic data to unlabeled real-world LiDAR scans. The key challenge is context shift due to different class co-occurrences between domains. Existing methods fall into adversarial [2,4,7,11–16] and self-training methods [3,5,6]. CycleGAN [17] can be used for translating real Bird-Eye-View images (BEVs) into synthetic BEVs obtained from synthetic point clouds [11]. Zhao et al. [12] adversarially simulate LiDAR dropout noise on real data using a synthetic dataset. Xiao et al. [2] decompose the synthetic-to-real gap into an appearance component and a sparsity component and then align the synthetic and real feature distribution at the input level and feature level. Yi et al. [13] propose to conduct UDA through the auxiliary task of 3D surface completion to transfer knowledge between different LiDAR sensors. Yuan et al. [4] propose a category-level adversarial alignment to translate point density between domains with an adaptive adversarial loss reweighting and source-aware consistency loss. Li et al. [14] simulate the pattern of target noise to bridge the domain gap using a learnable masking module. Self-training methods use target pseudo-labels to bridge the domain gap. Saltori et al. [3] propose a self-training approach that employs a semantic mixing strategy to augment the data and mitigate domain shift. Xiao et al. [6] introduce LiDAR-specific augmentation through scene-level swapping and instance-level rotation. Zhao et al. [5] construct a bridge domain through spatial, intensity, and semantic distribution mixing. While some approaches tackle aspects of NT [7,14], they do not explicitly address context shifts due to varying class co-occurrences. These approaches typically focus on input-level

transformations, global domain alignment, or class-wise alignment. They overlook contextual relationships and spatial dependencies, which are key sources of domain shift in urban LiDAR data. Specifically, they do not explicitly address how changes in object co-occurrence and scene composition between domains lead to negative transfer during adaptation. They also struggle to capture different contextual relationships, leading to suboptimal feature transfer in complex urban environments. Our proposed method addresses this gap with a novel context-aware feature disentanglement approach and cross-attention refinement, specifically designed to handle spatial dependencies and class co-occurrence variations to mitigate NT caused by context shifts in 3D LiDAR segmentation.

Negative Transfer in UDA. The effectiveness of UDA methods can be compromised by NT, where knowledge from the source domain harms target performance. NT often occurs when domains are dissimilar [18] or when source models overfit to domain-specific features. This is problematic in outdoor mobile mapping point clouds, where context shifts between domains can occur due to complex spatial relationships, class co-occurrences, and varying point densities across different environments. NT mitigation approaches can be categorized into two main categories [8]: data transferability enhancement (DTE) and model transferability enhancement (MTE). DTE methods focus on enhancing the quality of input instances or features. These include domain decomposition [19,20], which performs UDA on subdomains by partitioning the source and target into different parts. For instance, Zhu et al. [19] minimize class-wise discrepancy rather than global alignment. Intermediate domain construction methods [21,22] bridge the domain gap through constructed intermediate domains. Other DTE approaches include instance selection and weighting [23] and feature enhancement methods, such as batch spectral shrinkage [24] and adaptive channel weighting [25]. MTE approaches aim to enhance model transferability. These include TransNorm [25], which adapts normalization statistics, and parameter selection methods, such as TransPar [26], which identify and select transferable parameters from the source model. Parameter regularization methods, including co-tuning [27], side-tuning [28], and concept-wise fine-tuning [29], also fall under the MTE category. While these strategies improve feature transferability, none explicitly model spatial context or account for semantic co-occurrence discrepancies across domains, which are critical factors for 3D point clouds due to the varying scene compositions across different urban environments. This gap highlights the need for a method that tackles NT by adapting context, rather than relying solely on distributional alignment.

Our proposed method addresses the limitations of existing approaches in handling context shift by introducing a context-aware feature disentanglement approach coupled with cross-attention refinement. This allows for targeted adaptation of contextual features while preserving object-specific information.

3. Materials and Methods

Our proposed approach explicitly handles contextual differences during the adaptation process and is illustrated in Figure 1. We consider a set of source domain samples $S = (X^s, Y^s)$ and target domain samples $T = (X^t, Y^t)$, $X^s \in \mathbb{R}^{n_s \times d_{in}} \sim P_s$ are source input points and $X^t \in \mathbb{R}^{n_t \times d_{in}} \sim P_t$ are target points. P_s and P_t denote source and target feature distributions, respectively, and d_{in} is the number of input features. $Y^s \in \mathcal{Y}$, $\mathcal{Y} = \{1, 2, \ldots, C\}$ are source labels for C classes. n_s and n_t are the number of source and target points, respectively. The target domain shares the same classes as the source domain; the target labels Y^t are unknown.

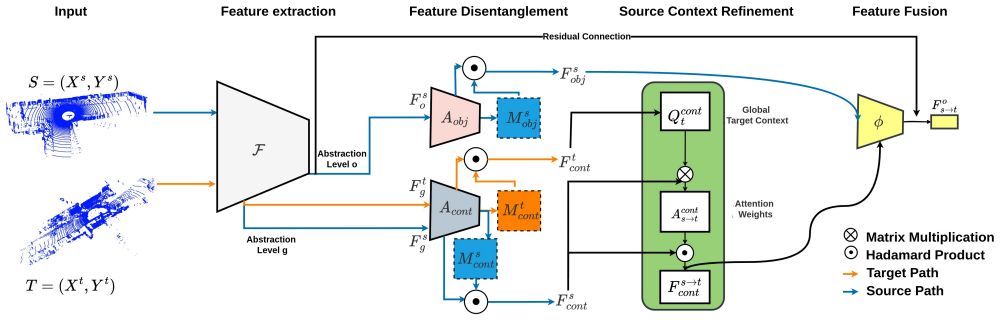


Figure 1. Architecture of the proposed CAFA method. The pipeline illustrates the key components: feature extraction, feature disentanglement, source context refinement using global target context, and feature fusion. Blue paths represent source domain processing, orange paths represent target domain processing.

3.1. Context-Aware Feature Adaptation

In the following sections, we detail each component of our proposed Context-Aware Feature Adaptation (CAFA) method.

3.1.1. Object and Context Feature Disentanglement

We decompose source and target features into object-specific and context-specific components to adapt contextual information while preserving object-specific features. Features are extracted at two abstraction levels through the backbone network $\mathcal{F}(.)$ as shown in Equation (1):

$$\{F^s_l\}^L_{l=1} = \mathcal{F}(X^s), \tag{1}$$

where $F^s_l \in \mathbb{R}^{n^s_l \times d_l}$ are source features at level l, $L = 2$ is the number of levels, d_l is the feature dimension, and n^s_l is the number of source points at level l. This applies to both source and target domains. Source features are disentangled using two attention modules to generate context (Equation (2)) and object attention maps (Equation (3)), which modulate the features via element-wise multiplication to produce context-specific (Equation (4)) and object-specific features (Equation (5)).

$$M^s_{cont} = \sigma(A_{cont}(F^s_g)), \tag{2}$$

$$M^s_{obj} = \sigma(A_{obj}(F^s_o)), \tag{3}$$

$$F^s_{cont} = M^s_{cont} \odot F^s_g, \tag{4}$$

$$F^s_{obj} = M^s_{obj} \odot F^s_o, \tag{5}$$

We select the lower abstraction levels ($g = 1$) to extract context features and the higher level ($o = L$) for object-specific features. This choice is supported by prior research showing that in deep convolutional networks, lower layers tend to capture local patterns such as textures or geometric context, while higher layers encode abstract, object-level semantics [30,31]. A_{cont} and A_{obj} are 1×1 convolutional layers, and σ is the Softmax function. Lower levels (g) capture low-level context patterns, while higher levels (o) encode abstract object information. Only context-specific features F^t_{cont} are extracted for the target domain to guide source context refinement.

This disentanglement process uses multi-scale features and attention for a soft separation, allowing for shared information between object and context features. Unlike strict orthogonality constraints (ineffective in our experiments), our method allows for soft disentanglement that maintains some shared information between object and context features.

3.1.2. Source Context Feature Refinement

We use cross-attention to refine source context features using target domain information. The process involves three steps:

$$C_t^{cont} = \frac{1}{n_t} \sum_{i=1}^{n_t} F_{con}^t, \tag{6}$$

$$Q_t^{cont} = A(C_t^{cont}), \tag{7}$$

$$A_{s \to t}^{cont} = \sigma(Q_t^{cont} \cdot F_{cont}^s), \tag{8}$$

$$F_{cont}^{s \to t} = F_{cont}^s \odot A_{t \to s}^{cont}, \tag{9}$$

First, we compute the global target context (Equation (6)) using global average pooling. Next, we compute a query context vector Q_t^{cont} using a 1×1 convolutional layer A (Equation (7)) that is then used to calculate attention weights for the source context features (Equation (8)). Finally, we refine source context features using these attention weights through element-wise multiplication (Equation (9)). This mechanism allows the model to selectively focus on source context features relevant to the target domain and mitigate context shift effects.

3.1.3. Cross-Domain Feature Fusion

We fuse the refined source context features with the original source object features following Equation (10):

$$F_o^{s \to t} = \phi([F_{cont}^{s \to t}; F_{obj}^s]) + \gamma F_o^s, \tag{10}$$

ϕ is a learnable 1×1 convolutional layer and $[;]$ denotes channel-wise concatenation. A residual connection with a learnable scalar γ (initialized small) allows the network to retain information from the original representation and adjust its importance during training. This process aligns source context features with the target domain while preserving important object-specific information for segmentation.

Algorithm 1 synthesizes the main steps of the proposed approach.

Algorithm 1 Context-Aware Feature Adaptation (CAFA)

Require: Source point cloud X_s, Target point cloud X_t, Backbone network F
Ensure: Adapted source features $F_{s \to t}$

function DISENTANGLEFEATURES(X, F)
  $F_1, F_2 \leftarrow F(X)$ ▷ Extract multi-scale features
  $M_{obj} \leftarrow \sigma(\text{Conv}_{obj}(F_2))$ ▷ Object attention
  $M_{cont} \leftarrow \sigma(\text{Conv}_{cont}(F_1))$ ▷ Context attention
  return $M_{obj} \odot F_2, M_{cont} \odot F_1$
end function
function CROSSATTENTIONREFINEMENT($F_{s,cont}, F_{t,cont}$)
  $C_t \leftarrow \frac{1}{N_t} \sum_{i=1}^{N_t} F_{t,cont}^i$ ▷ Global target context
  $A \leftarrow \sigma(C_t \cdot F_{s,cont}^T)$ ▷ Cross-attention weights
  return $F_{s,cont} \odot A$
end function
function FEATUREFUSION(F_{cont}, F_{obj}, F_s)
  $F_{fused} \leftarrow \text{Conv}_{fusion}([F_{cont}; F_{obj}])$
  return $F_{fused} + \gamma F_s$ ▷ γ: learnable parameter
end function
$F_{s,obj}, F_{s,cont} \leftarrow \text{DisentangleFeatures}(X_s, F)$
$F_{t,cont} \leftarrow \text{DisentangleFeatures}(X_t, F)[2]$
$F_{s \to t,cont} \leftarrow \text{CrossAttentionRefinement}(F_{s,cont}, F_{t,cont})$
$F_{s \to t} \leftarrow \text{FeatureFusion}(F_{s \to t,cont}, F_{s,obj}, F_s)$
return $F_{s \to t}$

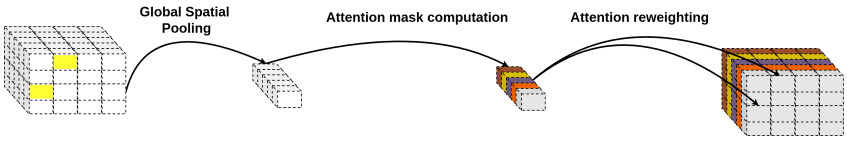
3.2. Relationship to Other Attention Mechanisms

3.2.1. Vector Attention

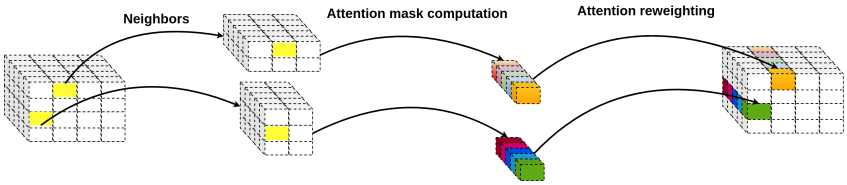
Vector self-attention was introduced in Point Transformer [32] for a set of feature vectors $\mathcal{X} = \{x_i\}_i$ as

$$y_i = \sum_{\mathbf{x}_j \in \mathcal{X}} \rho\big(\gamma\big(\big(\varphi(\mathbf{x}_i) - \psi(\mathbf{x}_j)\big) + \delta\big)\big) \odot \alpha(\mathbf{x}_j) \tag{11}$$

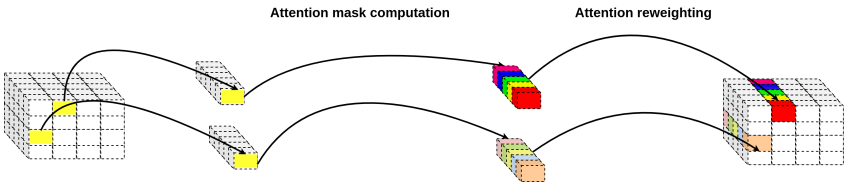
y_i is the output feature. φ, ψ, and α are point-wise feature transformations, such as multilayer perceptrons (MLPs) or linear projections. δ is a positional encoding function and γ is a mapping function (e.g., an MLP) that produces attention vectors for feature aggregation. The overall mechanism computes a vector-valued weighted sum and allows each channel in y_i to selectively aggregate information from its neighbors as shown in Figure 2b. In the proposed feature disentanglement module, each point feature is re-calibrated independently based solely on its own representation—essentially performing a self-modulation akin to squeeze and excitation [33], without explicitly aggregating information from neighboring points. This process is illustrated in Figure 2c.



(**a**) In squeeze and excitation, all the points are compressed spatially using global spatial pooling. The resulting vector is used to compute an attention vector applied channel-wise to all the points indiscriminately.



(**b**) In vector self-attention, the neighbors of the two points are used to compute per-point attention vectors that are then applied channel-wise.



(**c**) The feature vectors of each point are used to compute per-point attention vectors that are then applied channel-wise.

Figure 2. Illustration of how the features of two points (highlighted in yellow) are calculated using the following: (**a**) The squeeze-and-excitation attention module. (**b**) Vector self-attention. (**c**) Our proposed attention mechanism for object and context feature refinement.

3.2.2. Squeeze-and-Excitation Module

The squeeze-and-excitation module [33] performs global average pooling to squeeze the spatial dimension. The resulting vector is then fed through an MLP and a sigmoid function to produce channel-wise weights, which are applied for all spatial positions (Figure 2a). In our proposed source context feature refinement module, the spatial dimension is squeezed but further used to compute point-wise attention vectors for each target domain position.

3.3. Training Objective

Our method enhances existing UDA techniques by addressing NT and can integrate with various UDA methods to improve performance. The overall training objective (Equation (13)) includes the cross-entropy loss on source data (Equation (12)) and a UDA-specific loss $\mathcal{L}^{\text{UDA}}$, which depends on the chosen UDA method (e.g., adversarial loss, self-training loss, or other domain alignment objectives):

$$\mathcal{L}_s^{ce} = -\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{c=1}^{C} \mathbb{1}_{[\hat{y}_i^{(c)}=c]} \log\left[h(F_o^{s\to t})_i^c\right], \tag{12}$$

$$\mathcal{L} = \mathcal{L}_s^{ce} + \mathcal{L}^{\text{UDA}}, \tag{13}$$

h is the classifier computed on the refined source features.

4. Results

4.1. Datasets and Baselines

4.1.1. Datasets

We evaluate our method in a synthetic-to-real UDA setting, training on synthetic data and adapting to real-world datasets. We align our validation split and training setup with previous works [2,3]. The SynLiDAR dataset [2], created with Unreal Engine and a simulated 64-beam LiDAR, serves as the source domain. We use SemanticPOSS [34] and SemanticKITTI [35] for the target domains. The labels are aligned into common sets: 13 classes for SynLiDAR to SemanticPOSS and 19 for SynLiDAR to SemanticKITTI [2,3]. Segmentation performance is measured using mean Intersection over Union (mIoU) [36].

4.1.2. Baselines and Training

We compare our method against state-of-the-art 3D UDA methods and NT mitigation techniques. For 3D UDA, we use PCAN [4], a density-guided adversarial method, and CoSMix [3], a self-training approach with cross-domain semantic mixing. To assess NT mitigation, we use DSAN [19], CWFT [29], TransPar [26], and BSS [24] with each UDA method. We use MinkUNet32 [37] as the backbone for point cloud semantic segmentation. The input voxel size is set to 0.05 m, and the implementation is in PyTorch (v22.04) using a single NVIDIA Tesla V100 GPU (Santa Clara, CA, USA). We pretrain on the source domain for ten epochs using SGD (learning rate 0.01, momentum 0.9, and batch size 4), and this pretrained model serves as the baseline. We initialize with pretrained weights during adaptation and follow each baseline's original implementation and hyperparameters for a fair comparison. Our method sets the context abstraction level to $g = 1$ (first network layer), and object-specific features o are from the last layer before classification. All the attention modules use 1×1 sparse convolution layers. We provide below more details about the hyperparameters used for adaptation and training all the negative transfer methods for both PCAN [4] (Table 1) and CosMix [3] (Table 2).

Table 1. Main hyperparameters for PCAN training.

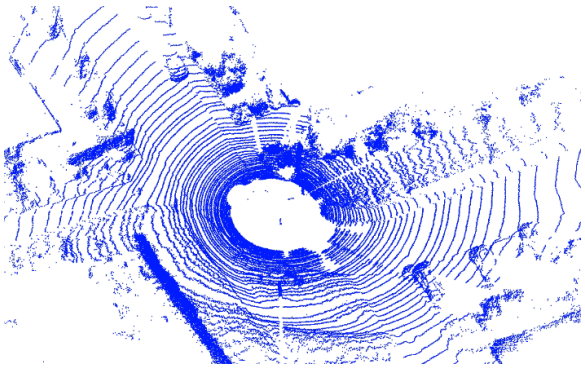
Hyperparameter	SemanticKITTI	SemanticPOSS
Maximum Epochs	100,000	100,000
Entropy Threshold	0.05	0.05
Adversarial Loss Weight	0.001	0.001
Mean Teacher α	0.9999	0.9999
Voxel Size	0.05	0.05
Learning Rate (Generator)	2.5×10^{-5}	2.5×10^{-4}
Learning Rate (Discriminator)	1×10^{-5}	1×10^{-4}

Table 2. Main hyperparameters for CoSMix.

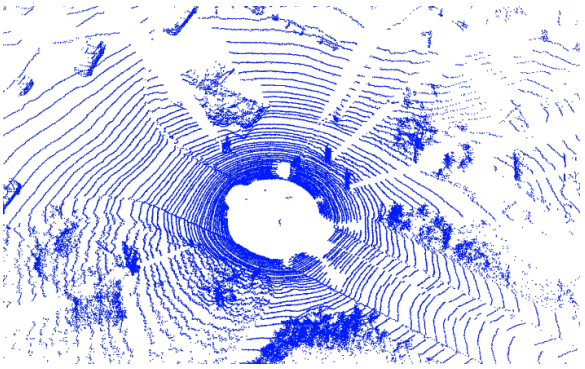
Hyperparameter	SemanticKITTI	SemanticPOSS
Voxel Size	0.05	0.05
Number of Points	80,000	50,000
Epochs	20	20
Train Batch Size	1	1
Optimizer	SGD	SGD
Learning Rate	0.001	0.001
Selection Percentage	0.5	0.5
Target Confidence Threshold	0.90	0.85
Mean Teacher α	0.9	0.99
Teacher update frequency	500	500

4.2. Feature Disentanglement Results

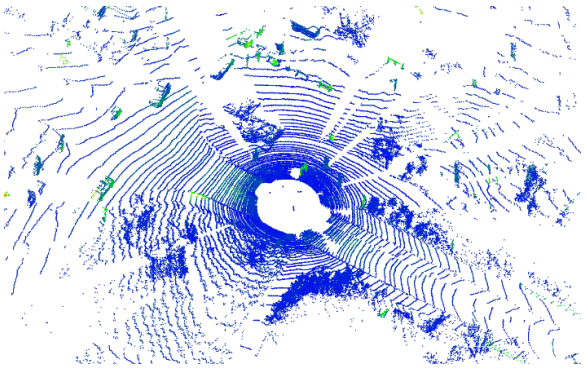
The object and context attention maps generated by our CAFA method are illustrated in Figure 3. As described in Section 3.1.1, these attention maps are computed at different abstraction levels within the network and confirm that the model learns a meaningful separation of object and context features. In Figure 3, object attention maps indicate points that the model deems most important for discriminating objects, such as parts of cars and buildings and urban objects on sidewalks. On the other hand, context attention maps highlight points that provide contextual cues, such as the surroundings that help the model interpret the object in its setting (e.g., ground, cars, buildings, or other spatial context).



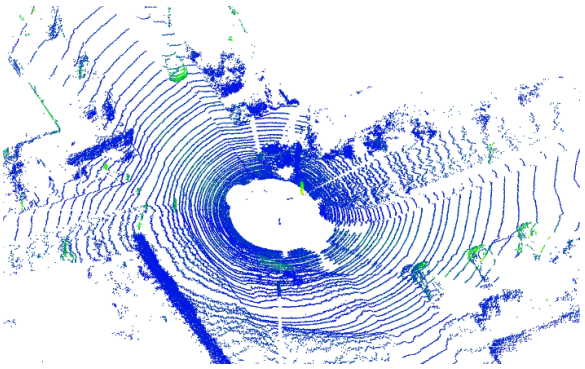
(**a**) Example 1: Input Point Cloud



(**b**) Example 2: Input Point Cloud

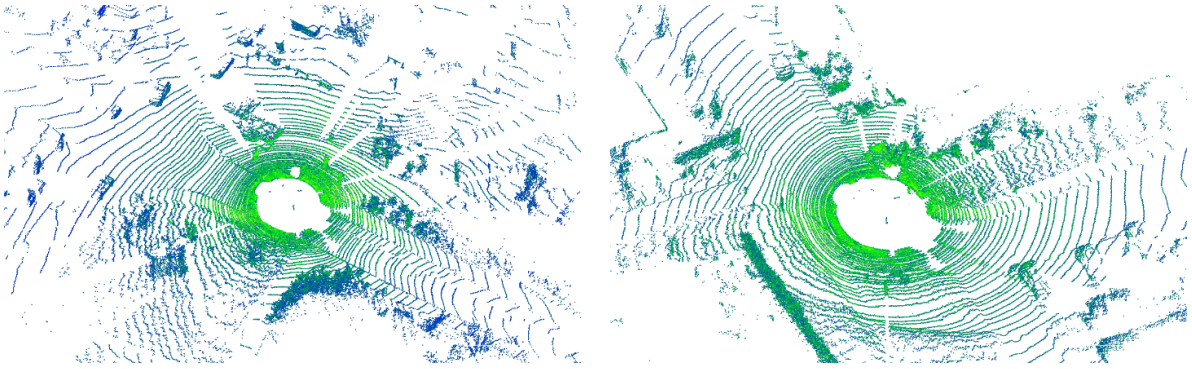


(**c**) Example 1: Object Attention Map



(**d**) Example 2: Object Attention Map

Figure 3. *Cont.*



(**e**) Example 1: Context Attention Map (**f**) Example 2: Context Attention Map

Figure 3. Object and context attention maps for two example point clouds. (**a**,**b**) Input point clouds. (**c**,**d**) Object attention maps highlighting object-specific features. Points relevant to objects are highlighted in green. (**e**,**f**) Context attention maps emphasizing contextual information. Points important for scene context are highlighted in green.

4.3. Comparison with Previous Methods

We evaluate CAFA on two challenging UDA scenarios: SynLiDAR → SemanticKITTI (Table 3) and SynLiDAR → SemanticPOSS (Table 4). We report the results in terms of the mean Intersection over the Union (mIoU).

Overall Performance: CAFA consistently outperforms baseline UDA methods in both adaptation scenarios. In SynLiDAR → SemanticKITTI, it improves CoSMix by 1.6 mIoU to 34.5%, with notable gains in "motorcycle" (+7.6%), "bicyclist" (+7.9%), and "terrain" (+6.7%) but slight decreases in "car" (−0.4%) and "building" (−2.5%). Applied to PCAN, CAFA improves the mIoU by 0.4 points, with gains in "motorcycle" (+4.3%) and "bicyclist" (+7.5%) but drops in "traffic sign" (−5.6%). In SynLiDAR to SemanticPOSS (Table 4), CAFA shows consistent improvements. Integrated with CoSMix, it gains 1.0 mIoU to reach 45.0%, improving "car" (+9.3%) and "pole" (+1.4%) but decreases in "person" (−3.0%) and "traffic" (−2.7%). Applied to PCAN, CAFA adds 0.9 mIoU, with gains in "rider" (+1.7%) and "building" (+1.2%) but drops in "traffic" (−0.3%) and "bike" (−1.6%).

Table 3. Adaptation results on SynLiDAR → SemanticKITTI. The source corresponds to the model trained on the source dataset. The results are reported in terms of mean Intersection over Union (mIoU).

Model	car	Bike	Mot	Truck	Other-v	Perso	Bcyst	Mclst	Road	PARK	sidew	Other-g	Build	Fence	Vege	Trunk	Terra	Pole	Traff	mIoU	Gain
Source	72.6	6.7	11.6	3.7	6.7	22.5	29.5	2.8	67.2	11.9	35.7	0.1	59.9	23.5	74.3	25.7	42.0	39.6	13.3	28.9	-
CoSMix [3]	83.6	10.2	14.4	**8.8**	**15.2**	27.4	23.5	0.7	77.4	17.4	43.6	**0.3**	55.1	24.5	72.7	**44.8**	40.8	**45.3**	19.8	32.9	+0.0
CoSMix + CWFT [29]	**85.1**	**11.2**	14.3	4.0	11.9	28.0	16.5	3.9	76.4	16.9	44.9	0.1	52.1	25.2	72.4	42.2	42.8	44.2	19.6	32.2	−0.7
CoSMix + DSAN [19]	84.6	1.8	15.0	5.1	10.7	**33.5**	26.9	2.7	76.6	18.8	44.2	0.2	**61.0**	**30.8**	**74.3**	41.8	39.6	42.9	**24.8**	33.4	+0.5
CoSMix + TransPar [26]	83.8	5.4	13.1	7.1	13.0	22.5	15.4	2.5	77.2	17.3	43.4	0.1	58.6	24.9	73.4	41.6	38.5	43.5	15.8	31.4	−1.5
CoSMix + BSS [24]	82.6	5.3	17.4	7.8	11.7	26.7	12.7	3.5	77.5	**19.9**	44.6	0.2	52.5	25.9	72.1	38.9	41.1	43.4	20.1	31.8	−1.2
CoSMix + CAFA (ours)	83.2	9.4	**22.1**	7.9	13.4	**33.2**	**31.4**	**5.7**	**78.0**	15.9	**46.5**	0.1	52.6	28.5	72.5	43.0	**47.5**	45.0	19.9	**34.5**	**+1.6**
PCAN [4]	85.6	16.2	27.4	9.9	10.4	28.4	64.2	2.9	77.1	13.9	50.3	**0.1**	67.4	19.4	75.9	41.4	47.7	40.8	21.7	36.9	+0.0
PCAN + CWFT [29]	86.6	17.0	25.7	10.9	10.1	30.6	60.1	2.7	**77.4**	12.8	50.1	0.1	64.9	**23.3**	74.7	**43.6**	46.5	**42.7**	22.8	37.0	+0.1
PCAN + DSan [19]	85.8	16.9	27.7	9.9	10.3	28.7	65.3	2.8	77.0	**14.2**	**50.4**	0.1	69.0	19.9	76.4	42.1	46.6	41.4	22.1	37.2	+0.3
PCAN + TransPar [26]	**87.3**	**17.6**	28.9	**11.5**	**12.7**	30.8	65.1	2.4	76.5	12.9	48.9	0.1	**69.6**	19.0	**77.2**	40.7	44.4	40.7	22.3	37.3	+0.4
PCAN + BSS [24]	86.0	17.0	28.1	10.5	11.6	26.9	65.9	3.3	77.2	13.9	50.3	0.1	68.1	19.4	76.2	41.2	47.4	40.7	**22.9**	37.2	+0.3
PCAN + CAFA (ours)	85.2	13.3	**31.7**	**11.2**	12.4	**33.3**	**71.7**	**3.7**	77.0	11.0	49.9	0.0	66.9	18.0	75.7	42.4	**50.5**	37.8	16.1	**37.3**	**+0.4**

Table 4. Adaptation results on SynLiDAR → SemanticPOSS. The source corresponds to the model trained on the source dataset. The results are reported in terms of mean Intersection over Union (mIoU).

Model	Person	Rider	Car	Trunk	Plants	Traffic	Pole	Garbage	Building	Cone	Fence	Bike	Grou.	mIoU	Gain
Source	45.7	40.2	51.5	22.1	71.9	4.9	22.2	21.9	71.9	4.8	29.8	2.6	76.1	35.8	-
CoSMix [3]	55.3	52.4	47.6	**43.5**	72.0	13.7	40.9	35.4	67.7	30.2	35.3	5.6	81.3	44.0	+0.0
CoSMix + CWFT [29]	53.9	50.7	54.0	31.2	72.6	13.2	41.8	35.7	69.9	28.4	31.6	**7.1**	81.3	43.9	−0.1
CoSMix + DSAN [19]	52.8	51.5	51.6	35.9	70.8	13.3	38.2	**36.9**	62.7	**31.8**	33.0	4.8	79.1	43.3	−0.7
CoSMix + Transpar [26]	54.6	**54.1**	53.2	35.4	**74.1**	13.7	40.7	31.5	**72.8**	24.4	32.9	6.3	81.1	44.2	+0.2
CoSMix + BSS [24]	**55.6**	52.7	48.0	35.2	73.3	**15.5**	40.1	28.4	70.8	29.6	**38.4**	6.2	**81.4**	44.3	+0.3
CoSMix + CAFA (ours)	52.3	53.9	**56.9**	34.0	72.5	11.0	**42.3**	**36.9**	70.6	**31.7**	36.6	5.2	81.1	**45.0**	**+1.0**
PCAN [4]	60.9	52.3	60.1	41.2	74.5	18.0	35.0	23.9	74.8	8.0	38.7	12.3	79.3	44.6	+0.0
PCAN + CWFT [29]	59.7	51.5	59.1	40.9	74.0	17.6	35.0	24.8	74.4	**8.7**	40.2	11.3	79.2	44.3	−0.3
PCAN + DSAN [19]	62.3	50.3	60.7	41.3	**74.9**	13.3	**36.9**	21.4	75.4	1.9	42.7	9.9	77.6	43.7	−0.9
PCAN + TransPar [26]	64.0	**55.2**	60.4	**42.8**	74.5	16.3	36.1	19.9	74.3	4.5	40.9	**15.4**	**79.9**	44.9	+0.3
PCAN + BSS [24]	62.2	52.5	60.5	38.7	74.6	**20.0**	35.6	18.3	**77.1**	4.2	44.4	14.1	79.8	44.8	+0.2
PCAN + CAFA (ours)	**64.4**	54.0	**63.9**	40.9	74.1	17.7	36.0	**25.2**	76.0	3.2	**45.5**	10.7	79.7	**45.5**	**+0.9**

Comparison with Negative Transfer Mitigation Methods: CAFA outperforms all other mitigation methods when applied to CoSMix on SynLiDAR → SemanticKITTI, achieving 34.5% mIoU versus 33.4% for the next best (DSAN). Some techniques (CWFT, GCR, TransPar, and BSS) reduce CoSMix's performance, highlighting challenges in mitigating negative transfer in 3D segmentation. For PCAN on SemanticKITTI, the improvements are smaller; CAFA and TransPar achieve the highest mIoU (37.3%), suggesting PCAN may be more robust, but CAFA still provides benefits. In SynLiDAR to SemanticPOSS, CAFA again leads when applied to CoSMix, achieving 45.0% mIoU compared to 44.3% for BSS. Applied to PCAN, CAFA reaches the highest mIoU (45.5%), with TransPar close at 44.9%. These results demonstrate that CAFA consistently outperforms other negative transfer mitigation techniques across UDA methods and datasets.

Training Time Overhead Evaluation: To evaluate the runtime overhead of CAFA, we measured the per epoch training time on the SynLiDAR → SemanticKITTI task using CoSMix. The base CoSMix model required 4.87 h per epoch, while CoSMix with CAFA took 4.88 h, which an increase of ∼0.2%. This amounts to an hour of extra training time over 100 epochs of training. This overhead confirms that CAFA's added modules (disentanglement, cross-attention, and fusion) are practical for large-scale 3D point cloud segmentation.

4.4. Qualitative Analysis

We further illustrate how CAFA leverages target context by comparing segmentation outputs with baseline approaches. Figures 4 and 5 illustrate qualitative segmentation results for the SynLiDAR → SemanticPOSS and SynLiDAR → SemanticKITTI adaptations, respectively, comparing ground truth labels with predictions from CoSMix and our CAFA method.

In Figure 4, classes such as `buildings` and `fence` show improved performance. Confusion arises in these classes due to contextual similarities with other semantic categories. For instance, `buildings` are frequently misclassified as `vegetation`, and `fences` are often mislabeled as `vegetation` or `cars`. This is due to the higher co-occurrence of `buildings` with `vegetation` and `vegetation` with tree `trunks` in SemanticPOSS. Consequently, a model might learn a misleading association linking `vegetation` to `trunks`, and at test time, it may mislabel facade points as `vegetation` if `trunk` points are nearby. Cars co-occur with `vegetation` and `ground` often in SynLiDAR. When a context shift makes the model wrongly spot `vegetation` around a `fence` in SemanticPOSS, that same association can cause fences to be mislabeled as cars. After CAFA, `car`, which is one of the classes showing the most context shift in SemanticPOSS, shows less confusion with `fences`.

In Figure 5, CAFA effectively distinguishes between the classes `terrain`, `sidewalk`, and `other-ground`, despite their frequent co-occurrence in SynLiDAR. Specifically, the high proximity of `other-ground` to roads in SynLiDAR creates a spurious correlation, prompting simultaneous predictions for these classes. After applying CAFA, the erroneous prediction of `other-ground` is significantly reduced, which reflects its weaker association with the other surfaces in SemanticKITTI. Figure 5i demonstrates CAFA's capability to distinguish `trunk` from `person`, a class affected by context shift. In SynLiDAR, `person` frequently co-occurs with sidewalks, creating a strong correlation that causes confusion when sidewalks are predicted nearby. CAFA mitigates this issue, resolving such misclassifications even when sidewalks are still predicted in its neighborhood.

Figure 6b illustrates the effectiveness of CAFA in handling context shifts. CAFA correctly identifies the sidewalk beneath parked cars, a common scenario in the target domain but not the source, while CoSMix misclassifies it as road. To understand this improvement, we visualize the points contributing most to the class prediction using GradCAM [38]. CoSMix's attention map (Figure 6e) shows that when predicting road, it focuses on a wide range of elements, including the cars themselves and buildings. In contrast, CAFA's attention map (Figure 6d) demonstrates a more refined focus. When classifying the same area, CAFA concentrates on the cars and nearby urban objects commonly found on sidewalks. This targeted attention indicates that CAFA has learned to associate these specific elements with sidewalks in the target domain.

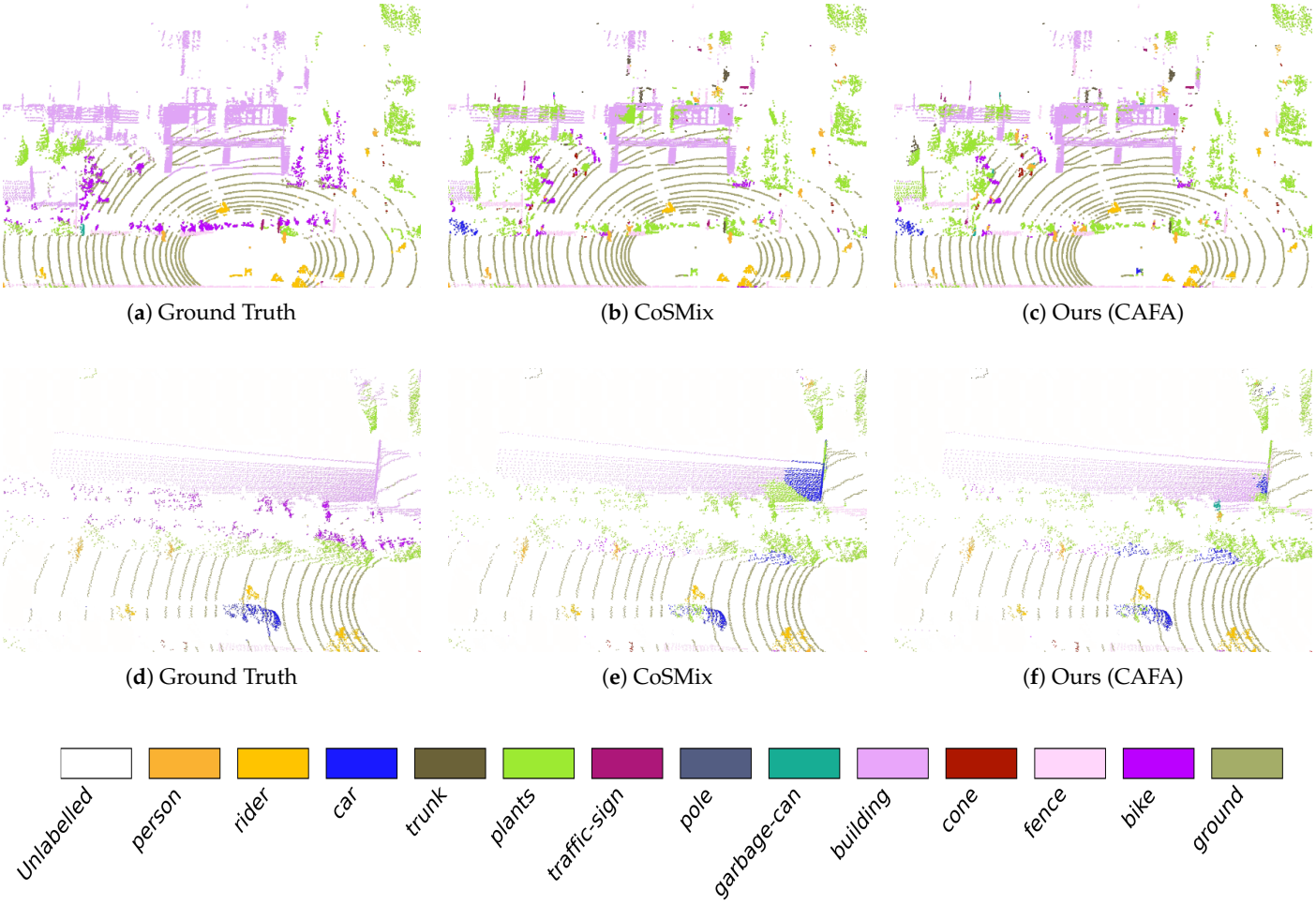
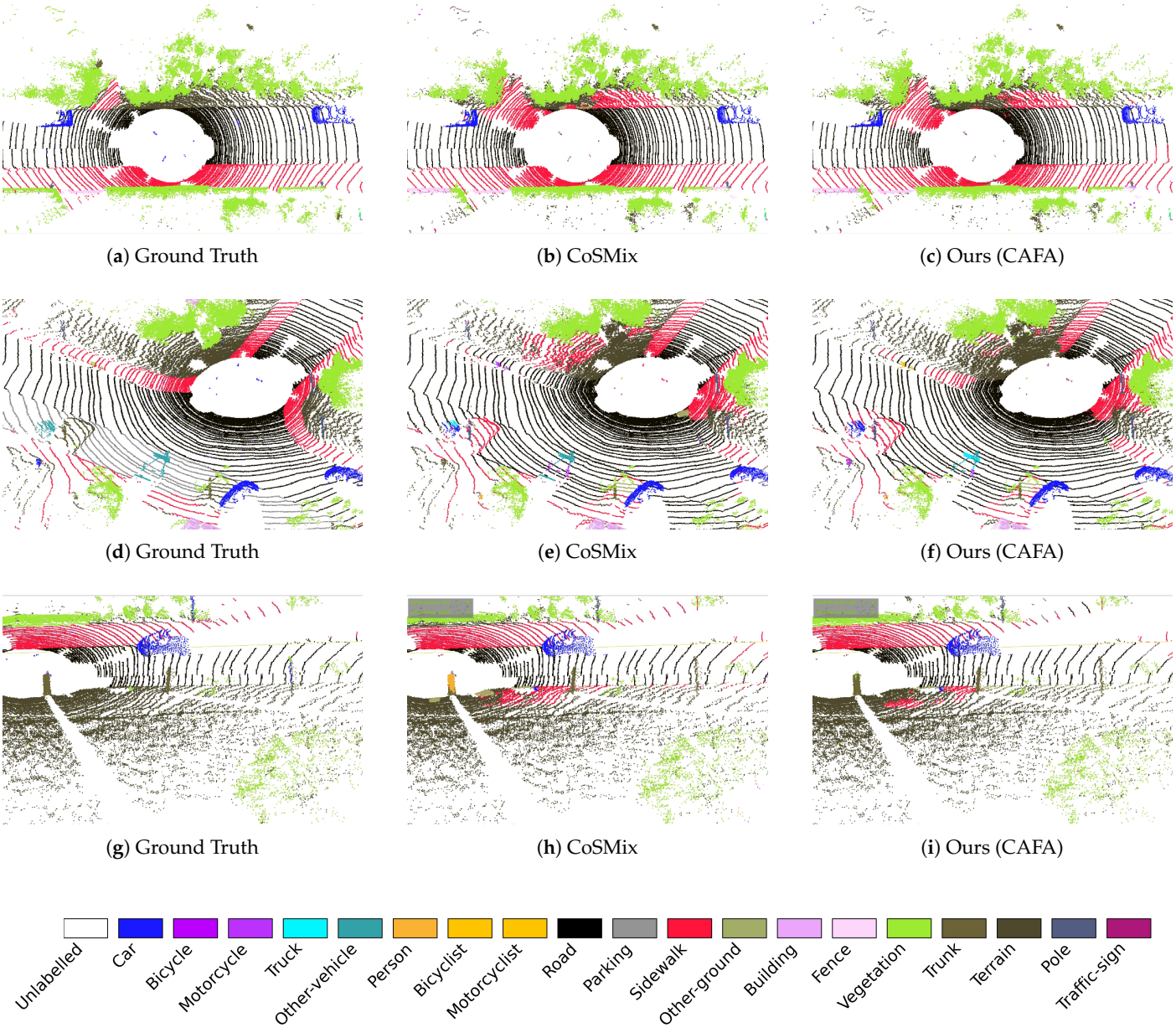


Figure 4. Segmentation comparison on SynLiDAR → SemanticPOSS. Two example scenes are shown, comparing ground truth, CoSMix, and our CAFA method.

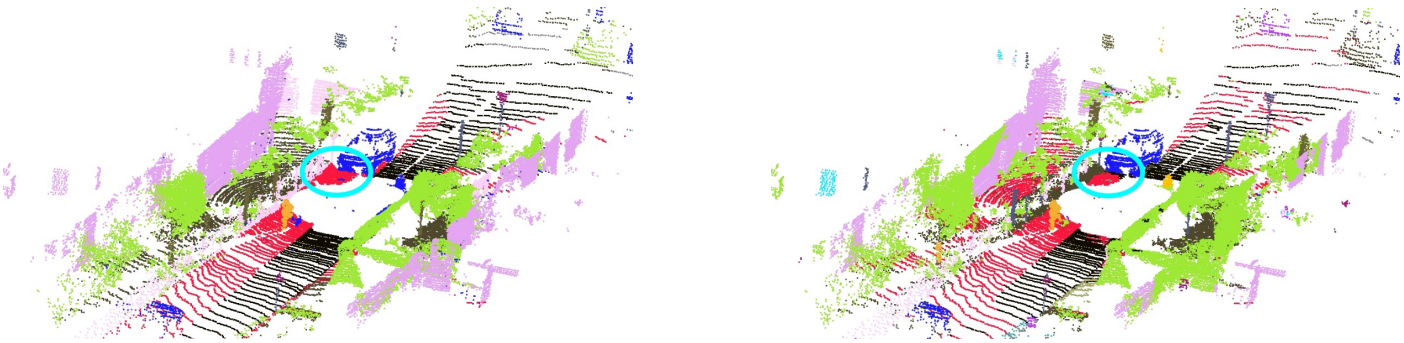


(**a**) Ground Truth (**b**) CoSMix (**c**) Ours (CAFA)

(**d**) Ground Truth (**e**) CoSMix (**f**) Ours (CAFA)

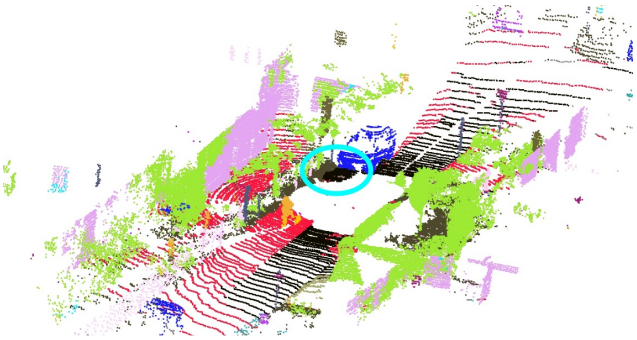
(**g**) Ground Truth (**h**) CoSMix (**i**) Ours (CAFA)

Figure 5. Segmentation comparison on SynLiDAR → SemanticKITTI. Three example scenes are shown, comparing ground truth, CoSMix, and our CAFA method.

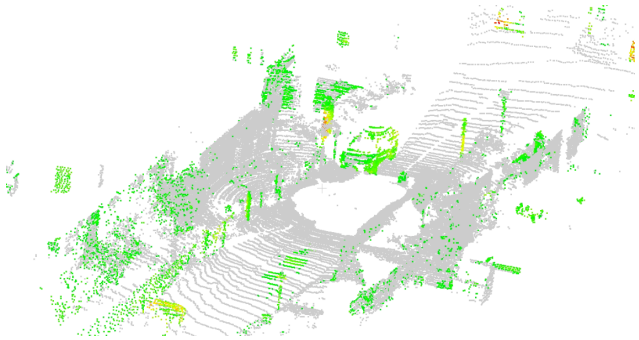


(**a**) Ground Truth Segmentation (**b**) CAFA's Correct Sidewalk Prediction

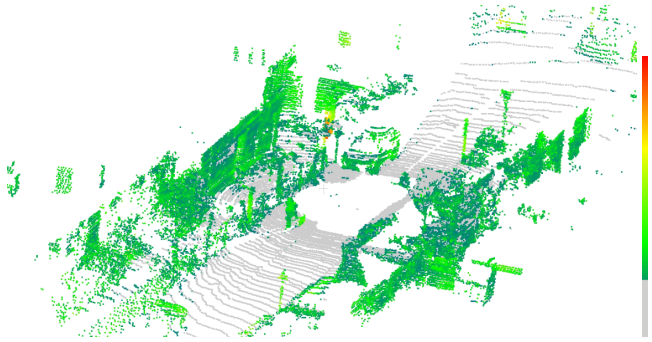
Figure 6. *Cont.*



(**c**) CoSMix's Misclassification as Road



(**d**) CAFA Attention Map for Sidewalk Prediction



(**e**) CoSMix Attention Map for Road Prediction



Figure 6. Analysis of our approach in mitigating negative transfer for sidewalk segmentation.

4.5. Ablation Study

To evaluate each component of our design, we conducted ablation studies aimed at clarifying the impact of our specific architectural choices on SynLiDAR → SemanticPOSS. Specifically, we explored five key modifications:

Single-scale vs. multi-scale features: We investigated whether using single-scale features for both object and context representations is sufficient, compared to our original multi-scale design.

Impact of different context representation: To determine whether performance gains result primarily from integrating additional contextual information, we compared our approach to a baseline employing single-domain self-attention for both source and target. We also benchmarked our context modeling against the Global Context Reasoning (GCR) module proposed by Ma et al. [39], which leverages channel similarity to construct graph nodes that are processed by Graph Convolutional Networks (GCNs).

Use of object features alone: To isolate the contribution of context alignment, we assessed performance when classification relied solely on object features.

Fusion strategy evaluation: Finally, we evaluated the effectiveness of our fusion approach by comparing it against a simpler alternative using direct summation of object and context representations. We also analyze the impact of using a residual connection for gradual adaptation.

We summarize the results of our ablation studies in Table 5.

Multi-scale Feature Disentanglement: Using single-scale features results in a 0.9 mIoU drop, confirming that multi-scale features capture richer contextual and object-specific information.

Cross-attention: Cross-domain context transfer outperforms single-domain context modeling. Self-attention on source and target features decreases the mIoU by 0.8% and 1.3%, respectively, and the GCR module [39] also reduces performance, emphasizing the necessity of cross-attention to address context shift.

Feature Fusion: Replacing adaptive fusion with simple summation reduces performance by 0.7 mIoU.

Context Features: Using only object features for classification decreases performance, which highlights the value of integrating both context and object-specific features.

Residual Connection: Removing the residual connection in feature fusion caused a −2.8 mIoU drop, showing its role in preserving original representations while gradually introducing target context features.

Table 5. Ablation study results on SynLiDAR → SemanticPOSS.

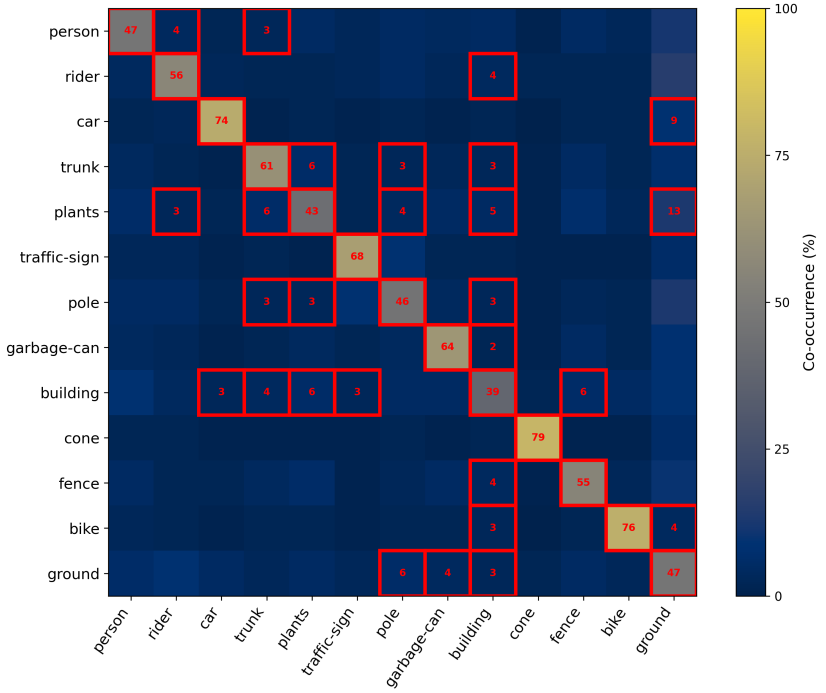
Method	mIoU	ΔmIoU
CAFA (full)	45.0	0.0
CAFA w/o fusion (simple summation)	44.3	−0.7
CAFA w/ single-scale features	44.1	−0.9
CAFA w/ object features only	44.6	−0.4
CAFA w/o residual connection	42.4	−2.8
CAFA w/ self-attention (source)	44.2	−0.8
CAFA w/ self-attention (target)	43.7	−1.3
CAFA w/ 3D context module [39]	43.2	−1.8

4.6. Context Shift Performance Analysis

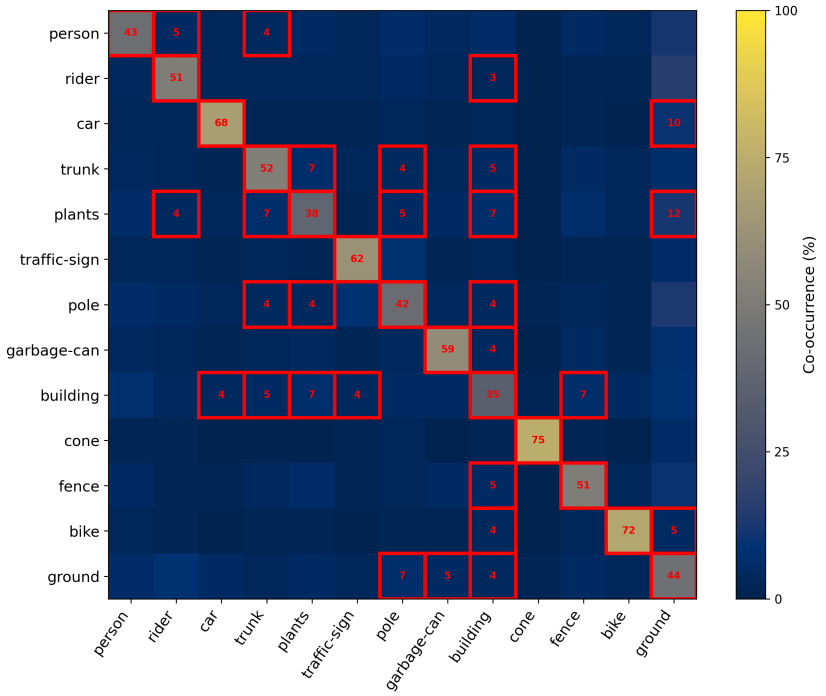
To evaluate CAFA's efficacy in addressing context shift, we analyzed class-wise performance improvements relative to context shift, quantified by the L1 distance between class co-occurrence matrices of the source and target domains. Using balanced sampling ($N = 5000$ points per cloud), we computed neighborhood class statistics to create normalized co-occurrence matrices. We sample points from each domain and look at their neighborhood to determine how often each class co-occurs with every other class. The frequency of these pairwise co-occurrences is then normalized and arranged into a matrix. A high value in row_i and $column_j$ means $class_i$ appears frequently near $class_j$. Figures 7 and 8 show the differences between the source and target domains. For instance, SynLiDAR shows weaker co-occurrence between `vehicles` and `drivable surfaces` and classes such as `car` or `person` have lower co-occurrence with environmental classes like `building`, `vegetation`, or `sidewalk` compared to SemanticKITTI (Figure 8b). In SemanticPOSS, `cars` appear alongside environmental labels more often than in SynLiDAR (Figure 7b).

The L1 distance provides a measure of context shift for each class, defined as the absolute difference between its class occurrence distributions in the source and target domains. Specifically, it corresponds to the sum of absolute differences between the rows representing class occurrences in the co-occurrence matrices, where each row characterizes a specific class's occurrence with other classes. We plot context shift against performance improvement for CoSMix and PCAN on SemanticKITTI and SemanticPOSS (Figure 9). The results show a positive correlation: classes with more significant shifts exhibit greater performance gains, as indicated by CAFA's trend line (green), which consistently slopes positively. In contrast, the best-performing NT method (purple) occasionally exhibits negative slopes, highlighting CAFA's ability to mitigate NT for high-shift classes. To validate further, we computed the mIoU for the top three high-shift classes in PCAN and

CoSMix across the UDA tasks (Table 6). CAFA consistently outperforms other methods, achieving a 5.8 mIoU improvement on average compared to 1.4 for the best NT approach.

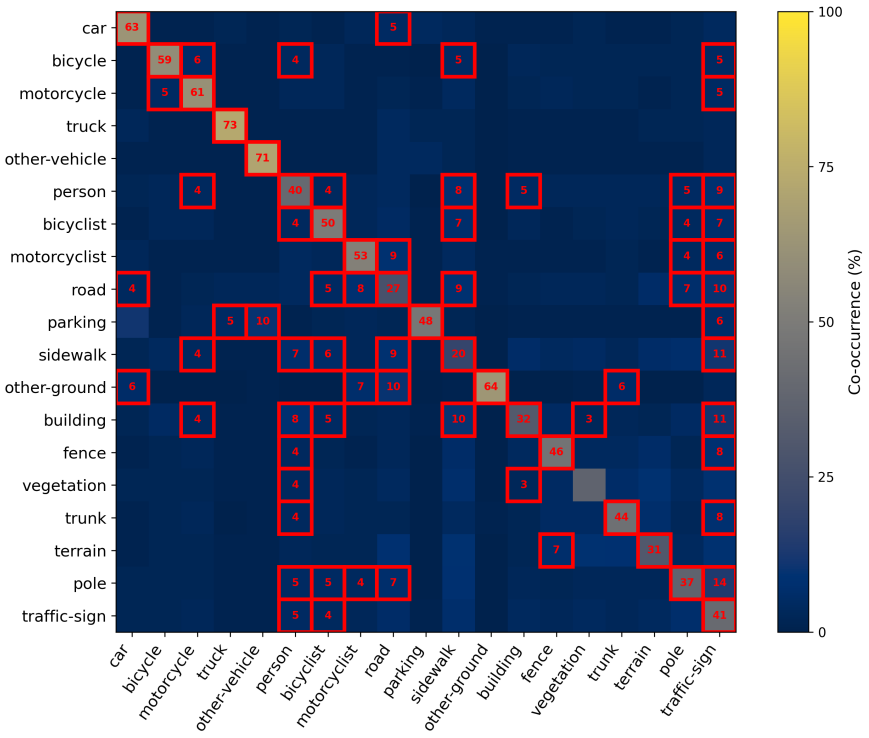


(**a**) SynLiDAR Co-occurrence Matrix

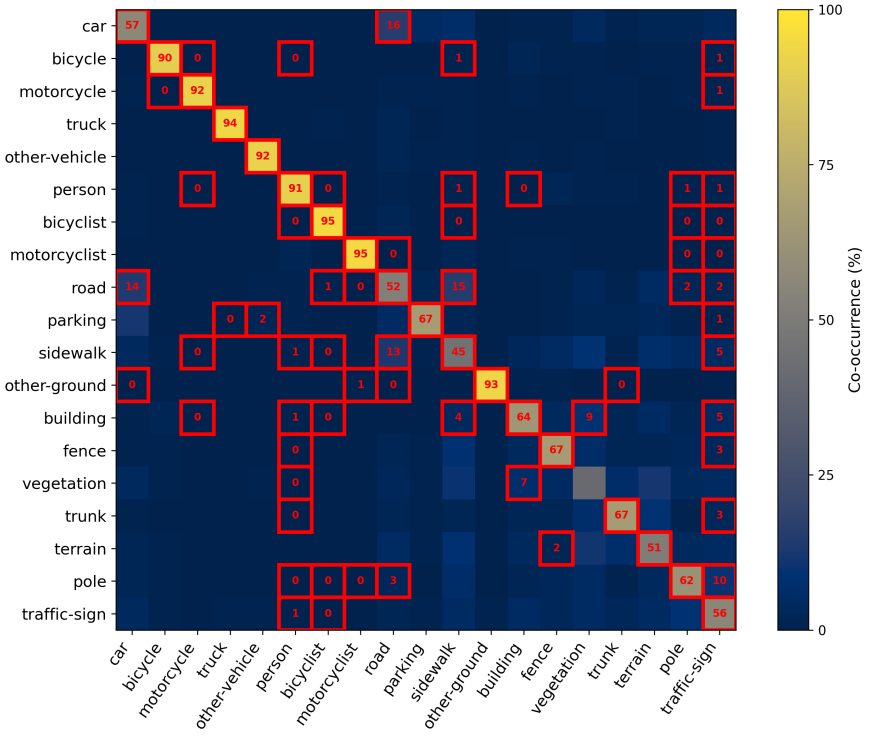


(**b**) SemanticPOSS Co-occurrence Matrix

Figure 7. Co-occurrence matrices for source (SynLiDAR) and target (SemanticPOSS) domains. Red boxes mark the entries with the largest inter-domain co-occurrence differences; for clarity, only these top difference entries are highlighted. (**a**) SynLiDAR co-occurrence matrix. (**b**) SemanticPOSS co-occurrence matrix. Color intensity represents the frequency of co-occurrence between classes.

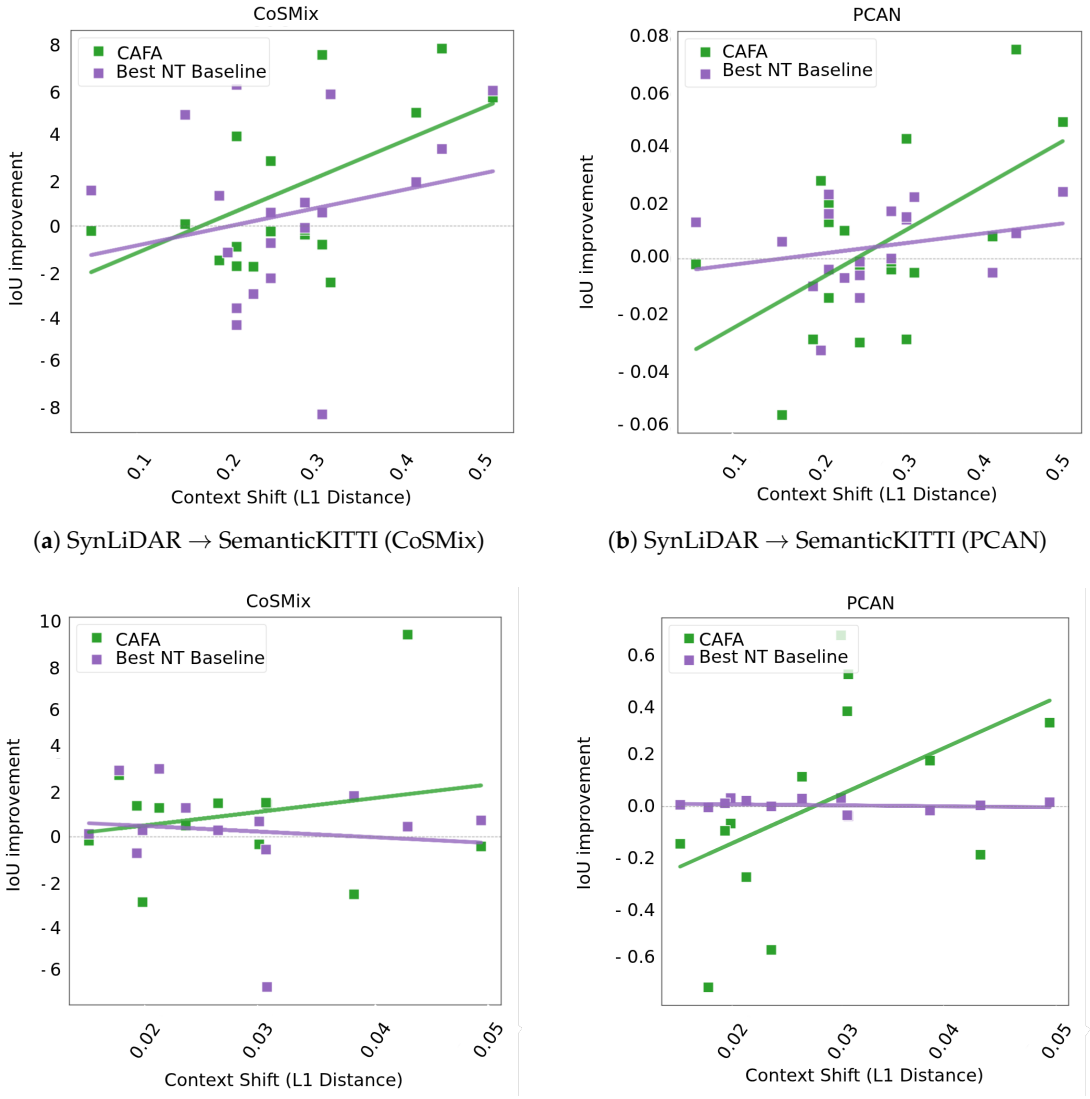


(**a**) SynLiDAR Co-occurrence Matrix



(**b**) SemanticKITTI Co-occurrence Matrix

Figure 8. Co-occurrence matrices for source (SynLiDAR) and target (SemanticKITTI) domains. Red boxes mark the entries with the largest inter-domain co-occurrence differences. (**a**) SynLiDAR co-occurrence matrix. (**b**) SemanticKITTI co-occurrence matrix. Color intensity represents the frequency of co-occurrence between classes.



(**a**) SynLiDAR → SemanticKITTI (CoSMix)

(**b**) SynLiDAR → SemanticKITTI (PCAN)

(**c**) SynLiDAR → SemanticPOSS (CoSMix)

(**d**) SynLiDAR → SemanticPOSS (PCAN)

Figure 9. Class-wise performance gain (IoU difference) against context shift magnitude. X-axis: per class L1 distance between source/target co-occurrence matrices. Y-axis: IoU improvement. Each point represents a class; trend lines show correlation between shift and performance gain.

To further illustrate how context shift affects adaptation, we highlight in Table 6 the top three classes in each scenario with the largest co-occurrence mismatch, i.e., the greatest context shift, across both PCAN and CoSMix on the SynLiDAR → SemanticKITTI and SynLiDAR → SemanticPOSS adaptations. Compared to both the baseline UDA methods (PCAN and CoSMix) and other negative transfer mitigation approaches, CAFA consistently yields larger mIoU gains for these high-shift classes. For SynLiDAR → SemanticKITTI, CAFA achieves a gain of +4.4% in mIoU compared to +0.9% for the best-performing NT method with PCAN. Similarly, for CoSMix, CAFA achieves a gain of +6.2% compared to +3.8% for NT. For SynLiDAR → SemanticPOSS, CAFA achieves a gain of +10.6% in mIoU compared to +0.1% for the best-performing NT method with PCAN. For CoSMix, CAFA achieves a gain of +2.0% compared to +1.0% for NT.

This analysis illustrates that targeting negative transfer due to context shift is an effective way to mitigate performance drops that occur when knowledge from the source domain is not directly applicable in the target domain.

Table 6. mIoU (%) for top 3 classes with greatest context shift for SynLiDAR→ SemanticKITTI and SynLiDAR → SemanticPOSS.

Dataset	Class	PCAN			CoSMix		
		Baseline	**+NT**	**+CAFA**	**Baseline**	**+NT**	**+CAFA**
SynLiDAR → SemanticKITTI	person	28.4	30.8	33.3	27.4	33.47	33.18
	bicyclist	64.2	65.1	71.7	23.4	26.89	31.36
	motorcyclist	2.9	2.4	3.7	0.7	2.66	5.74
	mIoU	**31.8**	**32.7**	**36.2**	**17.2**	**21.0**	**23.4**
SynLiDAR → SemanticPOSS	Trunk	41.2	42.8	74.1	34.48	35.22	34.03
	Car	60.1	60.4	40.9	47.57	48.02	56.94
	Traffic-sign	18.0	16.3	36.0	13.65	15.54	10.96
	mIoU	**39.7**	**39.8**	**50.3**	**31.9**	**32.9**	**33.9**

5. Discussion

The first component of CAFA is its ability to disentangle object-specific and context-specific features. This separation is achieved through attention mechanisms at different abstraction levels as opposed to rigid approaches that enforce strict orthogonality between features. This soft disentanglement preserves useful shared information and allows for the selective adaptation of source context features using the target domain. It also prevents the transfer of misleading contextual biases and is supported by the performance improvements observed in our ablation studies and context shift analysis. To mitigate context shift, it is crucial to have a method to quantify it and evaluate performance improvements relative to context variability. Our quantitative analysis using normalized class co-occurrence matrices to measure context shift shows a positive correlation between the magnitude of context shift and segmentation performance gains. For example, as shown in Figure 9 and Table 6, classes with high context shifts exhibit significant improvements in mIoU when CAFA is applied. These results confirm our working hypothesis: when the degree of context mismatch between source and target domains is large, incorporating target domain context into the adaptation process is particularly beneficial. Our findings indicate that CAFA outperforms other mitigation strategies (e.g., DSAN, CWFT, TransPar, and BSS) by delivering higher mIoU gains, particularly for classes with significant context mismatches.

From a practical standpoint, mitigating negative transfer in applications such as autonomous driving, urban mapping, and robotics is essential for developing robust real-world systems. This enables segmentation models to generalize better across varied urban environments—where the spatial relationships between objects may differ significantly between the training and deployment domains. The demonstrated positive correlation between context shift and performance improvement provides a quantitative basis for future evaluation metrics and shows the importance of integrating context in UDA for semantic segmentation.

6. Conclusions

Our proposed method addresses negative transfer in unsupervised domain adaptation for 3D LiDAR semantic segmentation. Disentangling object-specific and context-specific features and refining source context features using target domain information effectively mitigates the impact of context shift between domains. Our domain shift analysis, quantifying the relationship between class-wise performance improvements and the degree of shift for each class, provides strong evidence for its effectiveness. Furthermore, experimental results on challenging synthetic-to-real adaptation scenarios consistently show perfor-

mance improvements over state-of-the-art UDA methods and existing negative transfer mitigation techniques.

However, our approach has some limitations. The proposed feature disentanglement relies on a simple attention mechanism, which may only partially model complex contextual relationships in some scenarios and does not explicitly address sensor noise. Incorporating explicit noise modeling or denoising mechanisms could further enhance CAFA's effectiveness, especially in challenging real-world scenarios. Furthermore, our method adopts a soft attention-based disentanglement strategy rather than strict orthogonality and thus retains some overlap between object and context features. Exploring quantitative measures such as mutual information or representation similarity could provide deeper insight into the nature of the learned feature separation.

Our analysis of context shift relies on fixed sampling parameters (number of points per cloud and a fixed neighborhood radius), which can influence the granularity of the co-occurrence matrices. Although we observe that the positive correlation between context shift and performance improvement remains consistent across datasets with different point densities and sensors, a systematic sensitivity analysis of these parameters is an important direction for future work.

Additionally, the method assumes a closed-set adaptation setting, which can limit its applicability in real-world scenarios where new classes may appear in the target domain. Future work could explore more sophisticated feature disentanglement techniques to improve object and context information separation. Finally, extending the method to handle open-set and partial-domain adaptation scenarios would also increase its practical utility.

Author Contributions: Conceptualization, L.E.M., S.D. and T.B.; methodology, L.E.M. and S.D.; software, L.E.M.; validation, L.E.M., S.D. and T.B.; formal analysis, L.E.M. and S.D.; investigation, L.E.M.; resources, S.D.; data curation, L.E.M.; writing—original draft preparation, L.E.M.; writing—review and editing, L.E.M., S.D. and T.B.; visualization, L.E.M.; supervision, S.D. and T.B.; project administration, S.D.; funding acquisition, S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the MITACS Accelerate program (Grant Number IT19956) and the Natural Sciences and Engineering Research Council (NSERC) of Canada (Grant Number RGPIN-2018-04046).

Data Availability Statement: The SemanticKITTI dataset's official website is www.semantic-kitti.org. The SemanticPOSS dataset's official website is www.poss.pku.edu.cn/semanticposs.html. The SynLiDAR dataset's official website is www.github.com/xiaoaoran/SynLiDAR (all accessed on 27 July 2025).

Acknowledgments: The authors gratefully acknowledge the support of the MITACS Accelerate program, the Natural Sciences and Engineering Research Council, the Research Center in Geospatial Data and Intelligence of Université Laval, the Institute on intelligence and Data of Université Laval, and the Digital Research Alliance of Canada for access to their Advanced Research Computing platform and computational resources to complete our experiments.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UDA	Unsupervised Domain Adaptation
NT	Negative Transfer
MLP	Multilayer Perceptron
mIoU	mean Intersection over Union
BEVs	Bird-Eye-View images
DTE	Data Transferability Enhancement

MTE	Model Transferability Enhancement
SE-Net	Squeeze and Excitation
SGD	Stochastic Gradient Descent

References

1. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4376–4382.
2. Xiao, A.; Huang, J.; Guan, D.; Zhan, F.; Lu, S. Transfer Learning from Synthetic to Real LiDAR Point Cloud for Semantic Segmentation. *Proc. AAAI Conf. Artif. Intell.* **2022,** *36*, 2795–2803. [CrossRef]
3. Saltori, C.; Galasso, F.; Fiameni, G.; Sebe, N.; Poiesi, F.; Ricci, E. Compositional Semantic Mix for Domain Adaptation in Point Cloud Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 14234–14247. [CrossRef] [PubMed]
4. Yuan, Z.; Zeng, W.; Su, Y.; Liu, W.; Cheng, M.; Guo, Y.; Wang, C. Density-guided Translator Boosts Synthetic-to-Real Unsupervised Domain Adaptive Segmentation of 3D Point Clouds. *arXiv* **2024**, arXiv:2403.18469.
5. Zhao, H.; Zhang, J.; Chen, Z.; Zhao, S.; Tao, D. UniMix: Towards Domain Adaptive and Generalizable LiDAR Semantic Segmentation in Adverse Weather. *arXiv* **2024**, arXiv:2404.05145.
6. Xiao, A.; Huang, J.; Guan, D.; Cui, K.; Lu, S.; Shao, L. PolarMix: A General Data Augmentation Technique for LiDAR Point Clouds. *arXiv* **2022**, arXiv:2208.00223.
7. Yuan, Z.; Cheng, M.; Zeng, W.; Su, Y.; Liu, W.; Yu, S.; Wang, C. Prototype-Guided Multitask Adversarial Network for Cross-Domain LiDAR Point Clouds Semantic Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 5700613. [CrossRef]
8. Zhang, W.; Deng, L.; Zhang, L.; Wu, D. A Survey on Negative Transfer. *IEEE/CAA J. Autom. Sin.* **2023**, *10*, 305–329. [CrossRef]
9. Liang, J.; Wang, Y.; Hu, D.; He, R.; Feng, J. A Balanced and Uncertainty-aware Approach for Partial Domain Adaptation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020.
10. Li, S.; Xie, M.; Lv, F.; Liu, C.H.; Liang, J.; Qin, C.; Li, W. Semantic Concentration for Domain Adaptation. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 9082–9091.
11. Saleh, K.; Abobakr, A.; Attia, M.; Iskander, J.; Nahavandi, D.; Hossny, M.; Nahvandi, S. Domain Adaptation for Vehicle Detection from Bird's Eye View LiDAR Point Cloud Data. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 3235–3242. [CrossRef]
12. Zhao, S.; Wang, Y.; Li, B.; Wu, B.; Gao, Y.; Xu, P.; Darrell, T.; Keutzer, K. ePointDA: An End-to-End Simulation-to-Real Domain Adaptation Framework for LiDAR Point Cloud Segmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, AAAI, Online, 2–9 February 2021.
13. Yi, L.; Gong, B.; Funkhouser, T. Complete & Label: A Domain Adaptation Approach to Semantic Segmentation of LiDAR Point Clouds. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition CVPR, Nashville, TN, USA, 20–25 June 2021; pp. 15358–15368. [CrossRef]
14. Li, G.; Kang, G.; Wang, X.; Wei, Y.; Yang, Y. Adversarially Masking Synthetic to Mimic Real: Adaptive Noise Injection for Point Cloud Segmentation Adaptation. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 20464–20474.
15. Wang, Z.; Ding, S.; Li, Y.; Zhao, M.; Roychowdhury, S.; Wallin, A.; Sapiro, G.; Qiu, Q. Range Adaptation for 3D Object Detection in LiDAR. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 2320–2328. [CrossRef]
16. Luo, H.; Khoshelham, K.; Fang, L.; Chen, C. Unsupervised scene adaptation for semantic segmentation of urban mobile laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* **2020**, *169*, 253–267. [CrossRef]
17. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2242–2251. [CrossRef]
18. Rosenstein, M.T.; Marx, Z.; Kaelbling, L.P.; Dietterich, T.G. To transfer or not to transfer. In Proceedings of the Neural Information Processing Systems, Vancouver, BC, Canada, 5–8 December 2005.
19. Zhu, Y.; Zhuang, F.; Wang, J.; Ke, G.; Chen, J.; Bian, J.; Xiong, H.; He, Q. Deep Subdomain Adaptation Network for Image Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1713–1722. [CrossRef] [PubMed]
20. Long, M.; Wang, J.; Ding, G.; Cheng, W.; Zhang, X.; Wang, W. Dual Transfer Learning. In Proceedings of the SDM, Anaheim, CA, USA, 26–28 April 2012.
21. Tan, B.; Zhang, Y.; Pan, S.J.; Yang, Q. Distant Domain Transfer Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.

22. Na, J.; Jung, H.; Chang, H.; Hwang, W. FixBi: Bridging Domain Spaces for Unsupervised Domain Adaptation. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2020; pp. 1094–1103.
23. Wang, Z.; Dai, Z.; Póczos, B.; Carbonell, J.G. Characterizing and Avoiding Negative Transfer. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2018; pp. 11285–11294.
24. Chen, X.; Wang, S.; Fu, B.; Long, M.; Wang, J. Catastrophic Forgetting Meets Negative Transfer: Batch Spectral Shrinkage for Safe Transfer Learning. In Proceedings of the Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
25. Wang, X.; Jin, Y.; Long, M.; Wang, J.; Jordan, M.I. Transferable Normalization: Towards Improving Transferability of Deep Neural Networks. In Proceedings of the Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
26. Han, Z.; Sun, H.; Yin, Y. Learning Transferable Parameters for Unsupervised Domain Adaptation. *IEEE Trans. Image Process.* **2021**, *31*, 6424–6439. [CrossRef] [PubMed]
27. You, K.; Kou, Z.; Long, M.; Wang, J. Co-Tuning for Transfer Learning. In Proceedings of the Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020.
28. Zhang, J.O.; Sax, A.; Zamir, A.; Guibas, L.J.; Malik, J. Side-Tuning: A Baseline for Network Adaptation via Additive Side Networks. In Proceedings of the European Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019.
29. Yang, Y.; Huang, L.K.; Wei, Y. Concept-wise Fine-tuning Matters in Preventing Negative Transfer. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023; pp. 18707–18717.
30. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? *Adv. Neural Inf. Process. Syst.* **2014**, *27*.
31. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
32. Zhao, H.; Jiang, L.; Jia, J.; Torr, P.H.; Koltun, V. Point transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 16259–16268.
33. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
34. Pan, Y.; Gao, B.; Mei, J.; Geng, S.; Li, C.; Zhao, H. SemanticPOSS: A Point Cloud Dataset with Large Quantity of Dynamic Instances. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 19 October–13 November 2020; pp. 687–693.
35. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9296–9306. [CrossRef]
36. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
37. Choy, C.; Gwak, J.; Savarese, S. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, 15–20 June 2019; pp. 3070–3079. [CrossRef]
38. Selvaraju, R.R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2016**, *128*, 336–359. [CrossRef]
39. Ma, Y.; Guo, Y.; Liu, H.; Lei, Y.; Wen, G. Global Context Reasoning for Semantic Segmentation of 3D Point Clouds. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Aspen, CO, USA, 1–5 March 2020; pp. 2920–2929.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.