Unsupervised Domain Adaptation for Multi-Modal 3D Object Detection under Asymmetric Sensor Degradation

ME-CoR MSc Thesis

by

M.D. Yang

Student:	M. D. Yang | 5122546
Supervisor:	Dr. J.F.P. Kooij
Daily Supervisor:	S. Wang
Date:	May 11, 2026
Faculty:	Faculty of Mechanical Engineering, Delft
Department:	Cognitive Robotics

TU Delft

Unsupervised Domain Adaptation for Multi-Modal 3D Object Detection under Asymmetric Sensor Degradation

M.D. Yang — 5122546
Student
TU Delft CoR
m.d.yang@student.tudelft.nl

S. Wang
Daily Supervisor
TU Delft CoR

Dr. J.F.P. Kooij
Supervisor
TU Delft CoR

Abstract

Multi-modal 3D object detectors achieve state-of-the-art performance but remain notoriously brittle to asymmetric sensor degradation, such as when LiDAR point clouds become sparse in new environments. In this paper, we investigate unsupervised cross-modal adaptation to rescue a degraded sensor using an unaffected reference modality, without requiring target-domain labels. Using UniBEV on the nuScenes dataset, we simulate severe degradation by reducing LiDAR resolution from 32 to 8 beams. We systematically compare two leading adaptation paradigms anchored by the reliable camera stream: output-level camera pseudo-labeling and feature-level cross-modal mapping via a Bird's-Eye-View (BEV) Attention U-Net. Our experiments reveal a compelling insight: while feature mapping successfully aligns coarse spatial structures (improving LiDAR-only mAP by 5.6%), it fails to preserve fine-grained localization metrics. In contrast, simple confidence-filtered pseudo-labeling provides a significantly stronger recovery, yielding a 13.1% mAP improvement. Ultimately, our findings suggest that basic feature-level alignment may be insufficient to restore fine-grained 3D detection under severe spatial degradation, indicating that direct output-level supervision can be a more effective and reliable strategy for cross-modal adaptation in this regime.

1. Introduction

As AV systems are deployed across the world in diverse environments, they face significant challenges in maintaining robust perception due to distribution shifts in sensor data [24]. These shifts arise from changes in weather, lighting, road infrastructure, and sensor configurations, leading to performance degradation of perception models when deployed in new domains [15, 17, 24]. Collecting and annotating new data for every deployment location is impractical, necessitating methods that can adapt existing models to new domains without requiring extensive labeled target-domain data. Especially when one realizes that to train an AV system from scratch requires millions of labeled data

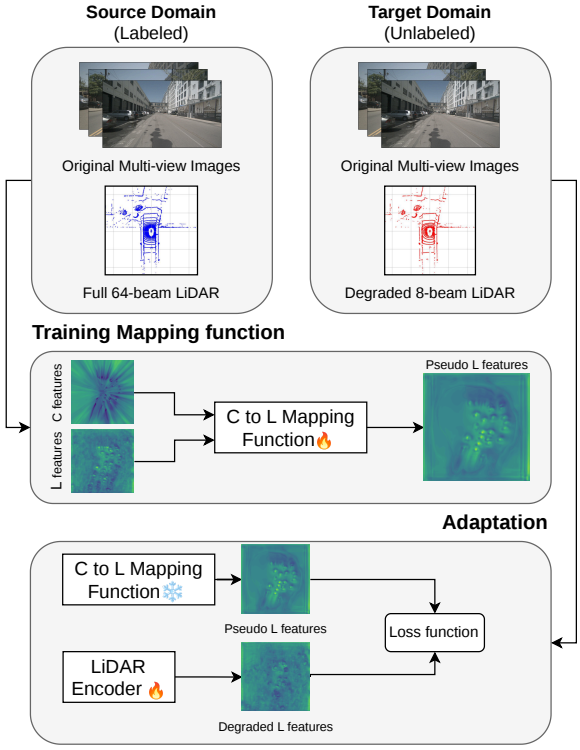

Figure 1. **Overview of our CM-FKD framework.** We address the challenge of cross-modal degradation by leveraging a robust camera modality as an anchor. (Top) In the source domain, we learn a mapping function from camera to LiDAR Bird's-Eye-View (BEV) features. (Bottom) During adaptation, the frozen camera encoder and mapping function generate "Pseudo-LiDAR" features from the target domain images. These serve as a stable supervisory signal to align and adapt the degraded LiDAR encoder without requiring target labels. Fire indicates trainable and snowflake indicates frozen parameters

points [1, 4], which is infeasible to repeat for every new domain.

Specifically, in multi-modal systems, sensor modalities frequently demonstrate asymmetric failure modes. LiDAR degrades heavily under motion-level corruption and adverse weather, while cameras maintain more stable performance across these same shifts [10, 15, 32]. This asymmetry presents a unique opportunity: when one modality degrades, we can leverage the unaffected modality as a stable anchor for cross-modal knowledge transfer, recovering the degraded sensor's performance without manual annotation.

LiDAR degradation is particularly critical to address, as its output is weighted more heavily in the fusion process and its failure can lead to severe performance drops compared to camera failing [11, 19, 27]. When degrading the resolution of the sensors in a camera and LiDAR system, the LiDAR's performance drops significantly more than the camera's, creating a clear asymmetric degradation scenario [32].

This observation motivates our focused study of *asymmetric sensor degradation*: a regime where one modality remains relatively robust while another severely degrades in a new domain. In our case, we simulate this by reducing LiDAR resolution from 32 beams to 8 beams, while keeping the camera modality intact. This setup allows us to systematically investigate how to best leverage the robust camera stream to recover the degraded LiDAR's performance through unsupervised cross-modal adaptation.

Standard unsupervised domain adaptation (UDA) methods typically assume uniform domain shift across all inputs or focus solely on mono-modal adaptation [28, 30, 31]. Even multi-modal UDA approaches primarily rely on using the unaffected modality only during source domain training [6, 20]. Existing literature overlooks the compelling possibility of actively using the unaffected modality to supervise the degraded one dynamically during target domain adaptation.

To address this gap, we design a methodology to systematically compare two paradigms of cross-modal adaptation under asymmetric degradation: standard output-level *pseudo-labeling* and our proposed framework, *Cross-Modal Feature-Level Knowledge Distillation (CM-FKD)*. Unlike standard UDA, our approach translates the robust camera signal into the degraded LiDAR's native feature space to provide dense, structural supervision.

Our CM-FKD framework utilizes an explicit two-phase mapping logic:

1. **C.M.F.M. Learning (Source Domain):** We train an Attention U-Net to reconstruct full high-fidelity LiDAR BEV features (e.g., 32-beam) directly from camera features. This network learns the complex cross-modal relationship where labels and clean data are abundant.
2. **Knowledge Distillation (Target Domain):** We freeze the mapper and generate "Pseudo-LiDAR" features from target domain images. These clean features serve as a stable supervisory signal to guide the degraded LiDAR encoder (e.g., 8-beam), requiring zero target labels.

An overview of CM-FKD is illustrated in Fig. 1 and detailed in Sec. 3.

Fundamentally, this thesis investigates the core research question: **Under asymmetric sensor degradation and unlabeled domain shift, when does feature-level mapping outperform simple output-level pseudo-labeling?** We investigate this through three specific sub-questions:

1. What is the baseline degradation when one modality is compromised without any adaptation?
2. Which strategy gives the best recovery: no adaptation, pseudo-labeling or cross-modal feature mapping?
3. Can feature-level distillation using reconstructed features improve the degraded encoder's performance in the target domain?

By employing PCA and variance analysis to interpret the feature representations, we observe that while our feature mapper successfully aligns overarching coarse structural patterns between modalities, it struggles to recover fine-grained spatial acuity. Consequently, we demonstrate that direct output-level pseudo-labeling remains ultimately more effective for recovering strict detection metrics like mAP (13.1% vs 5.6%), despite the theoretical appeal of dense feature distillation.

2. Related Work

2.1. AV Datasets

Domain adaptation research relies heavily on creating reproducible domain shifts. Most often, common AV datasets are used to test for domain adaptation. Either by splitting existing datasets into different domains based on the domain gap (nuScenes into day/night or Boston/Singapore) or by using different datasets as source and target (e.g., KITTI to nuScenes) as shown used by [6, 18, 20, 23]. The domain adaptation is then evaluated based on the performance drop when testing on the target domain without adaptation.

The most common AV datasets for 3D Object Detection include KITTI [3], nuScenes [4], Waymo Open Dataset [25]. These datasets provide a variety of sensor data, including LiDAR, camera, and radar, across different geographic locations and environmental conditions. They have been widely used for training and evaluating 3D object detection models, as well as for studying domain adaptation. However, they often contain multiple domain shifts at once, making it difficult to isolate specific factors contributing to performance degradation.

In terms of sensor degradation, there is no pre-defined dataset that simulates sensor degradation in a controlled manner. Therefore, researchers often simulate degradation by modifying the existing datasets, such as reducing the res-

olution of LiDAR point clouds or camera images, adding noise, or simulating adverse weather conditions. This allows for testing the robustness of models under specific types of sensor degradation and evaluating the effectiveness of domain adaptation techniques in mitigating performance drops [10, 15, 16, 33].

2.2. Multi-Modal 3D Object Detection

3D Object Detection in Autonomous Vehicle (AV) systems is a critical task that involves classifying and localizing objects in 3D space using sensor data. The goal is to enable AVs to understand their surroundings and make informed decisions for safe navigation. Uni-modal models only use one sensor modality, I.E., LiDAR, cameras, or radar, while multi-modal models combine data from multiple sensors to leverage their complementary strengths.

SoTA multi-modal methods combine the strengths of both modalities, often using LiDAR for accurate depth estimation and cameras for rich semantic information through feature fusion. Leading to superior performance when tested on AV datasets [24]. These methods typically involve a shared feature space where features from both modalities are fused to improve detection performance. [14] Birds' Eye View (BEV) has recently become the paradigm shared feature space used in 3D object detection [2, 5, 11, 14, 19, 24, 27, 29, 34]. BEV features provide a top-down view of the scene, allowing for effective fusion of LiDAR and camera data.

For ease of use, we adapt UniBEV [27] as our base detector, which uses a shared BEV feature space for multi-modal fusion and has shown strong performance on the nuScenes dataset [4]. UniBEV's architecture allows for flexible integration of different modalities and serves as a suitable platform for testing our proposed cross-modal feature mapping approach under asymmetric sensor degradation.

2.3. Knowledge Distillation and Cross-Modal learning

Knowledge Distillation (KD) is a technique where a smaller, often less complex model (the student) learns to mimic the behavior of a larger, more complex model (the teacher) [12, 26]. This is typically achieved by training the student to replicate the teacher's output distributions or intermediate feature representations. KD can be classified in mainly three categories: response-based, feature-based, and relation-based distillation. Response-based distillation focuses on matching the output logits of the teacher, while feature-based distillation aims to align intermediate feature representations. Relation-based distillation captures the relationships between different samples or features in the teacher's output.

MonoDistill [8] aligns modalities in image space by projecting LiDAR into camera views, but its transfer quality depends on noisy depth estimation. BEVDistill [7] and UniDistill [35] move distillation to shared BEV space, improving cross-modal alignment through feature-level (and in UniDistill also relation- and response-level) supervision with flexible teacher student assignments. However, these methods mainly target uni-modal performance gains and do not explicitly address unlabeled target-domain adaptation under sensor degradation. In contrast, our work uses cross-modal BEV mapping for adaptation, leveraging an unaffected modality to improve a degraded modality in unseen domains without target labels.

2.4. Modality Robustness

Prior work shows that sensor robustness is corruption-dependent rather than absolute. In [17] they report strong LiDAR sensitivity to several corruptions, while [33] find that under noisy sensing and synchronization issues, camera inputs can be more resilient than LiDAR. The large-scale benchmark of [10] confirms that robustness varies by corruption type and model design, and that fusion generally benefits from modality complementarity.

However they find that generally in fusion, Camera degrades more with the corruptions they tested, even for weather types. [15] in contrast found that LiDAR is more sensitive to weather corruption compared to Camera. In our target setting of hardware-like degradation, our prior study [32] shows a larger performance drop from LiDAR beam reduction than from camera resolution reduction. In fact, lowering the camera resolution showed that the performance of Camera barely got affected.

[22] compares the performance of LiDAR-only models and found that the LiDAR model always degrades when moving to another sensor, independent of whether the LiDAR used for training was of high- or low-resolution. Therefore, this thesis adopts an asymmetric degradation assumption: camera features serve as the relatively unaffected supervision signal to adapt a degraded LiDAR branch in unlabeled target domains.

2.5. Domain Adaptation for Autonomous Driving Perception

Domain Adaptation (DA) aims to address the challenge of distributional shifts between training (source) and deployment (target) environments. In the context of autonomous driving (AV) perception, DA is crucial due to the variability in environmental conditions, sensor configurations, and geographic locations that AV systems may encounter. Without effective DA techniques, models trained on one domain may perform poorly when deployed in a different domain. Unsupervised Domain Adaptation (UDA) methods are particularly relevant, as they do not require labeled data in the target domain, making them practical for real-world applications.

Unsupervised Domain Adaptation methods in AV have primarily focused on mono-modal settings [18, 23, 28]. For instance, ST3D [30] and ST3D++ [31] are self-training frameworks for UDA in 3D object detection that focus on generating high-quality pseudo-labels in the target domain. ST3D first pre-trains the detector on the source domain using a Random Object Scaling (ROS) strategy to mitigate source domain bias. The detector then produces initial pseudo-labels on the target domain, which are filtered via a class-specific confidence threshold to select high-confidence predictions. This ensures a balanced selection of pseudo-labels across different categories. ST3D++ improves upon this by introducing a denoising strategy that removes noisy pseudo-labels through iterative refinement. These approaches demonstrate that pseudo-labeling is an effective adaptation strategy when high-quality predictions are prioritized.

However, a principal limitation of ST3D and ST3D++ is that they focus almost exclusively on uni-modal LiDAR UDA. They assume the exact same sensor configuration across domains and do not leverage secondary modalities, like cameras, to compensate for LiDAR degradation. When self-training relies on pseudo-labeling within the same degraded modality, it risks entering a "vicious circle" of error accumulation, especially if the sensor is severely hardware-degraded. In our work, we adopt a similar baseline strategy of confidence thresholding, but crucially shift to cross-modal supervision: generating pseudo-labels from an unaffected camera modality to break this cycle and guide the degraded LiDAR modality safely.

Beyond mono-modal methods, there has been limited exploration of multi-modal approaches for domain adaptation. Two recent works make use of multi-modal data for domain adaptation.

CMDA [6] is a framework that addresses the task of unsupervised domain adaptation for LiDAR-based 3D object detection by combining cross-modal distillation with domain adversarial training. The method is divided into two stages: LiDAR encoder pretraining and self-training. In the pretraining stage, CMDA leverages the complementary strengths of LiDAR and camera modalities by projecting them into a shared bird's-eye view (BEV) space, where features from both modalities can be spatially aligned. A cross-modal distillation objective is then applied to enforce consistency between LiDAR and camera BEV features, enabling the LiDAR branch to inherit semantic priors from the image modality. Once the LiDAR encoder is enriched through this interaction, the self-training stage applies domain adversarial learning, where a discriminator enforces domain invariance by penalizing features that reveal whether they come from the source or target domain. In this way, CMDA effectively combines cross-modal supervision and domain alignment to improve LiDAR-based 3D detection in the target domain. However, the cross-modal interaction is limited to a pretraining phase in the source domain. Its adaptation stage solely relies on domain adversarial training to make LiDAR features domain-invariant. Meaning that it does not leverage the stable camera modality during adaptation in the target domain to recover structural information lost in the degraded LiDAR features, which is critical under severe asymmetric sensor degradation.

DualCross [20] is a framework designed to address the challenges of cross-modality and cross-domain adaptation in monocular bird's-eye-view (BEV) 3D object detection. Unlike traditional methods that focus on a single domain or modality, DualCross aims to enhance monocular BEV perception by transferring knowledge from LiDAR sensors in one domain to camera-only testing scenarios in a different domain. The framework employs a two-stage approach: first, a LiDAR-Teacher model is trained using voxelized LiDAR point clouds to transform image features into the BEV frame, providing essential knowledge on how to guide image learning given LiDAR information. Second, a Camera-Student model is supervised by both the teacher model and the LiDAR ground truth, enabling it to learn from the rich 3D information provided by the LiDAR sensor. Additionally, discriminators are used to align features from source and target domains, ensuring domain invariance. By combining cross-modality knowledge transfer with cross-domain adaptation, DualCross effectively facilitates monocular BEV perception in real-world scenarios where only camera data is available during testing. However, the cross-modal interaction is fundamentally restricted to a pre-training/training stage where LiDAR acts as a teacher for a camera-student. Because DualCross aims for camera-only perception at test time, it does not explore using a robust camera modality to "rescue" a degraded LiDAR encoder during target-domain adaptation. This highlights a key gap in addressing asymmetric sensor degradation where both modalities remain present but one has severely failed.

While these methods are effective for standard domain shifts (e.g., changing geographic locations like Waymo to KITTI or removing a sensor entirely at test time), they do not focus on asymmetric sensor degradation. In our work, we isolate a scenario where one modality (camera) remains robust while another (LiDAR) is hardware-degraded in a new environment.

Furthermore, methods like CMDA and DualCross heavily focus on aligning the camera and LiDAR feature spaces, which is highly important for multi-modal fusion performance. In our research, we assume this foundational spatial alignment is already intrinsically handled by the shared BEV architecture of our base 3D object detector. Consequently, our work bypasses the alignment objective to solely focus on learning a direct, robust predictive mapping from

camera features to LiDAR features.

To evaluate adaptation in this unique regime, we compare two strategies: a naive pseudo-labeling baseline that generates bounding box predictions from the camera modality and filters them by confidence, and a feature-level mapping approach that reconstructs intermediate BEV features for the degraded LiDAR modality using the unaffected camera features as supervision. The former relies on potentially noisy output-level predictions, while the latter aims to recover structural information lost at the feature level, which is critical under severe sensor degradation.

Unlike CMDA or DualCross, which restrict cross-modal interaction to pre-training, our feature mapping framework enables the camera to provide a stable supervisory signal through the learned feature mapping during the entire adaptation process. By aligning the degraded modality's feature representation with our stable camera "anchor" in the target domain, our model recovers missing semantic cues. This ensures robustness even under severe sensor-level failures where the bounding box predictions would otherwise be too sparse or noisy to provide a useful training signal.

2.6. Contributions

This thesis investigates cross-modal feature mapping for unsupervised adaptation under sensor degradation in multi-modal 3D object detection. The main contributions are:

- We investigate and propose CM-FKD, cross-modal feature knowledge distillation as a mechanism to recover degraded LiDAR representations.
- We show that feature-level reconstruction preserves coarse spatial structure but fails at precise localization.
- We demonstrate that camera pseudo-labeling provides stronger adaptation signals than dense feature reconstruction.

3. Method

3.1. Problem Formulation

We address the problem of unsupervised cross-modal domain adaptation for multi-modal 3D object detection under *asymmetric sensor degradation*. The objective is to adapt a degraded sensor modality in an unseen target domain by leveraging a robust sensor modality as a stable "anchor", without access to target-domain annotations.

Let C and L denote two sensor modalities, where C is the unaffected modality (camera) and L is a degraded modality (low-resolution LiDAR). We are given a labeled source-domain dataset

$$\mathcal{S} = \{(x_i^C, x_i^L, y_i)\}_{i=1}^N, \tag{1}$$

where $x_i^C \in \mathcal{X}_C$ and $x_i^L \in \mathcal{X}_L$ are paired observations from modalities C and L, respectively, and $y_i \in \mathcal{Y}$ denotes the corresponding 3D object detection annotations.

In the target domain, we are given an unlabeled dataset

$$\mathcal{T} = \{(z_j^C, z_j^L)\}_{j=1}^M, \tag{2}$$

where $z_j^L \in \mathcal{Z}_L$ follow a different data distribution than the source domain, i.e.,

$$p_{\mathcal{S}}(x^L) \neq p_{\mathcal{T}}(z^L). \tag{3}$$

No ground-truth annotations are available for the target domain. We assume that the unaffected modality C is not affected by the domain shift, such that

$$p_{\mathcal{S}}(x^C) \approx p_{\mathcal{T}}(z^C) \tag{4}$$

or is not affected by the degradation at all. This assumes that the camera sensor is not being swapped out, and that the camera data distribution remains consistent across domains, while the LiDAR data distribution shifts due to degradation (sensor swap).

We consider a multi-modal 3D object detector, pre-trained on $\mathcal{S}$, composed of modality-specific encoders E_C and E_L for modalities C and L, respectively, followed by a fusion module and detection head. The encoders extract intermediate feature representations from the input data:

$$E_C : \mathcal{X}_C \to \mathcal{F}_C, \quad E_L : \mathcal{X}_L \to \mathcal{F}_L, \tag{5}$$

where $\mathcal{F}_C, \mathcal{F}_L \subset \mathbb{R}^{C \times H \times W}$ denote intermediate bird's-eye-view (BEV) feature representations. Let

$$f_x^C = E_C(x^C), \quad f_x^L = E_L(x^L) \tag{6}$$

denote the extracted modality-specific features.

Due to domain shift (sensor degradation), the feature distribution of the degraded modality encoder E_L may become misaligned in the target domain

$$p_{\mathcal{S}}(E_L(x^L)) \neq p_{\mathcal{T}}(E_L(z^L)), \tag{7}$$

resulting in degraded detection performance.

Our goal is therefore to adapt the degraded modality encoder E_L in the target domain such that its feature representations remain consistent with those induced by the unaffected modality, thereby improving detection performance without requiring target-domain labels.

3.2. Camera Pseudo-Labeling

A straightforward approach to leverage the unaffected modality for adaptation is to use it to generate pseudo-labels for the degraded modality. Specifically, we can use the pre-trained multi-modal detector to generate predictions based on the unaffected modality's features in the target domain. These predictions can then serve as supervisory signals to fine-tune the degraded modality encoder. In figure 2 we illustrate the pipeline for this method.

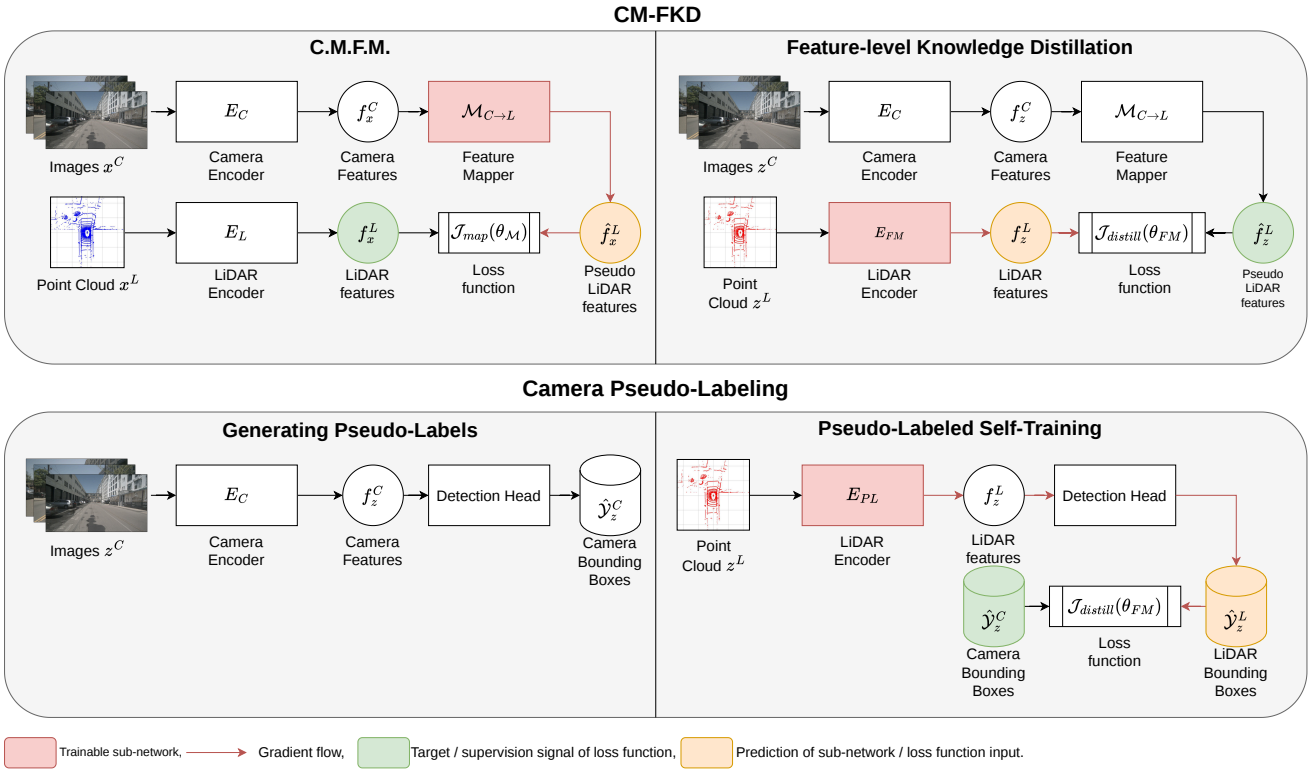

Figure 2. **Overview of two main methods.** Above we display the pipeline for the proposed cross-modal feature mapping framework CM-FKD, which consists of two main phases. (1) C.M.F.M. learning: we learn a cross-modal feature mapping network that reconstructs the degraded-modality features from the unaffected-modality features on the source domain. (2) Feature-level knowledge distillation: we use the learned mapping to generate pseudo-degraded-modality features in the target domain, and adapt the degraded modality encoder by minimizing a feature-level distillation loss between the generated pseudo-features and the features extracted by the adapted encoder. Below we display the pipeline for camera pseudo-labeling, which uses the unaffected modality to generate pseudo-labels for fine-tuning the degraded modality encoder in the target domain. Both methods leverage the unaffected modality to guide adaptation of the degraded modality without requiring target-domain annotations.

Since the C modality is unaffected by the domain shift, we can directly use the pre-trained detector to generate pseudo-labels for the target domain samples. For a target-domain sample pair (z^C, z^L), we compute the unaffected-modality features $f^C = E_C(z^C)$ and feed them into the detection head to obtain pseudo-labels $\hat{y}^C$. For all samples in the target domain, we can express this as:

$$\hat{\mathcal{Y}}_z^C = \text{Detector}(E_C(z_j^C|_{j=1}^M)), \tag{8}$$

where $\hat{\mathcal{Y}}_z^C = \{\hat{y}_j^C\}_{j=1}^M$ denotes the set of pseudo-labels generated for the target domain samples.

We can then use these pseudo-labels to fine-tune the degraded modality encoder E_L by minimizing a supervised loss between the predictions based on $E_L(z^L)$ and the pseudo-labels $\hat{\mathcal{Y}}_z^C$. This approach allows us to leverage the unaffected modality to guide the adaptation of the degraded modality in the target domain, without requiring any ground-truth annotations. However, the effectiveness of this method may be limited by the quality of the pseudo-labels, which can be noisy due to the worse performance compared to the L modality. To combat this, we filter the bounding boxes $\hat{\mathcal{Y}}_z^C$ by confidence score, only keeping those with a confidence score above a certain threshold. This helps to reduce the noise in the pseudo-labels and improve the stability of the fine-tuning process.

To select the confidence score threshold, we use the training set of nuScenes. We obtain the bounding boxes from the pre-trained detector on the training set and compute the confidence scores for each bounding box. Finally, we calculate the precision, recall and f1 score as shown in Equation 11.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{9}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{10}$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{11}$$

These metrics are computed for different confidence score thresholds, and we select the threshold that yields the highest F1 score on the validation set. By maximizing the F1 score, we aim to find a balance between precision and recall, ensuring that the pseudo-labels used for fine-tuning are both accurate and comprehensive enough to guide effective adaptation of the degraded modality encoder.

Finally, the loss function for self-training using pseudo-labels generated from E_C is expressed in Equation 12. The fine-tuned degrade modality is promptly renamed as $E_L \rightarrow E_{PL}$ *(PL means Pseudo-labels)*.

$$\mathcal{J}_{\text{pseudo}}(\theta_L) = \mathbb{E}_{(z^C, z^L)\sim\mathcal{T}}\left[\ell(\hat{y}_z^C, y_z^L)\right] \tag{12}$$

Where θ_L is the set of parameters of the LiDAR encoder E_L,$\hat{y}_z^C$ are the pseudo-labels generated from the camera modality, and y_z^L denotes the bounding boxes obtained through the detection head using E_{FM}. For the loss function $\ell(\cdot,\cdot)$, we use the same loss function as the detection head of the pre-trained multi-modal 3D object detector, which includes a combination of classification and regression losses. Specifically, a Focal Loss is employed for handling the class imbalance in the detection task, and an L1 loss is applied for bounding box regression. By minimizing this loss, we encourage the adapted encoder E_{PL} to produce feature representations that lead to predictions consistent with the pseudo-labels generated from the unaffected modality, thereby improving its performance in the target domain.

3.3. Cross-Modal Feature-Level Knowledge Distillation (CM-FKD)

Figure 2 illustrates our proposed cross-modal feature mapping framework for unsupervised domain adaptation in multi-modal 3D object detection. The framework consists of three main components: (1) modality-specific encoders for robust and degraded sensor modalities, (2) a cross-modal feature mapping network, and (3) a feature-level knowledge distillation mechanism.

3.3.1. Feature Extraction and Spatial Alignment

Given a paired input sample $x = (x^C, x^L)$ from the source domain, we first extract modality-specific features using the encoders E_C and E_L of the pre-trained multi-modal 3D object detector:

$$f_x^C = E_C(x^C), \quad f_x^L = E_L(x^L). \tag{13}$$

Here, f_x^C and f_x^L denote the BEV feature representations for modalities C and L, respectively. Crucially, we assume that the foundational spatial alignment between the camera and LiDAR modalities is already intrinsically handled by the shared BEV architecture of our base 3D object detector (e.g., UniBEV). Because the features f_x^C and f_x^L already reside in a unified spatial grid, our framework can safely bypass complex cross-modal spatial alignment objectives and instead focus entirely on learning a direct, predictive semantic mapping between the two distributed representations.

3.3.2. Cross-Modal Feature Mapper (C.M.F.M.)

To learn the relationship between the unaffected and degraded modalities, we introduce a cross-modal feature mapping network $\mathcal{M}_{C\rightarrow L}$. This network aims to reconstruct the degraded-modality features from the unaffected-modality features. As input it accepts the unaffected-modality features f_x^C and outputs the reconstructed degraded-modality features $\hat{f}_x^L = \mathcal{M}_{C\rightarrow L}(f_x^C)$. The mapping is learned by minimizing the feature reconstruction loss between the input f_x^C and the supervision signal f_x^L on the source domain:

$$\mathcal{J}_{\text{map}}(\theta_{\mathcal{M}}) = \mathbb{E}_{(x^C, x^L)\sim\mathcal{S}}\left[\ell\big(f_x^L, \hat{f}_x^L\big)\right], \tag{14}$$

where $\ell(\cdot,\cdot)$ denotes a feature-level distance metric. The mapping network is trained and the rest of the network is frozen during this phase, ensuring that the mapping function learns to reconstruct the degraded modality's features based solely on the unaffected modality's features.

The mapping network used is an Attention UNet architecture [21], which is designed to capture both local and global feature relationships. The UNet structure allows for multi-scale feature learning, while the attention mechanism helps to focus on the most relevant features for reconstruction.

3.3.3. Feature-level Knowledge Distillation

In the target domain, we leverage the learned cross-modal feature mapping $\mathcal{M}_{C\rightarrow L}$ to guide the fine-tuning of the degraded modality encoder E_L. For a target-domain sample (z^C, z^L), we compute the following features:

$$f_z^C = E_C(z^C), \quad f_z^L = E_L(z^L). \tag{15}$$

We then use the mapping network to obtain the pseudo-LiDAR features $\hat{f}_z^L = \mathcal{M}_{C\rightarrow L}(f_z^C)$. The degraded modality encoder E_L is adapted and becomes E_{FM} *(FM means Feature Mapper)* by minimizing the feature-level knowledge distillation loss:

$$\mathcal{J}_{\text{distill}}(\theta_{FM}) = \mathbb{E}_{(z^C, z^L)\sim\mathcal{T}}\left[\ell\big(\hat{f}_z^L, f_z^L\big)\right], \tag{16}$$

where $\hat{f}_z^L$ is the supervision signal and f_z^L is the output of the adapted encoder E_{FM}. Here $\ell(\cdot,\cdot)$ is a feature-level

distance metric. We minimize the loss between the features extracted by the adapted encoder E_{FM} and the pseudo-LiDAR features generated by the mapping network. This encourages the adapted encoder to produce feature representations that are consistent with those reconstructed from the unaffected modality, thereby improving its performance in the target domain without requiring any labeled target data.

For both the C.M.F.M. learning and the feature-level knowledge distillation phases, we use the Mean Squared Error (also known as L2 norm) loss as the feature-level distance metric, which encourages the adapted encoder to closely match the pseudo-LiDAR features generated by the mapping network. We chose this loss function as it is a common choice for feature-level distillation tasks and has been shown to be effective in encouraging the student model (adapted encoder) to learn from the teacher model (mapping network) by minimizing the distance between their feature representations [6, 7, 20]

3.4. Feature Space Analysis

To justify our comparative study between output-level pseudo-labeling and feature-level distillation, we systematically analyze what the Cross-Modal Feature Mapper (C.M.F.M.) learns to reconstruct. We hypothesize that while the feature mapper struggles to synthesize fine-grained spatial details, it successfully preserves the overarching coarse structural features of the LiDAR BEV space.

To formalize this qualitative rationale, we employ Principal Component Analysis (PCA) and spatial variance analysis on the reconstructed "Pseudo-LiDAR" features $\hat{f}^L$ versus the ground truth 32-beam features f^L. By extracting and projecting the principal components of the dimensional feature maps, we visually and quantitatively assess the extent to which major structural variances (coarse details) are retained and minor variances (fine-grained object boundaries critical for mAP) are blurred. This framework explicitly allows us to evaluate the trade-offs of feature-level mapping compared to boundary-exact output-level pseudo-labeling.

3.5. Assumptions

Both methods follow the assumption that the unaffected modality (camera) is not affected by the domain shift, and that its data distribution remains consistent across domains. This allows us to leverage the unaffected modality as a reliable source of information for guiding the adaptation of the degraded modality in the target domain. Additionally, we assume that the pre-trained multi-modal 3D object detector has learned meaningful feature representations that can be effectively utilized for both pseudo-labeling and feature mapping, enabling successful adaptation without requiring target-domain annotations. Furthermore, we assume that only retraining the affected modality encoder is sufficient for adaptation, and that the fusion module and detection head can remain unchanged during the adaptation process. This simplifies the adaptation process and focuses on aligning the feature representations of the degraded modality with those of the unaffected modality.

4. Experiments

4.1. Implementation Details

Dataset and Metrics. The nuScenes [4] dataset is a large-scale multi-modal dataset for autonomous driving, which includes paired LiDAR and camera data across multiple urban environments. We train both methods on the full nuScenes training data split. We evaluate our method on the nuScenes validation set, which consists of 1,000 scenes with annotated 3D bounding boxes for various object classes.

For fine-tuning the degraded LiDAR modality encoder through camera pseudo-labeling, we use only the LiDAR data with the camera-generated pseudo-labels and bounding boxes.

For training the Feature Mapper, we use the paired camera and LiDAR data from the source domain (32-beam LiDAR) to learn the mapping. For fine-tuning the degraded modality encoder through cross-modal feature mapping, we use the unlabeled paired camera and LiDAR target domain data (8-beam LiDAR) for unsupervised adaptation.

We report results on the full validation set to ensure a comprehensive evaluation of our method's performance under sensor degradation. We use mean Average Precision (mAP) and nuScenes Detection Score (NDS) [4] as our primary evaluation metrics for 3D object detection performance. We select both of these metrics, since mAP tells us about whether the object was found correctly according to a threshold (in nuScenes it is the 2D center distance). NDS provides a more comprehensive assessment of detection quality by considering the true positive errors. We additionally report extra information regarding true positive errors and AP based on the distance threshold.

Model. We use UniBEV [27], a pre-trained multi-modal 3D object detector as our base model, which consists of modality-specific encoders for the camera and LiDAR modalities, followed by a fusion module and detection head. Specifically, we use the model that is trained with the *channel normalized weights* fusion strategy. The encoders extract intermediate bird's-eye-view (BEV) feature representations from the input data. When we retrain the degraded modality encoder in both methods, we initialize it with the weights of the original LiDAR encoder and fine-tune it using the feature-level knowledge distillation from the UNet-generated pseudo-LiDAR features or the camera pseudo-labels and bounding boxes. The fusion module and

detection head are kept frozen during this process to ensure that the adaptation focuses on aligning the degraded modality features with the robust modality features without altering the overall detection architecture.

For the cross-modal feature mapping network, we adopt an Attention UNet architecture [21], which was primarily designed for image segmentation tasks. We adapt the UNet architecture to operate on the BEV feature maps of $C \times H \times W$ extracted by the encoders, allowing it to learn the mapping between the robust and degraded modality features effectively. The attention mechanism in the UNet helps to focus on the most relevant features for reconstruction, which is crucial for handling the domain shift caused by sensor degradation.

Since UniBEV produces BEV features at a resolution of $256 \times 200 \times 200$, we design our UNet to take these feature maps as input and output reconstructed features of the same size. Following the original UNet design naively, meaning we double the number of channels at each downsampling step, would lead to an unmanageable number of parameters and risk overfitting. Therefore, we modify the UNet architecture to maintain a more reasonable number of channels while still capturing the necessary feature relationships. We use a base channel size of 128, doubling it to 256 at the first downsampling step, doubling it again to 512 at the second downsampling step and maintaining it at 512 for the last step.

The pre-trained UniBEV model we call E_L for the LiDAR encoder and E_C for the camera encoder. The retrained LiDAR encoder is denoted as E_{FM}. The UNet-based model which translates camera into pseudo-LiDAR features is denoted as $\mathcal{M}_{C \to L}$. The retrained LiDAR encoder using camera pseudo-labels is denoted as E_{PL}.

Baselines. Since as far as we know, there are no existing methods that directly address the problem of unsupervised domain adaptation under sensor degradation in multi-modal 3D object detection, we compare against several baselines that represent different levels of performance under degradation. For both methods we compare against the original pre-trained UniBEV model, which serves as a baseline for performance under degradation without any domain adaptation. This allows us to assess the performance drop due to sensor degradation and the effectiveness of our adaptation methods in recovering performance.

For the C.M.F.M. module, we specifically compare it against the original LiDAR encoder E_L that has been trained on 32 beam data and is inferenced on 32 beam data. This serves as a baseline for the performance of the UNet-based pseudo-LiDAR generation when both are evaluated on the non-degraded data. This allows us to assess how well the Feature Mapper can reconstruct the LiDAR features from the camera features in the source domain, which is crucial for its effectiveness in guiding the adaptation of the degraded modality.

Then, the main two methods of Camera Pseudo-Labeling and CM-FKD are both compared against the original LiDAR encoder E_L that has been trained on 32 beam data and is inferenced on 8 beam data. This serves as a baseline for performance under degradation without any adaptation, allowing us to assess the effectiveness of both methods in recovering performance under sensor degradation.

Training Details.

We followed the standard data augmentation done to the input data during training of UniBEV [27]. For the LiDAR point clouds, we applied point shuffle, which randomly shuffles the order of the points in the point cloud. This helps to improve the robustness of the model to variations in the input data and prevents overfitting during training. For the camera images, we applied photometric distortions, which randomly adjust the brightness, contrast, saturation, and hue of the images. This augmentation technique helps to improve the model's ability to generalize to different lighting conditions and color variations in the target domain.

In table 1 you can find the training hyperparameters for all the methods. We first train CM-FKD. The C.M.F.M. module was trained till convergence on the source domain $\mathcal{S} = \{(x_i^C, x_i^L)\}_{i=1}^N$ without any labels. The Feature-Level Knowledge Distillation was then performed on the target domain $\mathcal{T} = \{(x_j^C, x_j^L)\}_{j=1}^M$ without any labels. For the pseudo-labeling method, we generate pseudo-labels using the camera encoder E_C and the detection head on the target domain data, and then fine-tune the degraded modality encoder E_{PL} using these pseudo-labels as supervisory signals. For every method, we only trained the corresponding subset of the model parameters, while keeping the rest of the model frozen to ensure a fair comparison and to focus on the adaptation of the degraded modality.

All models were trained on Tesla V100-SXM2-32GB gpu's. Our model is implemented in the open-sourced MMDetection3D [9]. A detailed summary of the training hyperparameters utilized across our framework components follows in Table 1. The Feature Mapper was trained for roughly 20 hours, the fine-tuning of the degraded modality encoder using camera pseudo-labels took roughly 4 days, and the fine-tuning of the degraded modality encoder using feature-level knowledge distillation took roughly 5 days.

4.2. Results

4.2.1. Quantitative Results

Performance Degradation. Table 2 we can see the effects of the LiDAR degradation on the performance of the original LiDAR encoder E_L. The left column under the section *Performance Degradation* shows the results for the original LiDAR encoder E_L when evaluated on the non-degraded data (32 beams) and the degraded data (8 beams). The right

Table 1. **Training Hyperparameters** for the Feature Mapper ($\mathcal{M}_{C\to L}$) and the two encoder fine-tuning configurations (E_{FM} and E_{PL}).

Hyperparameter	$\mathcal{M}_{C\to L}$	E_{FM}	E_{PL}
Batch size	5	5	5
Epochs	10	26	26
Optimizer	AdamW	AdamW	AdamW
Learning rate	2×10^{-3}	1×10^{-4}	2×10^{-3}
Weight decay	0.1	0.1	0.1
Schedule	Cosine Annealing	Cosine Annealing	Cosine Annealing
Loss weights	1.0 MSE	1.0 MSE	2.0 Focal cls, 0.25 L1 Bbox

Table 2. **Comprehensive evaluation of adaptation methods under LiDAR sensor degradation.** Results are reported on nuScenes validation set for camera+LiDAR and LiDAR-only configurations. mAP and NDS metrics are shown with relative change (%) computed with respect to the corresponding baseline. ↑ indicates higher is better. Relative change calculated as $\frac{result-baseline}{baseline} \times 100\%$ against baseline colored in . indicates a negative change (performance drop), while indicates a positive change (performance improvement).

Method	**LiDAR Beams Used**		**Metric**		**Relative change**	
	Train	**Test**	**mAP** ↑	**NDS** ↑	δ **mAP%** ↑	δ **NDS%** ↑
Performance Degradation						
$E_L + E_C$	32	32	0.642	0.685	0	0
$E_L + E_C$	32	8	0.391	0.516	-39.1	-24.7
E_L	32	32	0.582	0.653	0	0
E_L	32	8	0.176	0.395	-69.8	-39.5
Feature Mapper						
$E_L + E_C$	32	32	0.642	0.685	0	0
$\mathcal{M}_{C\to L} + E_C$	32	32	0.352	0.432	-45.2	-36.9
E_L	32	32	0.582	0.653	0	0
$\mathcal{M}_{C\to L}$	32	32	0.331	0.420	-43.1	-35.7
E_C	32	32	0.350	0.424	—	—
Adaptation Performance						
$E_L + E_C$	32	8	0.391	0.516	0	0
$E_{FM} + E_C$	8	8	0.368	0.482	-5.9	-6.6
$E_{PL} + E_C$	8	8	0.431	0.520	+10.2	+0.8
E_L	32	8	0.176	0.395	0	0
E_{FM}	8	8	0.186	0.365	+5.6	-7.6
E_{PL}	8	8	0.199	0.374	+13.1	-5.3

column shows the relative change in performance due to degradation. The Train and Test columns indicate the number of LiDAR beams used for training and evaluating, respectively. When evaluated on the non-degraded data (32 beams), the original LiDAR encoder E_L achieves a mAP of 0.642 and an NDS of 0.685 when combined with the camera encoder E_C. However, when evaluated on the degraded data (8 beams), the performance drops significantly to a mAP of 0.391 and an NDS of 0.516, which corresponds to a relative change of -29.1% in mAP and -24.7% in NDS. When using only the LiDAR encoder without the camera encoder, the performance degradation is even more pronounced, with a mAP of 0.176 and an NDS of 0.395 on the degraded data, corresponding to a relative change of -69.8% in mAP and -39.5% in NDS. These results highlight the significant impact of sensor degradation on the performance of UniBEV.

C.M.F.M. Looking again at table 2 we also display the results of the Cros-Modal Feature Mapper $\mathcal{M}_{C\to L}$. Under the section *Featur Mapper* we display the relevant method inference results. We can see that the performance of $\mathcal{M}_{C\to L}$ is significantly worse than the original LiDAR encoder E_L when both are evaluated on the non-degraded data (32 beams). When combined with the camera encoder E_C, $\mathcal{M}_{C\to L}$ achieves a mAP of 0.352 and an NDS of 0.432, which corresponds to a relative change of -45.2% in mAP and -36.9% in NDS compared to $E_L + E_C$. When using only the pseudo-LiDAR features generated by $\mathcal{M}_{C\to L}$ without the camera encoder, the performance drops further to a mAP of 0.331 and an NDS of 0.420, corresponding to a relative change of -43.1% in mAP and -35.7% in NDS compared to E_L only. These results indicate that while the

UNet-based pseudo-LiDAR generation can capture some useful information from the camera features, it is not able to fully reconstruct the LiDAR features, leading to a significant drop in detection performance. The performance of $\mathcal{M}_{C\to L}$ is also worse than using the camera encoder E_C alone, which suggests that the Feature Mapper may be introducing noise or artifacts that degrade the quality of the features for detection.

Looking at tables 3 and 4 we can investigate in which *way* our Feature Mapper $\mathcal{M}_{C\to L}$ is performing worse than the original LiDAR encoder E_L. First, in table 3 we can see that the performance of $\mathcal{M}_{C\to L}$ is significantly worse than E_L at a strict distance threshold of 0.5m, with an average relative drop of 82.2% across all classes. However, when we relax the distance threshold to 4.0m, the performance improves significantly, with an average relative drop of 11.42%. This indicates that while $\mathcal{M}_{C\to L}$ may be able to capture some coarse spatial information, it struggles to capture the precise spatial details necessary for accurate detection.

Table 4 further confirms this observation by showing that the true positive error metrics for $\mathcal{M}_{C\to L}$ are significantly worse than those for E_L across all metrics, with particularly large increases in translation and velocity errors. This suggests that the pseudo-LiDAR features generated by the UNet are not able to capture the fine-grained spatial and motion information that is crucial for accurate 3D object detection, leading to a significant degradation in detection quality compared to the original LiDAR features.

It is no surprise then, when combined with the camera modality, $\mathcal{M}_{C\to L}$ + E_C performs worse than E_L + E_C, which indicates that the pseudo-LiDAR features are not providing useful complementary information to the camera features, and may even be harming performance when combined. Overall, these results suggest that while the Feature Mapper can produce some features from the camera modality, it is not sufficient to match the performance of the original LiDAR features, and may require further refinement or additional constraints to improve its effectiveness.

CM-FKD and Camera Pseudo-Labeling. In table 2 under *Adaptation performance* we show the results for our two main methods. For the CM-FKD using the Cross-Modal Feature Mapper we denoted as E_{FM}. The method of using Camera Pseudo-Labeling to retrain the LiDAR encoder is denoted as E_{PL}. First we look at E_{FM}. When evaluated on the degraded data (8 beams), E_{FM} + E_C achieves a mAP of 0.368 and an NDS of 0.482, which corresponds to a relative change of -5.88% in mAP and -6.59% in NDS compared to E_L + E_C. When using only the retrained LiDAR encoder without the camera encoder, E_{FM} achieves a mAP of 0.186 and an NDS of 0.365, corresponding to a relative change of +5.6% in mAP and -7.6% in NDS compared to E_L only. These results indicate that while the retrained LiDAR encoder using the Feature Mapper E_{FM} is able to capture some useful information from the camera modality, it is not able to fully recover the performance of the original model without degradation. Specifically, the increase in mAP and decrease in NDS when using only E_{FM} suggests that the retrained LiDAR encoder is able to detect more objects, but the overall quality of those detections are worse. Furthermore, the generated feature maps from E_{FM} are not able to provide useful complementary information to the camera features, as the performance of E_{FM} + E_C is still worse than E_L + E_C.

The method where we use Camera Pseudo-Labeling to retrain the LiDAR encoder E_{PL} performs better than the Feature Mapper method E_{FM} in both settings (with and without the camera encoder). In the fusion setting E_{PL} + E_C achieves a mAP of 0.431 and an NDS of 0.520, which corresponds to a relative change of +10.2% in mAP and +0.8% in NDS compared to E_L + E_C. When using only the retrained LiDAR encoder without the camera encoder, E_{PL} achieves a mAP of 0.199 and an NDS of 0.374, corresponding to a relative change of +13.1% in mAP and -5.3% in NDS compared to E_L only. These results suggest that Camera Pseudo-Labeling is more effective at recovering the performance of the original model compared to the Feature Mapper method. However, it suffers from the same true positive quality issues. When evaluated independently, E_{PL} has worse NDS score. Through Fusion, the Camera Encoder E_C seems to provide additional benefits and offsets the true positive errors.

Looking at the true positive error metrics in table 5, we can see that both E_{FM} and E_{PL} have significantly worse translation and velocity errors compared to the degraded LiDAR encoder $E_{L,32\to 8}$. This suggests that both methods struggle to capture the fine-grained spatial and motion information necessary for accurate detection. However, E_{PL} has a smaller increase in mean relative drop, which explains its better performance in terms of NDS. Furthermore, looking at the AP by class at different distance thresholds in table 6, we can see that both methods perform worse than the degraded LiDAR encoder across all classes at a distance threshold of 0.5m, with particularly large drops for the less common classes. For example, a drop of -69.23% in mAP for the Motorcycle class for E_{FM} and a drop of -56.25% for E_{PL}. However, when we relax the distance threshold to 4.0m, the performance of both methods improves significantly, with E_{PL} even outperforming the degraded LiDAR encoder for the Barrier class. This suggests that while both methods struggle to capture precise spatial information, they are able to capture some coarse spatial information that can still be useful for detection at a larger distance threshold. Again, the Camera Pseudo-Labeling method E_{PL} seems to perform better than the Feature Mapper method E_{FM} across all classes and distance thresholds,

Table 3. **AP by class at two distance thresholds Feature Mapper.** AP for the original LiDAR encoder $E_{L,32\to32}$ and the camera-to-LiDAR mapper $\mathcal{M}_{C\to L}$ at 0.5m and 4.0m distance thresholds. Relative change is calculated with respect to $E_{L,32\to32}$.

Category	**Dist 0.5m AP**			**Dist 4.0m AP**		
	$E_{L,32\to32}$	$\mathcal{M}_{C\to L}$	**Relative change % ↑**	$E_{L,32\to32}$	$\mathcal{M}_{C\to L}$	**Relative change % ↑**
Car	0.749	0.198	-73.56	0.901	0.839	-6.88
Pedestrian	0.762	0.063	-91.73	0.862	0.703	-18.45
Barrier	0.485	0.157	-67.63	0.707	0.717	1.41
Motorcycle	0.537	0.033	-93.85	0.699	0.548	-21.6
Mean (all)	0.633	0.112	**-82.2**	0.792	0.701	**-11.42**

Table 4. **True Positive (TP) Error Analysis: Feature Mapper.** Higher values represent increased error (lower quality). With $E_{L,32\to32}$ as the original LiDAR encoder inference on 32 beam data and $\mathcal{M}_{C\to L}$ as the feature mapper.

Metric	$E_{L,32\to32}$	$\mathcal{M}_{C\to L}$	**Relative drop % ↓**
Translation (m)	0.327	0.747	**+128%**
Scale (IoU)	0.26	0.297	**+12.5%**
Orientation (rad)	0.321	0.409	**+27.4%**
Velocity (m/s)	0.292	0.792	**+171.2%**
Attribute (%)	0.178	0.202	**+13.5%**

which is consistent with the overall performance results in table 2.

4.2.2. Qualitative Results

Feature Maps. In figure 3 we visualize the intermediate feature maps produced for the three trained pipelines. The left column shows the ground truth feature maps, the middle column shows the input to the network and the right colum shows the output/prediction. With row one being the pipeline for training $\mathcal{M}_{C\to L}$, row two being the pipeline for training E_{FM} and row three being the pipeline for training E_{PL}. All feature maps are shown averaged across the channel dimension for better visualization.

For the Feature Mapper pipeline, we can see that the Feature Mapper is able to learn some features from the camera modality, but fails to produce high-quality features that are consistent with the original LiDAR features. The feature maps produced by the Feature Mapper show some activation in regions corresponding to objects in the scene, but they are much noisier and less distinct than the original LiDAR features. Since we are mapping from a camera modality to a LiDAR modality, we can expect there to be inherent limitations. We see that background activations are *washed* away, since the feature map of E_C has a lot of uncertainty in the background regions. The same is happening at places where the camera features are producing a *streak* of activations instead of a more focused activation. This

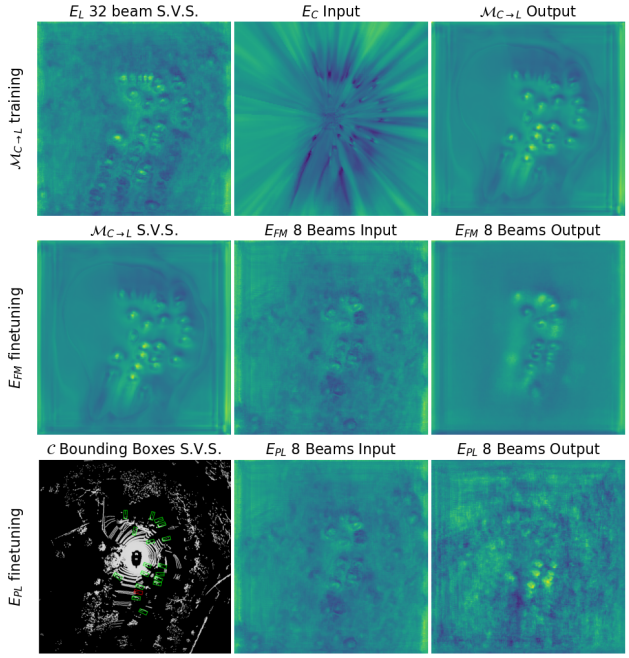

Figure 3. **Feature Maps of training pipelines.** In the left column we show the supervision signal. The middle column shows the input to the network and the right column shows the output/prediction. On the left side we name the corresponding pipelines. Feature maps are averaged across the channel dimension for visualization. Feature maps are scaled. High-intensity means there is high activation across all channels.

is consistent with the quantitative results, which show that E_{FM} is not able to match the performance of the original LiDAR features. This is further shown through the PCA projections of the feature spaces in figure 4. The streaks in the E_C features contain a lot of false positives, indicating that the camera features think there are objects in those regions but do not know the precise location. E_L knows the precise location, as seen from the corresponding activation with bounding boxes. However, E_{FM} is not able to learn the precise location and instead produces more sparse and less distinct features in those regions, which is likely con-

Table 5. **True Positive (TP) Error Comparison. Adapatation performance.** Higher values indicate larger error (lower quality). We compare the degraded LiDAR encoder $E_{L,32\to8}$ against the feature mapper E_{FM} and the pseudo-labeling encoder E_{PL}. Relative drop is computed with respect to $E_{L,32\to8}$.

Error metric ↓	$E_{L,32\to8}$	E_{FM}	**Relative drop (%)** ↓	E_{PL}	**Relative drop (%)** ↓
Translation (m)	0.438	0.580	**+32.42**	0.613	**+39.95**
Scale (IoU)	0.307	0.338	**+10.10**	0.317	**+3.26**
Orientation (rad)	0.582	0.579	**-0.52**	0.585	**+0.52**
Velocity (m/s)	0.401	0.553	**+37.91**	0.521	**+29.93**
Attribute (%)	0.200	0.237	**+18.50**	0.218	**+9**
Mean drop ↓			**+19.68**		**+16.53**

Table 6. **AP by class at two distance thresholds. Adaptation performance.** AP for the degraded LiDAR encoder $E_{L,32\to8}$, the feature mapper E_{FM}, and the pseudo-labeling encoder E_{PL} at 0.5m and 4.0m distance thresholds. Relative change is measured with respect to $E_{L,32\to8}$.

Category	**Dist 0.5m AP**					**Dist 4.0m AP**				
	$E_{L,32\to8}$	E_{FM}	**Rel. change %** ↑	E_{PL}	**Rel. change %** ↑	$E_{L,32\to8}$	E_{FM}	**Rel. change %** ↑	E_{PL}	**Rel. change %** ↑
Car	0.331	0.309	-6.65	0.268	-19.03	0.540	0.582	7.78	0.617	14.26
Pedestrian	0.297	0.184	-38.05	0.236	-20.54	0.386	0.420	8.81	0.470	21.76
Barrier	0.160	0.104	-35.00	0.070	-56.25	0.346	0.454	31.21	0.480	38.73
Motorcycle	0.026	0.008	-69.23	0.025	-3.85	0.050	0.063	26.00	0.103	106.00
Mean (all)	0.2035	0.15125	**-25.68**	0.14975	**-26.41**	0.3305	0.37975	**14.9**	0.4175	**26.32**

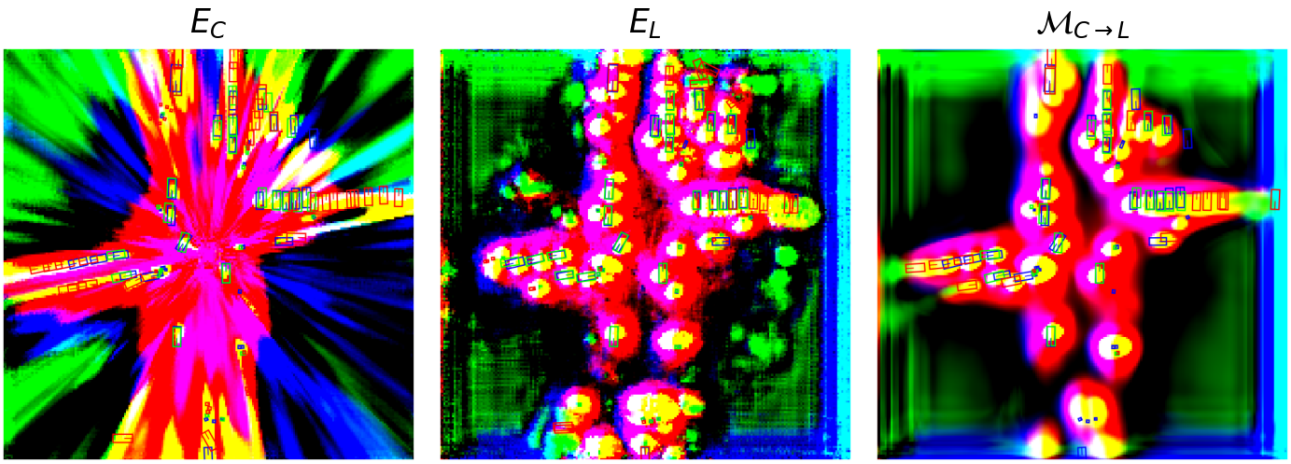

Figure 4. **Camera to LiDAR mapping limitations.** The left column shows the PCA projections of the Camera features, the middle column shows the PCA projections of the LiDAR features inferenced on 32 beam data, and the right column shows the PCA projections of the feature mapper features. Corresponding detections using only those features are shown overlayed on each feature map.

tributing to the performance drop. This makes sense, since we only provide the camera features as a supervision signal without anything else, so it becomes a very difficult task for the Feature Mapper to learn how to produce blobs along this uncertainty.

Following this, we can see that the retrained LiDAR encoder E_{FM} is able to produce features that are more similar to the original LiDAR features than the Feature Mapper generated pseudo-LiDAR features. This suggests that while the Feature Mapper based pseudo-LiDAR generation may not be able to produce high-quality features on its own, it is still able to capture some useful information from the camera modality that can guide the retraining of the LiDAR encoder to produce features that are more consistent with the original LiDAR features. However, as we can see in figure 3 at feature map E_{FM} *8 Beams Output*, some blobs are recovered but background information becomes more washed away. This is likely because the Feature Mapper in the first

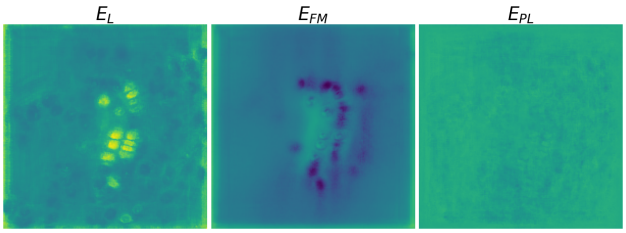

Figure 5. **Variance across channels.** The variance of activations across the channel dimension is shown for each feature map. The higher the intensity the higher the variance. Feature maps obtained through 8 beam inference.

phase, while able to learn a mapping for the LiDAR blobs, is not able to produce high-quality background features. Setting the Feature Mapper features as a supervision signal for retraining the LiDAR encoder, forces the retrained LiDAR encoder to also learn to produce these low-quality background features, which may be harming the performance of the retrained LiDAR encoder. This is consistent with the quantitative results, which show that E_{FM} performs worse than E_L and E_{PL} in terms of NDS, which takes into account the true positive quality.

Finally we observe that E_{PL} at feature map E_{PL} *8 Beams Output* is able to produce features with more background information, which is likely because the pseudo-labeling method provides a stronger supervision signal that encourages the retrained LiDAR encoder to produce features that are more similar to the original LiDAR features, including the background features. This is consistent with the quantitative results, which show that E_{PL} performs better than E_{FM} in terms of NDS, suggesting that the higher quality features produced by E_{PL} are providing more useful complementary information to the camera features, leading to better overall performance when combined.

However, when we look at figure 5 we can see that the variance of activations across the channel dimension. E_L we can see clear intensities at locations, meaning that the model is able to produce distinct features across the channels that are likely useful for the detection head to distinguish between different objects and background. Meaning it has high certainty that an object is there.

However, for E_{FM} we can see that the variance is much lower across the channels, meaning that the model is producing more similar features across the channels. This is likely because the model is not able to learn how to produce distinct features across the channels due to the low-quality supervision signal provided by the Feature Mapper. This could be contributing to the performance drop of E_{FM} compared to E_L, since the detection head may be relying on these distinct features across the channels to make accurate detections.

For E_{PL} we can see that the variance around the background is more aligned with what E_L produces, which is likely because the pseudo-labeling method provides a stronger supervision signal that encourages the retrained LiDAR encoder to produce features that are more similar to the original LiDAR features, including the background features. However, we observe no high intensities at the bounding box locations in the center. This is likely because the pseudo-labeling method is making the retrained LiDAR encoder more sensitive to activations across the whole feature space, which explains the high false positive rate of E_{PL} which can be seen in figure 6.

This is consistent with the quantitative results, which show that E_{PL} performs worse than E_L in terms of NDS, but still leading to better overall performance than E_{FM} when combined. This is likely because even though the features produced by E_{PL} have lower variance across the channels, they are still more similar to the original LiDAR features than those produced by E_{FM}, as seen in the feature map visualization in figure 3. Which means that when combined with the camera features, the features produced by E_{PL} are still able to provide useful complementary information that leads to better overall performance compared to E_{FM}.

Detections and PCA. Looking at the detection results and PCA projections of the feature spaces for the two pipelines in figure 6, we can see the detection results for every model overlayed on the PCA projections of the feature spaces. The PCA projections are fitted on the original 32 beam LiDAR features with 3 components and overlayed with the detected bounding boxes of corresponding models. This allows us to visually inspect how well the different feature representations are aligned with the original LiDAR features, and how this alignment (or lack thereof) may be affecting the detection result. The model names are shown in the title of each subfigure, for example $E_{L,32\to8}$ means the degraded LiDAR encoder trained on 32 beam data and inferenced on 8 beam data.

We can see in the detections row how E_L performs when we move from 32 beam data to 8 beam data, and how the performance drops significantly. Multiple objects are missing, namely the cars in the far right and smaller objects in the bottom center. When we look at the PCA projections, we can see that the features of $E_{L,32\to32}$ are well-clustered and distinct, with less activations on the degraded method $E_{L,32\to8}$. Specifically, activations are not found anymore at the boundingboxes on the far right. This is likely contributing to the performance drop of $E_{L,32\to8}$ compared to $E_{L,32\to32}$, since the detection head may be relying on these distinct features in the PCA space to make accurate detections.

When we look at the Feature Mapper method E_{FM} and the Camera Pseudo-Labeling method E_{PL}, we can see that both methods in the zoomed area are able to recover some

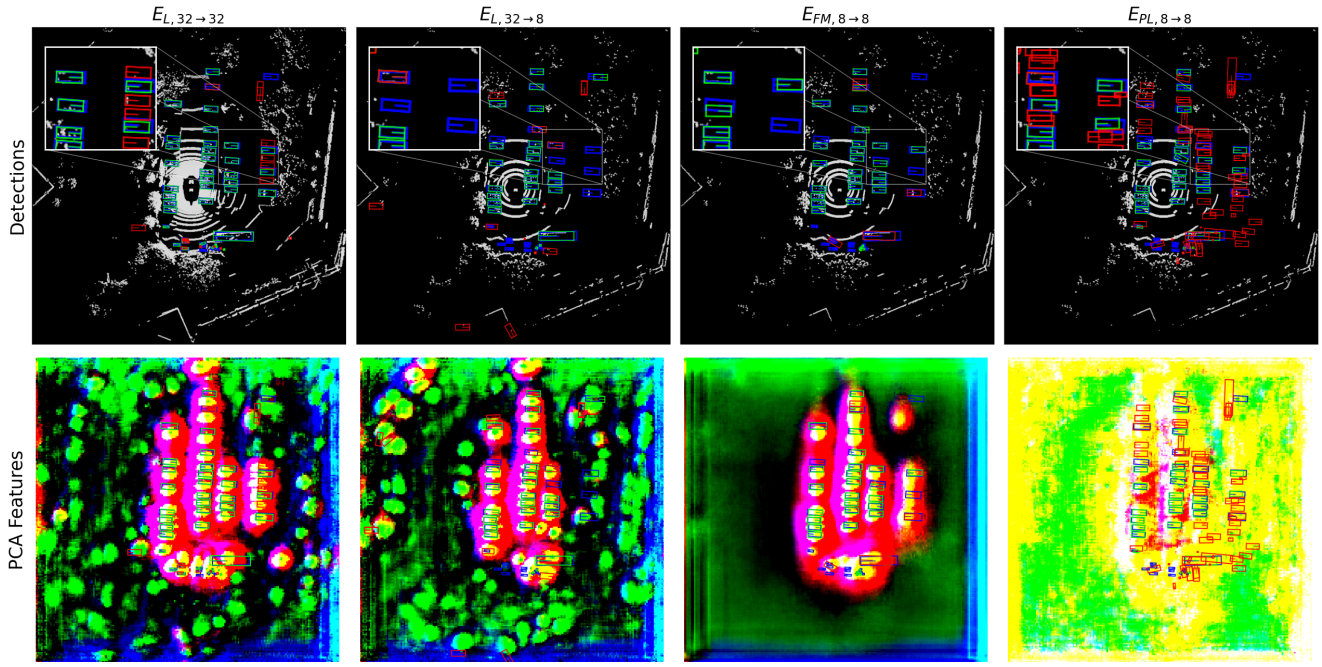

Figure 6. **Detections and PCA projections.** We display the detection results for each model overlayed on the PCA projections of the feature spaces and overlayed on the corresponding beam data.The bounding box colors are defined as follows, is the True Positive, is the False Positive, and is the Ground Truth.

of the missing detections compared to $E_{L,32\to8}$, but they still perform worse than $E_{L,32\to32}$. Specifically, they did not recover all objects such as the smaller objects in the bottom center. When we look at the pca projections, we can see that the features of E_{FM} and E_{PL} are not well-aligned with the original LiDAR features (E_L) in the PCA space. E_{FM} has less distinct background features, but was able to recover activations at bounding boxes, albeit washed out. E_{PL} has an interesting PCA projection. Through training with Pseudo Labels, it seems like it made the encoder more sensitive to activations through the whole feature space, as indicated by the yellow regions. This corresponds to E_{PL} having a high false positive rate, since the model is producing a lot of activations across the feature space, which may be harming the performance of E_{PL} compared to E_L. However, it is able to recover more distinct features at the bounding boxes compared to E_{FM}, which may explain why it performs better than E_{FM} in terms of NDS and mAP. Overall, when we look at the PCA projections for both methods, they are not well-aligned with the original LiDAR features, which is consistent with the quantitative results showing that both methods perform worse than the original LiDAR features NDS wise, which takes into account the true positive quality.

5. Conclusion

In this work, we showed that multi-modal 3D object detection models are highly sensitive to asymmetric sensor degradation: under LiDAR degradation from 32 to 8 beams without adaptation, performance dropped by -69.8% in mAP and -39.5% in NDS, highlighting the sensitivity of multimodal 3D detectors to sensor quality shifts when evaluated on the nuScenes dataset [4].

We tested two adaptation strategies to recover performance: pseudo-labeling and cross-modal feature mapping, both using the unaffected camera modality as reference. Inferencing with only the LiDAR modality, the pseudo-labeling approach provided a +13.1% mAP improvement over the no-adaptation baseline, while the cross-modal feature mapping approach provided a +5.6% mAP improvement. However, the NDS score of the pseudo-labeling approach decreased by -5.3% compared to the no-adaptation baseline, while the NDS score of the cross-modal feature mapping approach decreased by -7.6%, suggesting that both approaches may have introduced noise that hurt overall detection quality, and that the pseudo-labeling approach may have provided more useful supervisory signals for improving localization performance.

Overall, cross-modal adaptation provides partial recovery under asymmetric sensor degradation, but both approaches struggle to preserve the fine-grained geometric in-

formation required for accurate localization. The feature-level distillation approach appears particularly limited by the difficulty of reconstructing LiDAR-like BEV structure from camera features alone, whereas pseudo-labeling provides stronger task-level supervisory signals.

Limitations This study has several limitations. The dataset splits are not fully separated. In ideal scenario, we would have a pre-train split, a split for retraining the LiDAR encoder and a validation split. In our work, the pre-train split and the split for retraining the LiDAR encoder contained the same samples (albeit with different LiDAR beams).

We only evaluated one degradation setting (32-beam to 8-beam LiDAR), so the findings may not directly generalize to other degradation sources such as weather effects, calibration drift, or sensor noise.

The adaptation pipeline relies on one mapping design (Attention UNet) trained with a simple reconstruction objective, which likely under-constrains the fine-grained geometric detail needed for accurate localization. Fourth, the evaluation of the mapping function was only done with a non-corrupted camera modality, so it is unclear how well the mapping would perform if the reference modality also degrades or is affected by sensor degradation in the target domain.

Additionally, the employed attention mechanism may be insufficient for capturing the complex channel-wise and spatial relationships required for precise BEV feature reconstruction.

Future Work Future work should explore a wider range of degradation types and severities, and evaluate the mapping function under camera degradation in the target domain to test its robustness when the reference modality is also affected by domain shift.

On the modeling side, a promising approach would be to combine both pseudo-labeling and cross-modal feature mapping. Using the pseudo-labeling approach to generate supervisory signals for the degraded modality could help to improve the quality of the mapped features, while the cross-modal feature mapping could provide additional spatial alignment and context that may be missing from the pseudo-labels alone.

Additionally, future work could explore more sophisticated architectures for the mapping function that better preserve high-frequency BEV structure and local details, such as using multi-scale feature fusion or incorporating temporal information to capture motion cues.

Finally, future work should also explore training the mapping function with source domain labels to provide stronger supervision for learning the cross-modal feature mapping, which may help to improve the quality of the mapped features and lead to better adaptation performance in the target domain.

Acknowledgements

I would like to thank my daily supervisor S. Wang for his guidance and support throughout this project. I would also like to thank Dr. J.F.P. Kooij for his valuable feedback and insights that helped me throughout the project.

References

[1] Waymo - Self-Driving Cars - Autonomous Vehicles - Ride-Hail. https://waymo.com/. 2

[2] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection With Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1090–1099, 2022. 3

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 2

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 2, 3, 8, 15

[5] Hongxiang Cai, Zeyuan Zhang, Zhenyu Zhou, Ziyin Li, Wenbo Ding, and Jiuhua Zhao. BEVFusion4D: Learning LiDAR-Camera Fusion Under Bird's-Eye-View via Cross-Modality Guidance and Temporal Aggregation, 2023. 3

[6] Gyusam Chang, Wonseok Roh, Sujin Jang, Dongwook Lee, Daehyun Ji, Gyeongrok Oh, Jinsun Park, Jinkyu Kim, and Sangpil Kim. CMDA: Cross-Modal and Domain Adversarial Adaptation for LiDAR-Based 3D Object Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(2):972–980, 2024. 2, 4, 8

[7] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinhong Jiang, and Feng Zhao. BEVDistill: Cross-Modal BEV Distillation for Multi-View 3D Object Detection, 2022. 3, 8

[8] Zhiyu Chong, Xinzhu Ma, Hong Zhang, Yuxin Yue, Haojie Li, Zhihui Wang, and Wanli Ouyang. MonoDistill: Learning Spatial Features for Monocular 3D Object Detection, 2022. 3

[9] Mmdetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 9

[10] Yinpeng Dong, Caixin Kang, Jinlai Zhang, Zijian Zhu, Yikai Wang, Xiao Yang, Hang Su, Xingxing Wei, and Jun Zhu. Benchmarking Robustness of 3D Object Detection to Common Corruptions. In *Proceedings of the IEEE/CVF Con-*

ference on Computer Vision and Pattern Recognition, pages 1022–1032, 2023. 2, 3

[11] Chongjian Ge, Junsong Chen, Enze Xie, Zhongdao Wang, Lanqing Hong, Huchuan Lu, Zhenguo Li, and Ping Luo. MetaBEV: Solving Sensor Failures for 3D Detection and Map Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8721–8731, 2023. 2, 3

[12] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge Distillation: A Survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021. 3

[13] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. 1

[14] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. BEVDet: High-performance Multi-camera 3D Object Detection in Bird-Eye-View, 2022. 3

[15] Yihao Huang, Kaiyuan Yu, Qing Guo, Felix Juefei-Xu, Xiaojun Jia, Tianlin Li, Geguang Pu, and Yang Liu. Improving Robustness of LiDAR-Camera Fusion Model against Weather Corruption from Fusion Strategy Perspective, 2024. 1, 2, 3

[16] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robo3D: Towards Robust and Reliable 3D Perception against Corruptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023. 3

[17] Shuangzhi Li, Zhijie Wang, Felix Juefei-Xu, Qing Guo, Xingyu Li, and Lei Ma. Common Corruption Robustness of Point Cloud Detectors: Benchmark and Enhancement. *IEEE Transactions on Multimedia*, 27:848–859, 2023. 1, 3

[18] Zhenyu Li, Zehui Chen, Ang Li, Liangji Fang, Qinhong Jiang, Xianming Liu, and Junjun Jiang. Unsupervised Domain Adaptation for Monocular 3D Object Detection via Self-training. In *Computer Vision – ECCV 2022*, pages 245–262, Cham, 2022. Springer Nature Switzerland. 2, 4

[19] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L. Rus, and Song Han. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2781, London, United Kingdom, 2023. IEEE. 2, 3

[20] Yunze Man, Liangyan Gui, and Yu-Xiong Wang. DualCross: Cross-Modality Cross-Domain Adaptation for Monocular BEV Perception. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10910–10917, Detroit, MI, USA, 2023. IEEE. 2, 4, 8

[21] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y. Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention U-Net: Learning Where to Look for the Pancreas, 2018. 7, 9, 1

[22] Jasmine Richter, Florian Faion, Di Feng, Paul Benedikt Becker, Piotr Sielecki, and Claudius Glaeser. Understanding the Domain Gap in LiDAR Object Detection Networks. 2022. 3

[23] Cristiano Saltori, Stephane Lathuiliere, Nicu Sebe, Elisa Ricci, and Fabio Galasso. SF-UDA^{3D} : Source-Free Unsupervised Domain Adaptation for LiDAR-Based 3D Object Detection. In *2020 International Conference on 3D Vision (3DV)*, pages 771–780, Fukuoka, Japan, 2020. IEEE. 2, 4

[24] Ziying Song, Lin Liu, Feiyang Jia, Yadan Luo, Caiyan Jia, Guoxin Zhang, Lei Yang, and Li Wang. Robustness-Aware 3D Object Detection in Autonomous Driving: A Review and Outlook. *IEEE Transactions on Intelligent Transportation Systems*, 25(11):15407–15436, 2024. 1, 3

[25] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 2

[26] Guo-Hua Wang, Yifan Ge, and Jianxin Wu. Distilling Knowledge by Mimicking Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 3

[27] Shiming Wang, Holger Caesar, Liangliang Nan, and Julian F. P. Kooij. UniBEV: Multi-modal 3D Object Detection with Uniform BEV Encoders for Robustness against Missing Sensor Modalities. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 2776–2783, Jeju Island, Korea, Republic of, 2024. IEEE. 2, 3, 8, 9

[28] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q. Weinberger, and Wei-Lun Chao. Train in Germany, Test in the USA: Making 3D Object Detectors Generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020. 2, 4

[29] Ziteng Xue, Mingzhe Guo, Heng Fan, Shihui Zhang, and Zhipeng Zhang. CorrBEV: Multi-View 3D Object Detection by Correlation Learning with Multi-modal Prototypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27413–27423, 2025. 3

[30] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: Self-Training for Unsupervised Domain Adaptation on 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378, 2021. 2, 4

[31] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D++: Denoised Self-Training for Unsupervised Domain Adaptation on 3D Object Detection. *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, pages 1–17, 2022. 2, 4

[32] Ming Da Yang. Showcasing worsened prediction accuracy of the UniBEV L+C model due to LiDAR or camera sensor degradation, 2024. 2, 3

[33] Kaicheng Yu, Tang Tao, Hongwei Xie, Zhiwei Lin, Tingting Liang, Bing Wang, Peng Chen, Dayang Hao, Yongtao Wang, and Xiaodan Liang. Benchmarking the Robustness of LiDAR-Camera Fusion for 3D Object Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3188–3198, 2023. 3

[34] Yun Zhao, Zhan Gong, Peiru Zheng, Hong Zhu, and Shaohua Wu. SimpleBEV: Improved LiDAR-Camera Fusion Architecture for 3D Object Detection, 2024. 3

[35] Shengchao Zhou, Weizhou Liu, Chen Hu, Shuchang Zhou, and Chao Ma. UniDistill: A Universal Cross-Modality Knowledge Distillation Framework for 3D Object Detection in Bird's-Eye View. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5116–5125, 2023. 3

Unsupervised Domain Adaptation for Multi-Modal 3D Object Detection under Asymmetric Sensor Degradation

Supplementary Material

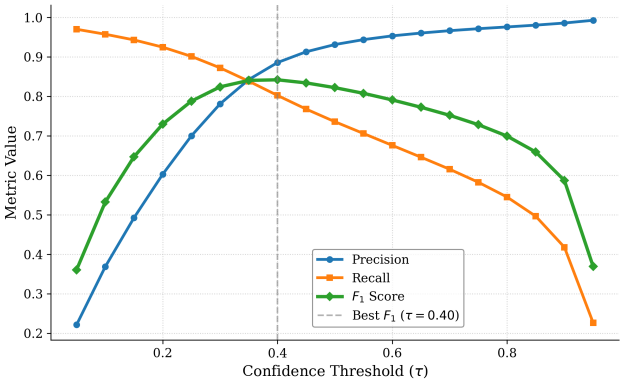

Figure 7. **Performance of pseudo-labeling approach across different confidence thresholds.** We experimented with different confidence thresholds, and found that a threshold of 0.4 provided the best balance between precision and recall for the pseudo-labels, leading to improved performance during fine-tuning of the degraded modality encoder.

6. Camera-Pseudo Labeling

6.1. Confidence Threshold selection

We experiments with the following confidence thresholds **[0.05 0.1 0.15 0.2 0.25 0.3 0.35 0.4 0.45 0.5 0.55 0.6 0.65 0.7 0.75 0.8 0.85 0.9 0.95]**. We calculated the precision and recall of the pseudo-labels generated by the pre-trained detector on the validation set for each confidence threshold. We found that a confidence threshold of 0.4 provided the best balance between precision and recall, thus the highest F1-score, leading to improved performance during fine-tuning of the degraded modality encoder. The performance of the pseudo-labeling approach across different confidence thresholds is shown in Fig. 7.

7. Cross-Modal Feature Mapper

7.0.1. LiDAR degradation

To simulate the degradation of LiDAR data, we apply a beam reduction technique to the original 32-beam LiDAR data. This involves selecting a subset of the beams to create a pseudo-LiDAR representation that mimics the characteristics of lower-density LiDAR sensors. The selection of beams is done in a way that maintains the spatial distribution of points while reducing the overall point cloud density. We apply this beam reduction technique in an online manner during training, allowing the model to learn from both the original and degraded LiDAR data. An algorithmic description of the beam reduction process is provided in Algorithm 1. The beam selection is handled by using the numpy library command `linspace` [13].

Algorithm 1: Spaced LiDAR Beam Reduction

Input: Point Cloud $\mathcal{P} \in \mathbb{R}^{N \times D}$ with D dimensions of $(x, y, z, \text{intensity}, \text{beam\_id})$, Total sensor beams N_{total}, Number of beams to drop N_{drop}
Output: Reduced Point Cloud $\mathcal{P}_{reduced}$
beam_id $\leftarrow$ sort_by_beam_id($\mathcal{P}$) `// Sort points by their beam ID`
$\mathcal{I}_{drop} \leftarrow \text{linspace}(0, N_{total} - 1, N_{drop})$ `// Calculate indices to remove`
foreach $id \in \mathcal{I}_{drop}$ **do**
  $\mathcal{P} \leftarrow \{p \in \mathcal{P} \mid \text{beam\_id}(p) \neq id\}$ `// Filter points by beam ID`
end
return $\mathcal{P}$

An example of the result of this algorithm is shown in Fig. 8, where we can see the original 32-beam LiDAR point cloud and the resulting 8-beam point cloud after applying the beam reduction technique.

7.1. UNet Architecture

An Attention UNet [21] architecture is adopted for the Cross-Modal Feature Mapper, which consists of an encoder-decoder structure with skip connections. Attention mechanisms are integrated into the skip connections to enhance the model's ability to focus on relevant features dur-

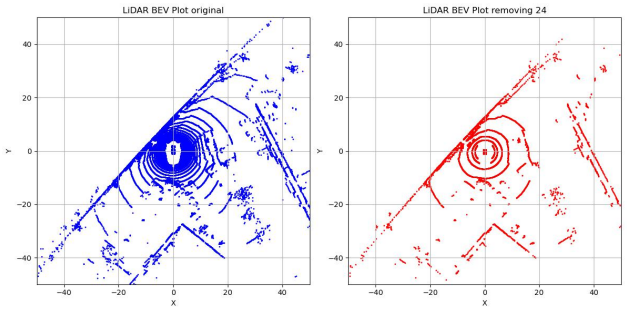

Figure 8. **Example of beam reduction technique.** The left image shows the original 32-beam LiDAR point cloud, while the right image shows the resulting 8-beam point cloud after applying the beam reduction technique.

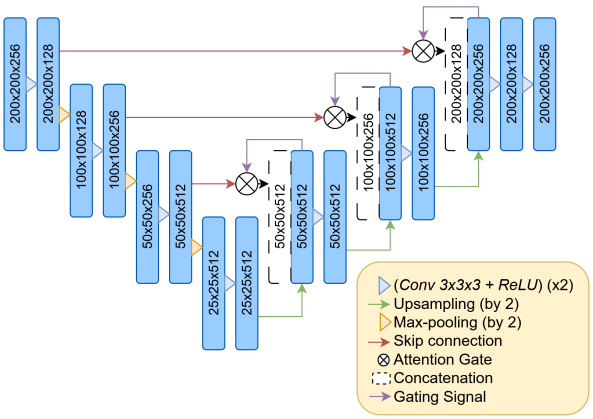

Figure 9. Architecture of the UNet-based Cross-Modal Feature Mapper. The model consists of an encoder-decoder structure with attention mechanisms integrated into the skip connections.

ing the feature mapping process. The UNet is designed to take in BEV image feature maps and output pseudo-LiDAR feature maps that closely resemble those generated by the original LiDAR encoder. The attention mechanism takes as input the *upsampled* feature maps from the decoder and the corresponding feature maps from the encoder, and generates an attention map that highlights the most relevant spatial regions for reconstruction. Important to note is that the attention map is generated for all channels, meaning we only have one attention map per layer, per sample.

In figure 9 we have drawn an architectural diagram of the UNet trained in this paper. The UNet has 4 levels of downsampling and upsampling. We sample the BEV image feature maps from W×H = 200×200 down in the following order: "200×200" → "100×100" → "50×50" → "25×25". The final output is upsampled back to "200×200" to match the original BEV feature map size. We have chosen the channel expansion as follows: 128 → 256 → 512 → 512 for the encoder, and the decoder mirrors this structure in reverse.

Training is done using AdamW optimizer with learning rate of 2e-3 and weight decay of 1e-1. We used CosineAnnealing as the learning rate scheduler, with warmup ratio 1e-4. The model is trained till convergence, thus for 5 epochs with a batch size of 5 on 2 Tesla V100-SXM2-32GB GPUs. The loss function used is the MSE loss between the UNet-generated pseudo-LiDAR features and the original LiDAR features extracted by the pre-trained encoder as seen in Equation 17.

$$\mathcal{J}_{\text{map}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \|\hat{\mathcal{L}}(E_C(x^C)) - E_L(x^L)\|^2 \qquad (17)$$

7.2. PTS Encoder retraining

For the PTS encoder retraining, we initialize the model with the weights from the original 32-beam LiDAR encoder E_L. The training is conducted on the degraded 8-beam LiDAR data, allowing the model to adapt to the new input distribution. The training is done using AdamW optimizer with a learning rate of 1e-4 and weight decay of 1e-1. We use CosineAnnealing as the learning rate scheduler with a warmup ratio of 0.1. We used a batch size of 5, with convergence reached after 26 epochs. The training is performed on 5 Tesla V100-SXM2-32GB GPUs. The loss function used for retraining is the MSE loss as seen in Equation 18 between the features extracted by the retrained encoder and the features outputted by the UNet.

$$\mathcal{J}_{\text{retrain}} = \frac{1}{N}\sum_{i=1}^{N} \|E_L(z^L) - \hat{\mathcal{L}}(E_C(z^C))\|^2 \qquad (18)$$