Unified Domain Generalization and Adaptation for Multi-View 3D Object Detection

Gyusam Chang[1*]   **Jiwon Lee**[2*]   **Donghyun Kim**[1]   **Jinkyu Kim**[1]
Dongwook Lee[2]   **Daehyun Ji**[2]   **Sujin Jang**[2†]   **Sangpil Kim**[1†]

[1]Korea University
[2]Samsung Advanced Institute of Technology
{gsjang95, d_kim, jinkyukim, spk7}@korea.ac.kr
{ji1.lee, dw12.lee, derek.ji, s.steve.jang}@samsung.com

Abstract

Recent advances in 3D object detection leveraging multi-view cameras have demonstrated their practical and economical value in various challenging vision tasks. However, typical supervised learning approaches face challenges in achieving satisfactory adaptation toward unseen and unlabeled target datasets (*i.e.*, direct transfer) due to the inevitable geometric misalignment between the source and target domains. In practice, we also encounter constraints on resources for training models and collecting annotations for the successful deployment of 3D object detectors. In this paper, we propose Unified Domain Generalization and Adaptation (UDGA), a practical solution to mitigate those drawbacks. We first propose Multi-view Overlap Depth Constraint that leverages the strong association between multi-view, significantly alleviating geometric gaps due to perspective view changes. Then, we present a Label-Efficient Domain Adaptation approach to handle unfamiliar targets with significantly fewer amounts of labels (*i.e.*, 1% and 5%), while preserving well-defined source knowledge for training efficiency. Overall, UDGA framework enables stable detection performance in both source and target domains, effectively bridging inevitable domain gaps, while demanding fewer annotations. We demonstrate the robustness of UDGA with large-scale benchmarks: nuScenes, Lyft, and Waymo, where our framework outperforms the current state-of-the-art methods.

1 Introduction

3D Object Detection (3DOD) is a pivotal computer vision task in various real-world applications such as autonomous driving and robotics. Recent progress in 3DOD [1–4] have showcased remarkable advancements, primarily due to the large-scale benchmark datasets [5–7] and the introduction of multiple computer vision sensors (*e.g.*, LiDAR, multi-view cameras, and RADAR). Among these, camera-based multi-view 3DOD [8–12] has drawn significant attention for its cost-efficiency and rich semantic information. However, a significant challenge remains largely unexplored: accurately detecting the location and category of objects in the presence of distributional shifts between the source and target domains (*i.e.*, data distributional gaps between the training and the testing datasets).

To successfully develop and deploy Multi-view 3DOD models, we need to solve two practical problems: (1) the geometric distributional shift across different sensor configurations, and (2) the limited amount of resources (*e.g.*, insufficient computing resources, expensive data annotations). The first problem poses a challenge in learning transferable knowledge for robust generalization in novel

---

*These authors contributed equally.
†Corresponding authors.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

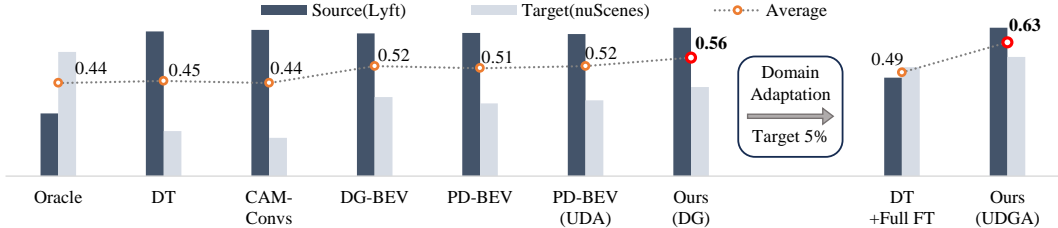



Figure 1: Comparison of performance in both source and target domains (Tab. 6). Here, "Average" (orange dots) refers to mean NDS in both the source and target domains. We draw comparisons with prior methods CAM-Conv [13], DG-BEV [14] and PD-BEV [15] offering an empirical lower and upper bounds, DT and Oracle. Note that we only use 5% of the target label for Domain Adaptation.

domains. The second issue inevitably requires efficient utilization of computing resources for training and inference, as well as label-efficient development of 3DOD models in practice. To tackle these practical problems, we introduce a **U**nified **D**omain **G**eneralization and **A**daptation (UDGA) strategy, which addresses a series of domain shift problems (*i.e.*, learning domain generalizable features significantly improves the quality of parameter- and label-efficient few-shot domain adaptation).

Prior studies aim to learn domain-agnostic knowledge alleviating domain shifts from drastic view changes in cross-domain environments. DG-BEV [14] disentangles the camera intrinsic parameters and trains the network with a domain discriminator for view-invariant feature learning. Similarly, PD-BEV [15] renders implicit foreground volumes and suppresses the perspective bias leveraging semantic supervision. However, these approaches struggle to capture optimal representations, highlighting that there is still room for improvements in novel target domains (*i.e.*, up to -50.8% Closed Gap compared to Oracle). To tackle these drawbacks, we first advocate a Multi-view Overlap Depth Constraint that leverages occluded regions between adjacent views, which serve as notable triangular clues to guarantee geometric consistency. This approach effectively addresses perspective differences between cross-domain environments by directly penalising the corresponding depth between adjacent views, and shows considerable generalization capacity (up to +75.8% Closed Gap compared to DT).

Nevertheless, the development of algorithms running on edge devices (*i.e.*, autonomous vehicles) faces the challenge of limited resources, which requires efficient utilization of computing systems. To resolve these challenges, we carefully design a *go-to* strategy, Label-Efficient Domain Adaptation, that bridges two different domains with cost-effective transfer learning. Precisely, motivated by Parameter-Efficient Fine-Tuning (PEFT) [16–18], we focus on smooth adaptation to target domains by fully exploiting well-defined source knowledge. Specifically, leveraging plug-and-play extra parameters, we substantially adapt to target domains while retaining information from the source domain (+14% Average gain compared to DT+Full FT as shown in Fig. 1). As a result, we note that UDGA practically expand base models, efficiently boosting overall capacity under limited resources.

Given landmark datasets in 3DOD, nuScenes [6], Lyft [7] and Waymo [5], we validate the effectiveness of our UDGA framework for the camera-based multi-view 3DOD task. Notably, we achieve state-of-the-art performance in cross-domain environments and demonstrate the component-wise effectiveness through ablation studies. To summarize, our main contributions are as follows:

- We introduce the Unified Domain Generalization and Adaptation (UDGA) framework, which aims to learn generalizable geometric features and improve resource efficiency for enhanced practicality in addressing distributional shift alignments.

- We advocate depth-scale consistency across multi-view images to effectively address 3D geometric misalignment problems. To this end, we leverage the corresponding triangular cues between adjacent views to seamlessly bridge the domain gap.

- We present a label- and parameter-efficient domain adaptation method, which requires fewer annotations and fine-tuning parameters while preserving source-domain knowledge.

- We demonstrate the effectiveness of UDGA on multiple challenging cross-domain benchmarks (*i.e.*, Lyft → nuScenes, nuScenes → Lyft, and Waymo → nuScenes). The results show that UDGA achieves a new state-of-the-art performance in Multi-view 3DOD.

2

2 Related Work

2.1 Multi-view 3D Object Detection

3D object detection [4, 19, 1, 20–26] is a fundamental aspect of computer vision tasks in the real world. Especially, Multi-view 3D Object Detection leveraging Bird's Eye View (BEV) representations [11, 12, 8] have rapidly expanded. We observe that this paradigm is divided into two categories: (i) LSS-based [27, 11, 12], and (ii) Query-based [8, 28, 10]. The former adopts explicit methods leveraging depth estimation network, and the latter concentrates on implicit methods utilizing the attention mechanism of Transformer [29]. Recently, these methods [9, 30, 31] significantly benefit from improved geometric understanding leveraging temporal inputs. Also, methods [32–35] that directly guide the model using the LiDAR teacher model significantly encourage BEV spatial details. In particular, this approach is being adopted to gradually replace LiDAR in real-world scenarios; however, it still suffers from poor generalizability due to drastic domain shifts (*e.g.*, weather, country, and sensor). To mitigate these issues, we present a novel paradigm, Unsupervised Domain Generalization and Adaptation (UDGA), that effectively addresses geometric issues leveraging multi-view triangular clues and smoothly bridge differenet domains without forgetting previously learned knowledge.

2.2 Bridging the Domain Gap for 3D Object Detection

Due to the expensive cost of sophisticated sensor configurations and accurate 3D annotations for autonomous driving scenes, existing works strive to generalize 3D perception models in various data distributions. Specifically, they often fail to address the covariate shift between the training and test splits. To bridge the domain gap, existing approaches have introduced noteworthy solutions as below.

LiDAR-based. Wang *et al.* [36] introduced Statistical Normalization (SN) to mitigate the differences in object size distribution across various datasets. ST3D [37] leveraged domain knowledge through random object scale augmentation, and their self-training pipeline refined the pseudo-labels. SPG [38] aims to capture the spatial shape, generating the missing points. 3D-CoCo [39] contrastively adjust the domain boundary between source and target to extract robust features. LiDAR Distillation [40] generates pseudo sparse point sets in spherical coordinates and aligns the knowledge between source and pseudo target. STAL3D [41] effectively extended ST3D by incorporating adversarial learning. DTS [42] randomly re-sample the beam and aim to capture the cross-density between student and teacher models. CMDA [2] aim to learn rich-semantic knowledge from camera BEV features and adversarially guide seen sources and unseen targets, achieving state-of-the-art UDA capacity.

Camera-based. While various groundbreaking methods based on LiDAR have been researched, camera-based approaches are still limited. Due to the elaborate 2D-3D alignment, not only are LiDAR-based approaches not directly applicable, but conventional 2D visual approaches [43–46] cannot be adopted either. To mitigate these issues, STMono3D [47] self-supervise the monocular 3D detection network in a teacher-student manner. DG-BEV [14] adversarially guide the network from perspective augmented multi-view images. PD-BEV [15] explicitly supervise models by the RenderNet with pseudo labels. However, camera domain generalization methods cannot meet the performance required for the safety, struggling to address the practical domain shift in the perspective change. To narrow the gap, we introduce a Unified Domain Generalization and Adaptation (UDGA) framework that effectively promotes depth-scale consistency by leveraging occluded clues between adjacent views and then seamlessly transfers the model's potential along with a few novel labels.

2.3 Parameter Efficient Fine-Tuning

Recent NLP works fully benefit from general-purpose Large-language Models (LLM). Additionally, they have proposed Parameter-Efficient Fine-Tuning (PEFT) [17, 16, 48–50] to effectively transfer LLM power to various downstream tasks. Specifically, PEFT preserves and exploits previously learned universal information, fine-tuning only additional parameters with a few downstream labels. This paradigm enables to notably reduce extensive computational resources, and large amounts of task-specific data and also effectively address challenging domain shifts in various downstream tasks as reported by [51]. Inspired by this motivation, to address drastic perspective shifts between source and target domains, we design Label-Efficient Domain Adaptation that fully transfers generalized source potentials to target domains by fine-tuning only our extra modules with few-shot target data.

3

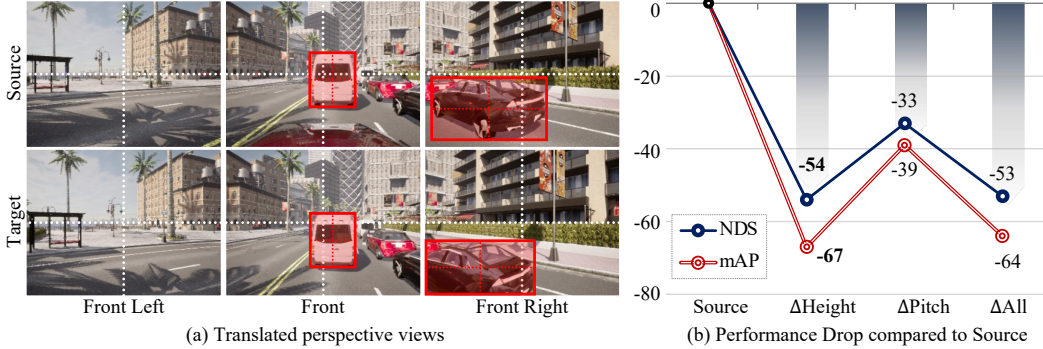



Figure 2: (a) An illustration of multi-view installation translation difference. The first (*i.e.*, source) and second (*i.e.*, target) rows are two perspective views of the same scene captured from different installation points. The translation gap between these views is substantial, approximately 30%. (b) Source trained network shows poor perception capability in target domain, primarily due to extrinsic shifts. In ΔHeight, mAP and NDS have dropped up to -67% compared to source. Note that we simulate the camera extrinsic shift leveraging CARLA [52] (refer to Appendix A for further details).

3 Methodology

3.1 Preliminary

Multi-view 3D Object Detection is a fundamental computer vision task that involves safely localizing and categorizing objects in a 3D space exploiting 2D visual information from multiple camera views. Especially, recent landmark Multi-view 3D Object Detection models [8, 10, 9, 11, 33] are formulated as follow; $\arg\min \mathcal{L}(Y, \mathcal{D}(\mathcal{V}(I, K, T)))$, where Y represents the size (l, w, h), centerness (cx, cy, cz), and rotation ϕ of each 3D object. Also, $I = \{i_1, i_2, ..., i_n\} \in \mathbb{R}^{N \times H \times W \times 3}$, K, and $T = [R|t]$ denotes multi-view images, intrinsic and extrinsic parameters. Specifically, these models, which fully benefit from view transformation modules $\mathcal{V}$, encode 2D visual features alongside the 3D spatial environment into a bird's eye view (BEV) representation. First, these works adopts explicit methods (BEV view transformation $\mathcal{P}$ as shown in Eq. 1) exploiting depth estimation network. Subsequently, Detector Head modules $\mathcal{D}$ supervises BEV features with 3D labels Y in a three-dimensional manner.

$$\mathcal{V}(I, K, T) = \mathcal{P}(F_{2d} \otimes D, K, T), \tag{1}$$

3.2 Domain Shifts in Multi-view 3D Object Detection

In this section, we analyze and report *de facto* domain shift problems arising in the Autonomous Driving system. As shown in 3.1, recent works adopt camera parameters K and T as extra inputs in addition to multi-view image I. As reported by [14], assuming that the conditional distribution of outputs for given inputs, is the same across domains, it is explained that shifts in the domain distribution are caused by inconsistent marginal distributions of inputs. To mitigate these issues, recent generalization approaches [14, 53, 47, 13, 54] often focus on covariate shift in geometric feature representation mainly due to optical changes (*i.e.*, Focal length, Field-of-View, and pixel size).

This is the only part of a story. We experience drastic performance drops (up to -54% / -67% performance drop in NDS and mAP, respectively, as shown in Fig 2 (b)) from non-intrinsic factors (*i.e.*, only extrinsic shifts). Especially, we capture a phenomenon wherein the actual depth scale from an ego-vehicle's visual sensor to an object (Fig 2 (a) red boxes) varies depending on the sensor's installation location. Followed by Pythagorean theorem, as the height difference Δh increases, the depth scale difference Δd also increases accordingly. Note that this is not limited to height solely; any shifts in deployment translation (*e.g.*, along the x, y, or z axis) lead to changes in actual depth scale. As a result, perspective view differences significantly hinder the model's three-dimensional geometric understanding by causing depth inconsistency. To address above drawbacks, we introduce a novel penalising strategy that effectively boost depth consistency in various camera geometry shifts.

4

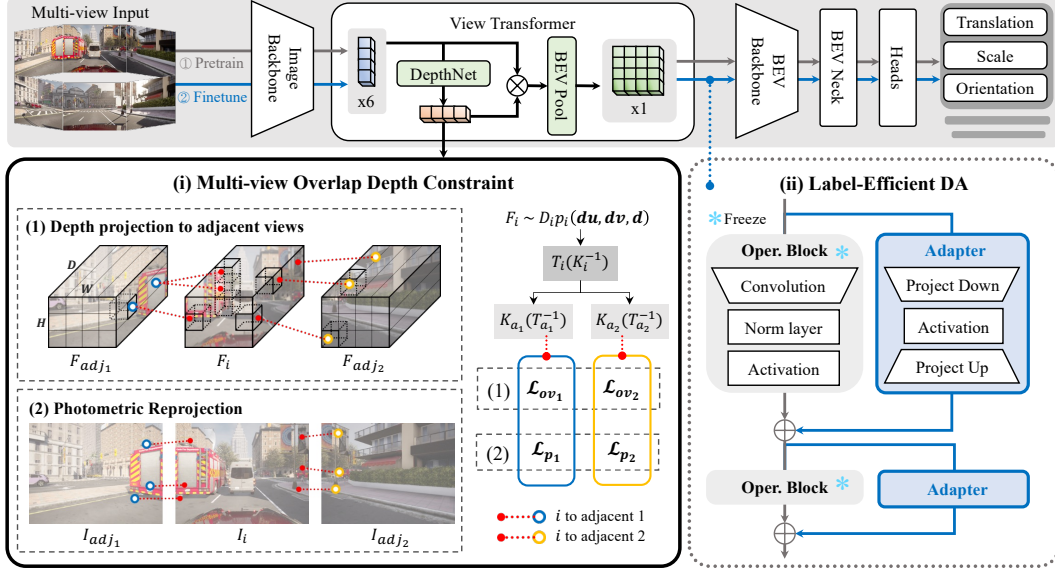



Figure 3: An overview of our proposed methodologies. Our proposed methods comprise two major parts: (i) Multi-view Overlap Depth Constraint and (ii) Label-Efficient Domain Adaptation (LEDA). In addition, our framework employs two phases (*i.e.*, pre-training, and then fine-tuning). Note that we adopt our proposed depth constraint in both phases, and LEDA only in the fine-tuning phase.

3.3 Multi-view Overlap Depth Constraint

Motivation. Recently, previous efforts [55, 14, 54, 56] augment multi-view images to generalize challenging perspective view gaps. However, these strategies suffer from poor generalizability in cross-domain scenarios, primarily due to the underestimated extent of view change between different sensor deployments as reported in section 3.2. To alleviate perspective gaps, we introduce Multi-view Overlap Depth Constraint, effectively encouraging perspective view-invariant learning. Here, we start from three key assumptions: First, perspective shifts between adjacent cameras in multi-view modalities are non-trivial and varied, closely akin to those observed in cross-domains (*e.g.*, nuScenes → Lyft). Second, visual odometry techniques such as Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM) often benefit from improved depth consistency through relationships between adjacent views (*e.g.*, relative pose estimation). Third, in multi-view modalities, overlap regions serve as strong geometric triangular clues, seamlessly bridging between adjacent views. However, under conditions where camera parameters are input, off-the-shelf pose estimation [57–61] leads to ambiguity in learning precise geometry. To mitigate these issues, we introduce a novel depth constraint (Fig. 3 (i)) with overlap regions between adjacent cameras.

Approach. To achieve generalized BEV extraction, we directly constrain depth estimation network from adjacent overlap regions between multi-view cameras. Also, we advocate that multi-frame image inputs substantially complement geometric understanding in dynamic scenes with speedy translation and rotation shifts. To this end, we formulate corresponding depth D^* leveraging spatial and temporal adjacent views. First, we calculate overlap transformation matrices $T_{i \rightarrow j}$ from Eq. 2.

$$D^*_{i \rightarrow j} p^*_{i \rightarrow j} \sim K_j(T_j^{-1}) T_i(K_i^{-1}) D_i p_i, \tag{2}$$

where K and T are the intrinsic and extrinsic camera parameters. $p^*_{i \rightarrow j}$ and p_i denote corresponding pixels between adjacent views and D represent depth prediction. Then, we directly penalise unmatched corresponding depth D^* to smoothly induce perspective-agnostic learning as follow Eq. 3

$$\mathcal{L}_{ov} = \sum_{(i,j)} d(D_j, D^*_{i \rightarrow j}), \tag{3}$$

where d represents Euclidean Distance. Also, we observe that the photometric reprojection error significantly alleviate relative geometric ambiguity. Especially, slow convergence may occur mainly

5

due to incorrect relationships in small overlap region (about 30% of full resolution). To mitigates these concern, we effectively boost elaborate 2D matching, formulating $\mathcal{L}_p$ as follow Eq. 4:

$$\mathcal{L}_p = \sum_{(i,j)} pe(I_j\langle K_j, P_j\rangle, I_j\langle K_j, T_{i\to j}, P^*_{i\to j}\rangle), \tag{4}$$

where P represents point clouds generated by D, and pe is photometric error by SSIM [62]. Also, $\langle\cdot\rangle$ denotes bilinear sampling on RGB images. Concretely, we take two advantages leveraging $\mathcal{L}_p$ in *narrow occluded regions*; First, $\mathcal{L}_p$ effectively mitigates the triangular misalignment. Second, $\mathcal{L}_p$ potentially supports insufficiently scaled $\mathcal{L}_{ov}$. Ultimately, we alleviate perspective view gaps by directly constraining the corresponding depth and the photometric matching between adjacent views.

3.4 Label-Efficient Domain Adaptation

Motivation. There exist practical challenges in developing and deploying multi-view 3D object detectors for safety-critical self-driving vehicles. Each vehicle and each sensor requires its own model that can successfully operate in various conditions (*e.g.*, dynamic weather, location, and time). Furthermore, while collecting large-scale labels in diverse environments is highly recommended, it is extremely expensive, inefficient and time-consuming. Among those, we are particularly motivated to tackle the following: (i) Stable performance, (ii) Efficiency of training, (iii) Preventing catastrophic forgetting, and (iv) Minimizing labeling cost. To satisfy these practical requirements, we carefully design an efficient and effective learning strategy, Label-Efficient Domain Adaptation (LEDA) that seamlessly transferring and preserving their own potentials leveraging a few annotated labels.

Approach. In this paper, we propose Label-Efficient Domain Adaptation, a novel strategy to seamlessly bridge domain gaps leveraging a small amount of target data. To this end, we add extra parameters $\mathcal{A}$ [48] consisting of bottleneck structures (*i.e.*, projection down ϕ_{down} and up ϕ_{up} layers).

$$\mathcal{A}(x) = \phi_{up}(\sigma(\phi_{down}(BN(x)))), \tag{5}$$

where σ and BN indicates activation function and batch normalization. We parallelly build $\mathcal{A}$ alongside pre-trained operation blocks $\mathcal{B}$ (*e.g.*, convolution, and linear block) in Fig. 3 (ii) and Eq. 6;

$$y = \mathcal{B}(x) + \mathcal{A}(x), \tag{6}$$

Firstly, we feed x into ϕ_{down} to compress its shape to $[H/r, W/r]$, where r is the rescale ratio, and then utilize ϕ_{up} to restore it to $[H, W]$. Secondly, we fuse each outputs from $\mathcal{B}$, and Adapter by exploiting skip-connections that directly link between the downsampling and upsampling paths. By doing so, these extensible modules allow to capture high-resolution spatial details while reducing network and computational complexity. Plus, it notes worthy that they are initialized by a near-identity function to preserve previously updated weights. Finally, our frameworks lead to stable recognition in both source and target domains, incrementally adapting without forgetting pre-trained knowledge.

3.5 Optimization Objective

In this section, we optimize our proposed framework UDGA using the total loss function $\mathcal{L}_{total}$ (as shown in Eq. 7) during both phases (*i.e.*, pre-train and fine-tune). $\mathcal{L}_{det}$ denotes the detection task loss.

$$\mathcal{L}_{total} = \lambda_{det}\mathcal{L}_{det} + \lambda_{ov}\mathcal{L}_{ov} + \lambda_p\mathcal{L}_p, \tag{7}$$

where we grid-search λ_{det}, λ_{ov} and λ_p to harmonize $\mathcal{L}_{det}$, $\mathcal{L}_{ov}$ and $\mathcal{L}_p$. Specifically, $\mathcal{L}_{total}$ supervises $\mathcal{B}$ during generalization and $\mathcal{A}$ during adaptation, respectively. As a result, these strategies enable efficient learning of optimal representations in target domains while preserving pre-trained ones.

4 Experimental Results

In this section, we showcase the overall performance of our methodologies on landmark datasets for 3D Object Detection: Waymo [5], Lyft [7], and nuScenes [6]. The three datasets have different specifications; thus, we convert them to a unified detection range and coordinates for accurate comparison. We also adopt only seven parameters to achieve consistent training results under the same conditions: the location of centerness (x, y, z), the size of box (l, w, h), and heading angle θ. Additionally, we summarize 3D Object Detection datasets and implementation details in Appendix A.

6

Table 1: Comparison of Domain Generalization performance with existing SOTA techniques. The **bold** values indicate the best performance. Note that all methods are evaluated on 'car' category.

Task	Method	NDS$^{\hat{*}}$↑	mAP↑	mATE↓	mASE↓	mAOE↓	Closed Gap↑
	Oracle	0.587	0.475	0.577	0.177	0.147	
	Direct Transfer	0.213	0.102	1.143	0.239	0.789	
Lyft → nuScenes	CAM-Convs [13]	0.181	0.098	1.198	0.209	1.064	-8.6%
	Single-DGOD [44]	0.198	0.105	1.166	0.222	0.905	-4.0%
	DG-BEV [14]	0.374	0.268	0.764	0.205	0.591	+43.0%
	PD-BEV [15]	0.344	0.263	**0.746**	0.186	0.790	+35.0%
	Ours	**0.421**	**0.281**	0.759	**0.183**	**0.377**	**+55.6%**
	Oracle	0.684	0.602	0.471	0.152	0.078	
	Direct Transfer	0.296	0.112	0.997	0.176	0.389	
nuScenes → Lyft	CAM-Convs	0.316	0.145	0.999	0.173	0.368	+5.2%
	Single-DGOD	0.332	0.159	0.949	0.174	0.358	+9.3%
	DG-BEV	0.437	0.287	0.771	0.170	0.302	+36.3%
	PD-BEV	0.458	0.304	0.709	0.169	0.289	+41.8%
	Ours	**0.487**	**0.324**	**0.709**	**0.162**	**0.180**	**+49.2%**
	Oracle	0.587	0.475	0.577	0.177	0.147	
	Direct Transfer	0.133	0.032	1.305	0.768	0.532	
Waymo → nuScenes	CAM-Convs	0.215	0.038	1.308	0.316	0.506	+18.1%
	Single-DGOD	0.007	0.014	1.000	1.000	1.000	-27.8%
	DG-BEV	0.472	0.303	0.689	**0.218**	0.171	+74.7%
	Ours	**0.477**	**0.326**	**0.684**	0.263	**0.168**	**+75.8%**
	Oracle	0.649	0.552	0.528	0.148	0.085	
	Direct Transfer	0.178	0.040	1.303	0.265	0.790	
nuScenes → Waymo	CAM-Convs	0.185	0.045	1.301	0.253	0.773	+1.5%
	Single-DGOD	0.164	0.034	1.305	0.262	0.855	-3.0%
	DG-BEV	0.415	0.297	0.822	**0.216**	0.372	+50.3%
	Ours	**0.459**	**0.349**	**0.754**	0.289	**0.250**	**+59.7%**

4.1 Evaluation Metric

In this paper, following DG-BEV [14] evaluation details, we adopt the alternative metric NDS$^{\hat{*}}$ (as shown in Eq. 8) that aggregates mean Average Precision (mAP), mean Average Translation Error (mATE), mean Average Scale Error (mASE), and mean Average Orientation Error (mAOE).

$$\text{NDS}^{\hat{*}} = \frac{1}{6}[3\text{mAP} + \sum_{\text{mTP} \in \mathbb{TP}} (1 - \min(1, \text{mTP}))] \quad (8)$$

We reconstruct the unified category for Unified Domain Generalization and Adaptation as follows: the 'car' for nuScenes and Lyft, and the 'vehicle' for Waymo. Furthermore, we only validate performance in the range of x, y axis from -50m to 50m. Note that we offer an empirical lower bound ***Direct Transfer*** (*i.e.*, directly evaluating the model pre-trained in the source domain only), and an empirical upper bound ***Oracle*** (*i.e.*, evaluating the model fully supervised in the target domain). We report **Full F.T.** (*i.e.*, fine-tuning all parameters from the pre-trained source model) and **Adapter** (*i.e.*, parameter efficient fine-tuning without our proposed depth constraint methods from the pre-trained source model) Furthermore, we formulate **Closed Gap**-representing the hypothetical closed gap by

$$\text{Closed Gap} = \frac{\text{NDS}_{\text{model}} - \text{NDS}_{\text{Direct Transfer}}}{\text{NDS}_{\text{Oracle}} - \text{NDS}_{\text{Direct Transfer}}} \times 100\%. \quad (9)$$

4.2 Experiment Results

Performance Comparison in Domain Generalization. As shown in Tab. 1, we showcase four challenging generalization scenarios, and quantitatively compare our proposed methodology with existing state-of-the-art methods, which include CAM-Conv [13], Single-DGOD [44], DG-BEV [14], and PD-BEV [15]. Here, we observe that these methods still struggle to fully pilot geometric shifts from perspective changes in cross-domain scenarios. Importantly, in Lyft → nuScenes, existing methods suffer from the orientation error mainly due to significantly different ground truth directions (*i.e.*, only recovering 0.198 mAOE). In nuScenes → Waymo (*i.e.*, one of the most challenging

7

Table 2: Comparison of UDGA performance on BEVDepth with various PEFT modules, SSF [50], and Adapter [48]. We construct six different target data splits from 1% to 100%. Additionally, # Params denote the number of parameters for training. Note that — represents *‘Do not support’*.

Task	Method	# Params	NDS⁺↑ / mAP↑					
			1%	5%	10%	25%	50%	100%
Lyft → nuScenes	*Oracle*	51.7M	—	—	—	—	—	0.587 / 0.475
	Full FT	51.7M	0.476 / 0.369	0.515 / 0.434	0.547 / 0.434	0.577 / 0.464	0.590 / 0.483	0.610 / 0.506
	SSF [50]	1M	0.245 / 0.079	0.294 / 0.112	0.360 / 0.256	0.374 / 0.266	0.421 / 0.327	0.439 / 0.275
	Adapter-B	21.3M	0.465 / 0.283	0.481 / 0.365	0.511 / 0.384	0.558 / 0.444	0.569 / 0.460	0.581 / 0.473
	Adapter-S	8.8M	0.326 / 0.134	0.372 / 0.161	0.444 / 0.255	0.465 / 0.283	0.509 / 0.390	0.538 / 0.443
	Ours	8.8M	**0.526 / 0.404**	**0.563 / 0.444**	**0.573 / 0.457**	**0.592 / 0.481**	**0.609 / 0.510**	**0.614 / 0.507**
nuScenes → Lyft	*Oracle*	51.7M	—	—	—	—	—	0.684 / 0.602
	Full FT	51.7M	0.531 / 0.390	0.594 / 0.473	0.623 / 0.513	0.650 / 0.549	0.678 / 0.587	0.700 / 0.615
	SSF	1M	0.316 / 0.115	0.355 / 0.145	0.386 / 0.185	0.420 / 0.230	0.447 / 0.269	0.470 / 0.300
	Adapter-B	21.3M	0.499 / 0.328	0.556 / 0.465	0.584 / 0.475	0.633 / 0.532	0.670 / 0.564	0.684 / 0.596
	Adapter-S	8.8M	0.420 / 0.230	0.463 / 0.325	0.500 / 0.356	0.537 / 0.400	0.561 / 0.426	0.573 / 0.442
	Ours	8.8M	**0.578 / 0.462**	**0.613 / 0.506**	**0.638 / 0.537**	**0.665 / 0.572**	**0.675 / 0.586**	**0.706 / 0.626**

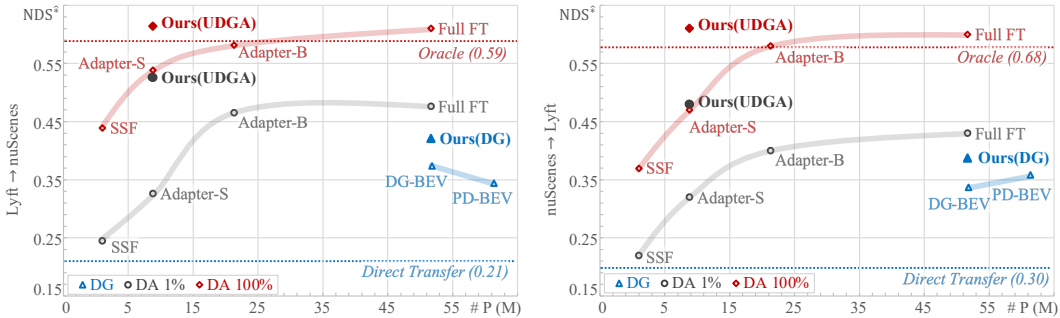



Figure 4: Performance relative to training parameters. The Domain Generalization task is represented in blue, while the Domain Adaptation task is divided into two stages: 1% in gray and 100% in red.

scenarios due to the rear camera drop), previous approaches still show a significant gap compared to *Oracle* (*i.e.*, -49.7% Closed Gap). In this paper, our novel depth constraint notably addresses these issues, outperforming existing SOTAs (especially, up to +4.7% NDS and +12.6% Closed Gap better than DG-BEV in Lyft → nuScenes). Especially, leveraging triangular clues to explicitly supervise occluded depth contributes significantly to improving geometric consistency compared to prior approaches [14, 15, 44, 13]. Overall, we demonstrate that our novel approaches significantly enhance perspective-invariance, featuring strong association in occluded regions between multi-views.

Performance Comparison in UDGA. In Tab. 2, we show that our proposed Unified Domain Generalization and Adaptation performance compared with various PEFT approaches (*i.e.*, SSF [50], and Adapter [48]). SSF directly scale and shift the deep features extracted by pre-trained operation blocks, leveraging additional normalization parameters. Adapter represents sole module performance without our proposed constraint; Adapter-B, and Adapter-S denotes base, and small version, respectively.

Existing PEFT paradigms benefit from fine-tuning only extra parameters, retaining previously updated weights. However, we observe that these paradigms do not successfully adapt to the covariate shifts originated by challenging geometric differences as reported in section 3.2. More specifically, SSF and Adapter-S, which exploit a small number of parameters, begin to capture transferable representations and then marginally adapt at the 10% data split. Also, Adapter-B leveraging 21.3M parameters provide poor adaptation capability (*i.e.*, inferior to Scratch and Full FT in Lyft → nuScenes 100%).

However, our proposed strategy seamlessly adapt to target domains in 1%, and 5%, effectively bridge perspective gaps. Furthermore, our proposed strategy show superior performance gain (outperforming Scratch in Lyft → nuScenes 50%, and Full FT in both Lyft → nuScenes, and nuScenes → Lyft 100%), effectively adapting to novel targets. It is noteworthy that the most effective adaptation is achieved by updating extra parameters (less than 20% of the total), which demonstrates the practicality and efficiency of our novel UDGA strategy as shown in Fig. 4. In addition, unlike Full FT, it proves that our UDGA framework stably adapts to the target without forgetting previously learned knowledge as

8

Table 3: Ablation studies on UDGA (10% Adaptation). $\mathcal{B}$ and $\mathcal{A}$ represents pre-trained blocks and LEDA blocks, respectively. Note that we train $\mathcal{B}$ and $\mathcal{A}$, alternatively (*i.e.*, pre-train and fine-tune).

Pre-train $\mathcal{B}$ (100% source)			Fine-tune $\mathcal{A}$ (10% target)			Lyft → nuScenes		nuScenes → Lyft	
$\mathcal{L}_{det}$	$\mathcal{L}_{ov}$	$\mathcal{L}_p$	$\mathcal{L}_{det}$	$\mathcal{L}_{ov}$	$\mathcal{L}_p$	NDS*↑	mAP↑	NDS*↑	mAP↑
✓						0.213	0.102	0.296	0.112
✓	✓					0.403	0.262	0.485	0.323
✓	✓	✓				0.421	0.281	0.488	0.309
✓			✓			0.444	0.255	0.500	0.356
✓	✓	✓	✓			0.516	0.407	0.590	0.482
✓	✓	✓	✓	✓		0.552	0.441	0.632	0.530
✓	✓	✓	✓	✓	✓	**0.638**	**0.537**	**0.573**	**0.457**

Table 4: Ablation studies on Domain Generalization with our novel depth constraint modules, $\mathcal{L}_{ov}$ and $\mathcal{L}_p$. Lidar and SS each represents LiDAR depth supervision and Self-Supervised overlap depth.

Task	Lidar	SS	Method	NDS*↑	mAP↑	mATE↓	mASE↓	mAOE↓
Lyft → nuScenes	✓		$\mathcal{L}_d$	0.213	0.102	1.143	0.239	0.789
	✓	✓	$\mathcal{L}_d + \mathcal{L}_{ov} + \mathcal{L}_p$	0.396	0.266	0.758	0.172	0.495
		✓	$\mathcal{L}_{ov}$	0.403	0.262	0.757	0.183	0.426
		✓	$\mathcal{L}_{ov} + \mathcal{L}_p$	0.407	0.265	**0.747**	**0.179**	0.428
		✓	$\mathcal{L}_{ov} + \mathcal{L}_p + ext\ aug$	**0.421**	**0.281**	0.759	0.183	**0.377**
nuScenes → Lyft	✓		$\mathcal{L}_d$	0.296	0.112	0.997	0.176	0.389
	✓	✓	$\mathcal{L}_d + \mathcal{L}_{ov} + \mathcal{L}_p$	0.483	0.327	0.718	0.163	0.204
		✓	$\mathcal{L}_{ov}$	0.485	0.323	0.731	**0.161**	0.171
		✓	$\mathcal{L}_{ov} + \mathcal{L}_p$	0.487	**0.324**	0.709	0.162	0.180
		✓	$\mathcal{L}_{ov} + \mathcal{L}_p + ext\ aug$	**0.488**	0.309	**0.705**	0.169	**0.123**

shown in Fig. 1 and Tab. 6. Overall, our proposed method demonstrate the effectiveness of training strategy in various experimental setups, efficiently expanding to targets with about 20% of overall parameters. Note that we report additional experiments and details of UDGA in Appendix C.

4.3 Ablation studies

Exploring the Synergy Between Modules. To better understand the role of each module, we present ablation studies of UDGA in this experiment (Tab. 3). Precisely, we aim to analyze the pros and cons in both training steps (*i.e.*, pre-train $\mathcal{B}$ and fine-tune $\mathcal{A}$), with the objective of effectively elucidating the plausibility of UDGA. First, the strategy trained from scratch leveraging our depth constraint significantly recovers performance drop from the sensor deployment shift (up to +20.8% NDS). However, this strategy finds it difficult to provide a practical solution for Multi-view 3DOD, mainly due to unsatisfying generalizability. Additionally, although LEDA without $\mathcal{L}_{ov}$ and $\mathcal{L}_p$ yields improved performance, it fails to transfer its previously learned potential, resulting in only +2.3% NDS compared to our individual depth constraint. To tackle these issues, we concentrate on bridging two distinct domains by capturing generalized perspective features. Especially, our depth constraint (only trained during pre-training $\mathcal{B}$) significantly encourages understanding of the target in LEDA during fine-tuning $\mathcal{A}$ with a 10% split, addressing the geometric covariate shift (+30.3% NDS). Furthermore, UDGA strategy using $\mathcal{L}_{ov}$ and $\mathcal{L}_p$ in both phases learns the transferable knowledge and shows impressive improvement (+42.5% NDS). Finally, UDGA successfully presents an effective and efficient paradigm for Multi-view 3DOD, highlighting notable recovery in novel target scenarios.

Effect of Overlap Depth Constraint. In Tab. 4, we carefully evaluate our depth constraint components in various cross-domain environments. Here, $\mathcal{L}_d$ denotes depth supervision by LiDAR. Also, we design *ext aug* that globally rotate ground truths with randomly initialized angle α to release the direction shift. More importantly, we observe that perspective view shifts from different sensor deployments lead to severe translation and orientation errors. To tackle these issues, we advocate that $\mathcal{L}_{ov}$, which leverages strong relationships between adjacent views, effectively alleviating perspective gaps compared to $\mathcal{L}_d$ (recovering up to +19% NDS in Lyft → nuScenes). $\mathcal{L}_p$ relieves slight misalignment, encouraging depth-scale consistency. Additionally, our *ext aug* substantially boost stable generalization, suppressing orientation errors (up to +1.4% additional NDS gain). Consequently, our novel objectives ($\mathcal{L}_{ov}$ and $\mathcal{L}_p$) demonstrate their effectiveness, significantly tackling geometric errors.

9

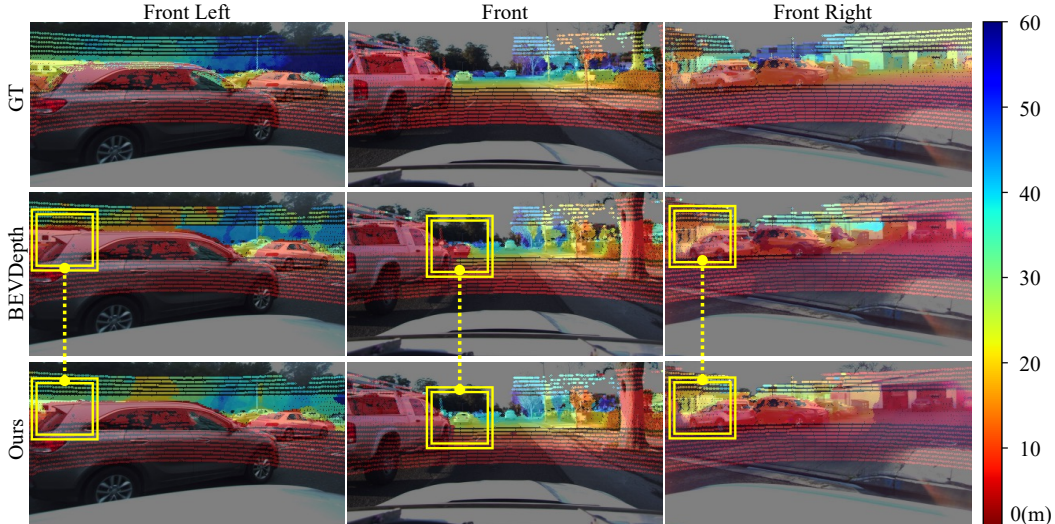



Figure 5: Qualitative depth visualizations of front view lineups in Lyft. The top row illustrates sparse depth ground truths projected from LiDAR point clouds. The middle and bottom rows are the qualitative results of BEVDepth and Ours, respectively. Yellow boxes highlight the improved depth.

4.4 Qualitative Analysis

To qualitatively analyze the effectiveness of Multi-view Overlap Depth Constraint, we present additional visualized results in Fig. 5. For accurate comparison, we conduct binary masking leveraging given sparse depth ground truths. In middle row, BEVDepth fail to perceive hard samples (*e.g.*, far distant and occluded objects) in yellow boxes, mainly due to different extent of deformation relative to perspective as reported in section 3.2. We aim to tackle this problem, explicitly bridging adjacent views in various dynamic scenes. Precisely, in bottom row, we showcase distinguishable results in yellow boxes, capturing semantic details from various view deformation. As as results, we qualitatively demonstrate that our proposed method effectively encourage depth consistency and detection robustness, significantly improving geometric understanding in cross-domain scenarios.

5 Conclusion

Limitations. While our work significantly improves the adaptability of 3D object detection, it cannot guarantee seamless adaptation due to several limitations, including: (1) The performance does not match that of 3D object detection models using LiDAR point clouds. (2) Our Multi-view Overlap Depth Constraint relies on the presence of overlapping regions between images. (3) Achieving fully domain-agnostic approaches without any target labels remains challenging. As a result, it is essential to incorporate a fallback plan when deploying the framework in safety-critical real-world scenarios.

Summary. Multi-View 3DOD models often face challenges in expanding appropriately to unfamiliar datasets due to inevitable domain shifts (*i.e.*, changes in the distribution of data between the training and testing phases). Especially, the limited resource (*e.g.*, excessive computational overhead and taxing expensive and taxing data cost) leads to hinder the successful deployment of Multi-View 3DOD. To mitigate above drawbacks, we carefully design Unified Domain Generalization and Adaptation (UDGA), a practical solution for developing Multi-View 3DOD. We first introduce Multi-view Overlap Depth Constraint that advocates strong triangular clues between adjacent views, significantly bridging perspective gaps. Additionally, we present a Label-Efficient Domain Adaptation approach that enables practical adaptation to novel targets with largely limited labels (*i.e.*, 1% and 5%) without forgetting well-aligned source potential. Our UDGA paradigm efficiently fine-tune additional parameters leveraging significantly fewer annotations by effectively transferring from the source to target domain. In summary, our extensive experiments in various landmark datasets(*e.g.*, nuScenes, Lyft and Waymo) show that our novel paradigm, UDGA, provide a practical solution, outperforming current state-of-the-art models on Multi-view 3D object detection.

10

Acknowledgments and Disclosure of Funding

This work was primarily supported by Samsung Advanced Institute of Technology (SAIT) (85%), partially supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00079, Artificial Intelligence Graduate School Program (Korea University), 5%), and Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2024(International Collaborative Research and Global Talent Development for the Development of Copyright Management and Protection Technologies for Generative AI, RS-2024-00345025, 5%; Research on neural watermark technology for copyright protection of generative AI 3D content, RS-2024-00348469, 5%).

References

[1] 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.

[2] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 972–980, 2024.

[3] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, 35:10421–10434, 2022.

[4] 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. IEEE, 2023.

[5] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. 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.

[6] 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.

[7] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, pages 409–418. PMLR, 2021.

[8] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022.

[9] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022.

[10] Wonseok Roh, Gyusam Chang, Seokha Moon, Giljoo Nam, Chanyoung Kim, Younghyun Kim, Jinkyu Kim, and Sangpil Kim. Ora3d: Overlap region aware multi-view 3d object detection. *arXiv preprint arXiv:2207.00865*, 2022.

[11] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021.

11

[12] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1477–1485, 2023.

[13] Jose M Facil, Benjamin Ummenhofer, Huizhong Zhou, Luis Montesano, Thomas Brox, and Javier Civera. Cam-convs: Camera-aware multi-scale convolutions for single-view depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11826–11835, 2019.

[14] Shuo Wang, Xinhai Zhao, Hai-Ming Xu, Zehui Chen, Dameng Yu, Jiahao Chang, Zhen Yang, and Feng Zhao. Towards domain generalization for multi-view 3d object detection in bird-eye-view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13333–13342, 2023.

[15] Hao Lu, Yunpeng Zhang, Qing Lian, Dalong Du, and Yingcong Chen. Towards generalizable multi-camera 3d object detection via perspective debiasing. *arXiv preprint arXiv:2310.11346*, 2023.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[17] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[18] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.

[19] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[20] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[21] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Probabilistic and Geometric Depth: Detecting objects in perspective. In *Conference on Robot Learning (CoRL) 2021*, 2021.

[22] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021.

[23] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9287–9296, 2019.

[24] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 996–997, 2020.

[25] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.

[26] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019.

[27] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020.

12

[28] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, pages 531–548. Springer, 2022.

[29] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[30] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris M Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. In *The Eleventh International Conference on Learning Representations*, 2022.

[31] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17830–17839, 2023.

[32] Yu Hong, Hang Dai, and Yong Ding. Cross-modality knowledge distillation network for monocular 3d object detection. In *ECCV*, Lecture Notes in Computer Science. Springer, 2022.

[33] Peixiang Huang, Li Liu, Renrui Zhang, Song Zhang, Xinli Xu, Baichao Wang, and Guoyi Liu. Tig-bev: Multi-view bev 3d object detection via target inner-geometry learning. *arXiv preprint arXiv:2212.13979*, 2022.

[34] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinhong Jiang, and Feng Zhao. Bevdistill: Cross-modal bev distillation for multi-view 3d object detection. *arXiv preprint arXiv:2211.09386*, 2022.

[35] Sujin Jang, Dae Ung Jo, Sung Ju Hwang, Dongwook Lee, and Daehyun Ji. Stxd: Structural and temporal cross-modal distillation for multi-view 3d object detection. *Advances in Neural Information Processing Systems*, 36, 2024.

[36] 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.

[37] 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.

[38] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R Qi, and Dragomir Anguelov. Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15446–15456, 2021.

[39] Zeng Yihan, Chunwei Wang, Yunbo Wang, Hang Xu, Chaoqiang Ye, Zhen Yang, and Chao Ma. Learning transferable features for point cloud detection via 3d contrastive co-training. *Advances in Neural Information Processing Systems*, 34:21493–21504, 2021.

[40] Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. In *European Conference on Computer Vision*, pages 179–195. Springer, 2022.

[41] Yanan Zhang, Chao Zhou, and Di Huang. Stal3d: Unsupervised domain adaptation for 3d object detection via collaborating self-training and adversarial learning. *IEEE Transactions on Intelligent Vehicles*, 2024.

[42] Qianjiang Hu, Daizong Liu, and Wei Hu. Density-insensitive unsupervised domain adaption on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17556–17566, 2023.

[43] Vidit Vidit, Martin Engilberge, and Mathieu Salzmann. Clip the gap: A single domain generalization approach for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3219–3229, 2023.

13

[44] Aming Wu and Cheng Deng. Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 847–856, 2022.

[45] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[46] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[47] 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 *European conference on computer vision*, pages 245–262. Springer, 2022.

[48] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[49] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *Advances in Neural Information Processing Systems*, 35:109–123, 2022.

[50] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022.

[51] Yue-Jiang Dong, Yuan-Chen Guo, Ying-Tian Liu, Fang-Lue Zhang, and Song-Hai Zhang. Ppea-depth: Progressive parameter-efficient adaptation for self-supervised monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 1609–1617, 2024.

[52] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[53] Qiqi Gu, Qianyu Zhou, Minghao Xu, Zhengyang Feng, Guangliang Cheng, Xuequan Lu, Jianping Shi, and Lizhuang Ma. Pit: Position-invariant transform for cross-fov domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8761–8770, 2021.

[54] Tzofi Klinghoffer, Jonah Philion, Wenzheng Chen, Or Litany, Zan Gojcic, Jungseock Joo, Ramesh Raskar, Sanja Fidler, and Jose M Alvarez. Towards viewpoint robustness in bird's eye view segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8515–8524, 2023.

[55] Yunhan Zhao, Shu Kong, and Charless Fowlkes. Camera pose matters: Improving depth prediction by mitigating pose distribution bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15759–15768, 2021.

[56] Ke Wang, Bin Fang, Jiye Qian, Su Yang, Xin Zhou, and Jie Zhou. Perspective transformation data augmentation for object detection. *IEEE Access*, 8:4935–4943, 2019.

[57] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019.

[58] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. Nope-nerf: Optimising neural radiance field with no pose prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4160–4169, 2023.

14

[59] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: High resolution self-supervised monocular depth estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 2294–2301, 2021.

[60] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. *arXiv preprint arXiv:2110.09482*, 2021.

[61] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Yongming Rao, Guan Huang, Jiwen Lu, and Jie Zhou. Surrounddepth: Entangling surrounding views for self-supervised multi-camera depth estimation. *arXiv preprint arXiv:2204.03636*, 2022.

[62] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4): 600–612, 2004.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[64] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. 2019.

15

Appendices

Table of Contents

A Datasets   **16**

B Implementation Details   **17**

C Additional Experiments   **17**

D Broader Impacts.   **20**

A Datasets

Table 5: Dataset details. Note that each statistical information is calculated from the whole dataset.

Dataset	Cameras	LiDAR	# scenes	# 3D boxes	Points per Beam	Range	Location	Night	Rain	Highway
nuScenes	6	32-beam	1000	1.4M	1,084	< 100m	USA and Singapore	✓	✓	-
Lyft	6	64-beam	366	1.3M	1,863	< 100m	USA	-	✓	-
Waymo	5	64-beam	1150	12M	2,258	< 100m	USA	✓	✓	-
CARLA	6	128-beam	10	2.0M	2,500	< 100m	Carla Town10	-	-	-

We evaluate overall performance on landmark datasets for 3D Object Detection: Waymo [5], Lyft [7], and nuScenes [6]. The three datasets have different point cloud ranges and specifications. Hence, we convert them to a unified range and coordinates for accurate comparison. We also adopt only seven parameters to achieve consistent training results under the same conditions: center locations (x, y, z), box size (l, w, h), and heading angle θ. Additionally, to estimate practical degradation due to changes in camera positioning, we conducted a proof of concept by generating data similar to the nuScenes using the CARLA simulation. The details are as follows:

Waymo The Waymo dataset [5] consists of high-quality and large-scale data with 230K frames from all 1,150 scenes using multiple LiDAR scanners and cameras. Furthermore, for the generalization purpose, Waymo is recorded at diverse cities, weather conditions, and times. For object detection in 2D or 3D, Waymo provides point cloud-annotated 3D bounding boxes as 3D data pairs and RGB image-annotated 2D bounding boxes as 2D data pairs.

nuScenes The nuScenes dataset [6] uses 6 cameras that cover a full 360-degree range of view and a single LiDAR sensor to obtain 40K frames from 20-second-long 1,000 video sequences, which are fully annotated with 3D bounding boxes for 10 object classes. The nuScenes dataset covers 28k annotated samples for training. Also, validation and test contain 6k scenes each. The nuScenes frames are captured in the same manner as Waymo dataset for the data diversity. But unlike Waymo, nuScenes provides labels only for the point cloud data with 23 classes of 3D bounding boxes.

Lyft The Lyft dataset [7] is motivated by the impact of large-scale datasets on Machine Learning and consists of over 1,000 hours of data. This was collected by a fleet of 20 autonomous vehicles along a fixed route in Palo Alto, California, over a four-month period. It consists of 170,000 scenes (each scene is 25 seconds long) and contains 3D bounding boxes with the precise positions of nearby vehicles, cyclists, and pedestrians over time. In addition, the Lyft dataset includes a high-definition semantic map with 15,242 labelled elements and a high-definition aerial view over the area.

CARLA To quantify the performance drop resulting from camera shifts, we employed an autonomous driving simulation powered by CARLA [52] 0.9.14 and Unreal Engine 4.26. We collected 24K frames for training and 1K frames for each evaluation, driving through Town10 under cloudless weather conditions between sunrise and sunset times. This dataset includes over 100 vehicles and 30 pedestrians in random locations. In Fig. 6, the Source utilizes 6 nuScenes-like cameras and 6 LiDARs, while the *Target* has perturbed sensors. From the Source sensors, the *Height* increases by 0.65m and the *Pitch* increases by 5 degrees. The *All* synthetically moves the x, y, z-coordinates by -0.12m, 0.65m, and -0.2m/+0.2m, respectively, and rotates the yaw by -5/+5 degrees, depending on their directions. Each target sets is collected simultaneously with the Source.

16

Table 6: Comparison of Unified Domain Generalization and Adaptation performance with state-of-the-art techniques. We validate our proposed methods with the same baseline model, named BEVDepth, on Cross-domain. The **bold** values indicate the best performance. Also, — denotes *'Do not support'*.

Task	Method	Branch	Source	Target
			NDS*↑ / mAP↑	NDS*↑ / mAP↑
Lyft → nuScenes	*Direct Transfer*		0.684 / 0.602	0.213 / 0.102
	Oracle		0.296 / 0.112	0.587 / 0.475
	DG-BEV [14]	DG	0.675 / 0.611	0.374 / 0.268
	PD-BEV [15]	DG	0.677 / 0.593	0.344 / 0.263
	PD-BEV	UDA	0.672 / 0.589	0.358 / 0.280
	Ours	DG	**0.702 / 0.630**	0.421 / 0.281
	Ours (1%)	UDGA	**0.702 / 0.630**	0.526 / 0.404
	Ours (5%)	UDGA	**0.702 / 0.630**	**0.563 / 0.444**
nuScenes → Lyft	*Direct Transfer*		0.587 / 0.475	0.296 / 0.112
	Oracle		0.213 / 0.102	0.684 / 0.602
	DG-BEV	DG	0.578 / 0.470	0.437 / 0.287
	PD-BEV	DG	—	0.458 / 0.304
	PD-BEV	UDA	—	0.476 / 0.316
	Ours	DG	**0.623 / 0.513**	0.487 / 0.324
	Ours (1%)	UDGA	**0.623 / 0.513**	0.578 / 0.462
	Ours (5%)	UDGA	**0.623 / 0.513**	**0.613 / 0.506**
Waymo → nuScenes	*Direct Transfer*		0.649 / 0.552	0.133 / 0.032
	Oracle		0.178 / 0.040	0.587 / 0.475
	DG-BEV	DG	**0.660 / 0.568**	0.472 / 0.303
	Ours	DG	0.656 / 0.547	0.477 / 0.326
	Ours (1%)	UDGA	0.656 / 0.547	0.534 / 0.409
	Ours (5%)	UDGA	0.656 / 0.547	**0.571 / 0.448**
nuScenes → Waymo	*Direct Transfer*		0.587 / 0.475	0.178 / 0.040
	Oracle		0.133 / 0.032	0.649 / 0.552
	DG-BEV	DG	0.563 / 0.461	0.415 / 0.297
	Ours	DG	**0.603 / 0.497**	0.459 / 0.349
	Ours (1%)	UDGA	**0.603 / 0.497**	0.509 / 0.378
	Ours (5%)	UDGA	**0.603 / 0.497**	**0.549 / 0.424**

B   Implementation Details

To validate the effectiveness of our proposed methods, we adopt BEVDepth [12] and BEVFormer [9] as our base detectors. Both detectors utilize ResNet50 [63] backbone that initialized from ImageNet-1K. Also, we construct BEV representations within a perception range of [-50.0m, 50.0m] for both the X and Y axes. In BEVDepth, we reshape multi-view input image resolutions as follow: [256, 704] for nuScenes, [384, 704] for Lyft, [320, 704] for Waymo. As following DG-BEV [14], we train 24 epochs with AdamW optimizer by learning rate 2e-4 in pre-training phase. The training takes approximately 18 hours using one A100 GPU. In fine-tuning phase, we conduct an extensive grid search to determine the optimal learning rate proportional to the number of learnable parameters. Note that we extensively augment various image conditions as detailed in [14].

C   Additional Experiments

In this appendix, we present additional experiments to validate the effectiveness of our proposed paradigm. First, Tab. 6 summarizes the overall results of our work from the perspective of domain shift. We also analyze how changes in camera positioning worsen the performance and evaluate whether existing augmentation methods can mitigate the deterioration. Additionally, we conduct ablation studies to enhance the LEDA structure, including comparisons with formal adapters. Finally, we present the comparison results with the transformer-based detector. The qualitative analysis of the multi-view results from our proposed paradigm is included towards the end of this chapter.

17

Performance across domains. In this section, we compare our proposed UDGA with existing solutions (*i.e.*, DG, UDA) in various cross-domain conditions (see Tab. 6). We aim to practically mitigate perspective shifts without hindering well-defined source knowledge. Our DG branch achieves top performance, surpassing *Direct Transfer* in the Source domain. The UDGA, which follows DG, improves Target accuracy without compromising Source performance. Especially, we advocate that UDGA enables efficient adaptation with significantly down-scaled data split (*i.e.*, 1% and 5%). Also, it is noteworthy that UDGA do not forget previously learned potentials, fully transferring to target domains (up to +14.2% NDS gain in Lyft→nuScenes). Overall, UDGA provide a practical solution to address perspective view changes, efficiently adapting with only tiny split.

Table 7: Performance under CALRA-simulated domain changes. The model is trained exclusively on Source. The *diff* shows the Source-Target difference. The **bold** values indicate the worst difference.

Test domain	NDS*↑	mAP↑	mATE↓	mASE↓	mAOE↓
Source	0.666	0.811	0.229	0.122	0.043
Target:*Pitch*	0.449	0.491	0.739	0.159	0.065
diff	-0.217	-0.319	0.510	0.036	0.022
Source	0.688	0.848	0.210	0.111	0.042
Target:*Height*	0.313	0.280	1.362	0.179	0.090
diff	**-0.374**	**-0.568**	**1.152**	0.067	0.048
Source	0.687	0.847	0.211	0.216	0.372
Target:*All*	0.321	0.301	1.357	0.181	0.110
diff	-0.366	-0.546	1.146	**0.069**	**0.071**

Table 8: Performance of multi-view augmentations in domain shift. Gray highlight denotes 'Ours'.

Method	Lyft → nuScenes		nuScenes → Lyft	
	NDS*↑	mAP↑	NDS*↑	mAP↑
Direct Transfer	0.213	0.102	0.296	0.112
GT sampling	0.269	0.211	0.405	0.263
2D augmentation	0.269	0.221	0.423	0.263
3D augmentation	0.289	0.235	0.403	0.243
Extrinsic augmentation	0.298	0.223	0.436	0.255
CBGS [64]	0.265	0.196	0.349	0.215
DG-BEV	0.374	0.268	0.437	0.287
Ours	**0.421**	**0.281**	**0.487**	**0.324**

Practical domain shift analysis. We analyze the impact of changes in camera geometry on 3D object estimation. The experimental model is trained using only the source dataset on ResNet50-based BEVDet and then evaluated on three sets of (source, target) to analyze performance differences. In Tab. 7, the performance of the source is similar in all test sets. On the other hand, the performance of the target decreases significantly in all cases. Since this experiment is conducted in the same environment with the same camera sensors, it demonstrates how much performance degradation is caused by the position of the camera. The set with the largest performance drop in the target is *Height*, where the mATE value increased significantly. The target *All* exhibits the worst mASE and mAOE, while the other measures also deteriorated by a similar amount as *Height*.

Conventional augmentation methods enhance the robustness of the model. We evaluate some of them in Tab. 8. GT sampling and CBGS [64] are techniques designed to balance ground truths. 2D augmentation directly augment multi-view inputs (i.e., image resize, crop and paste, contrast and brightness distortion). 3D and extrinsic methods are global augmentations that address both input and ground truths, and ground truths only, respectively. These methods enhance geometric understanding from input noises. However, in dynamic view changes (i.e., cross-domain), they still suffer from geometric inconsistency and show poor generalization capability. Moreover, various 2D approaches do not guarantee geometric alignments between 2D images and 3D ground truths and relevant studies have not been explored well, as reported in [14] and [55].

Searching adapter structures. We explore various modules and structures to find a suitable adapter architecture. Tab. 9, 10 show which structures and locations affects the model's performance. For adapter locations, performance is optimal when adapters are attached to all modules, gradually

18

Table 9: Performance comparison for each module (UDGA 5%). Gray highlight denotes 'Ours'.

Backbone	View transform	BEV encoder	Detection head	Lyft → nuScenes		nuScenes → Lyft	
				NDS*↑	mAP↑	NDS*↑	mAP↑
				0.421	0.281	0.487	0.324
			✓	0.333	0.237	0.489	0.352
		✓	✓	0.433	0.322	0.551	0.418
	✓	✓	✓	0.525	0.409	0.608	0.498
✓	✓	✓	✓	**0.563**	**0.444**	**0.613**	**0.506**

Table 10: Comparison with various adapter structures (UDGA 10%). Gray highlight denotes 'Ours'.

Method	Project Down	Project Up	# Params	Lyft → nuScenes		nuScenes → Lyft	
				NDS*↑	mAP↑	NDS*↑	mAP↑
Adapter-H	Conv.	Conv.	25.9M	0.547	0.439	0.592	0.484
Adapter-B	Conv.	Linear	21.3M	0.511	0.384	0.584	0.475
Adapter-S	Linear	Conv.	8.8M	0.444	0.255	0.500	0.356
Adapter-T	Linear	Linear	2.9M	0.282	0.262	0.398	0.376
Ours	Conv.	Linear	8.8M	**0.573**	**0.457**	**0.638**	**0.537**

improving with the addition of more. Exceptionally, attaching adapters only at the Detection Head leads to a decline in Lyft→nuScenes. In addition, Tab. 10 represents the performance of various adapter structures. The combination of Convolution and Linear layer respectively for Project Down and Up shows the best performance in both tasks. Note that training with fewer parameters(8.8M) is more effective. However, we suggest that large-scale parameters may require a larger dataset or more training, as we only trained on 10% of the target dataset for less than 20 epochs in this experiment.

Table 11: Comparison of UDGA performance on BEVFormer. We train with two different data splits 50%, and 100%. Additionally, # Params denote the number of parameters for training. The bold values indicate the best performance. — denotes *'Do not support'*.

Task	Method	# Params	NDS*↑	mAP↑
nuScenes → Lyft	*Oracle*	33.5M	0.635	0.534
	Direct Transfer	33.5M	0.338	0.245
	Full FT	33.5M	0.638	0.533
	Ours (50%)	12.2M	0.596	0.477
	Ours (100%)	12.2M	0.638	0.534

Comparison of UDGA on BEVFormer. To demonstrate the validation of UDGA, we further compare performance on BEVFormer-small (33.5M parameters) with Full FT. For accurate comparison, we provide *Oracle*, and *Direct Transfer* in nuScenes → Lyft task.

BEVFormer adopt Query-based view transformation modules $\mathcal{V}$ as follow Eq. 10:

$$\mathcal{V}(I, K, T) = CrossAttn(q : P_{xyz}, k\,v : F_{2d}), \tag{10}$$

where q, k and v represents query, value and key in Transformer, and then P_{xyz} denotes pre-defined anchor BEV positions by K, and T. Here, Query-based module benefits from $CrossAttn$ with sparse query sets, implicitly learning geometric information. Thus, we reconstruct our UDGA paradigm without explicit depth constraints. First, we adopt linear-based bottleneck structures with Layer Normalization in Eq. 11. ϕ_{up} and ϕ_{down} denote the projection up and down layer.

$$y = ffn(x) + \phi_{up}(\sigma(\phi_{down}(LN(x)))), \tag{11}$$

where *ffn* denotes feed-forward networks, and LN represents Layer Normalization. We conduct experiments by plugging these extra modules, which accounts for 36% of the total parameters, into BEVFormer. As a result, we achieve significant adaptation performance with the 50% data split. Notably, we demonstrate effectiveness, achieving parity with Full FT in the 100% data split.

Additional qualitative analysis. In this section, we further visualize our depth quality in various scenarios (*i.e.*, Lyft, and nuScenes). Not only our overlap depth constraint significantly improve depth

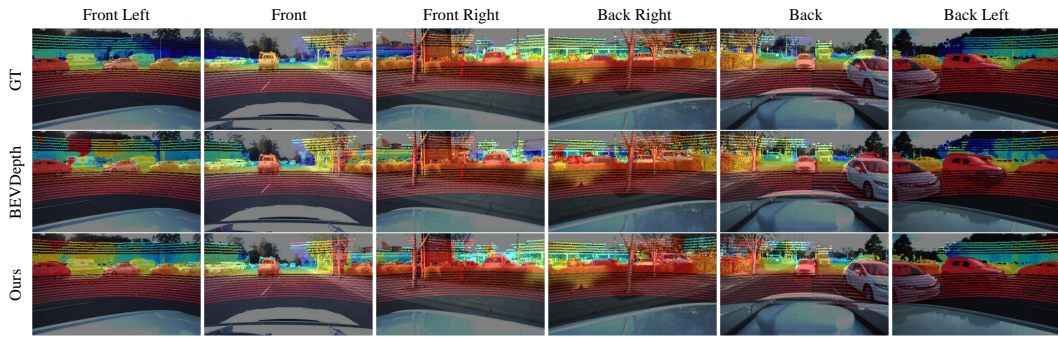
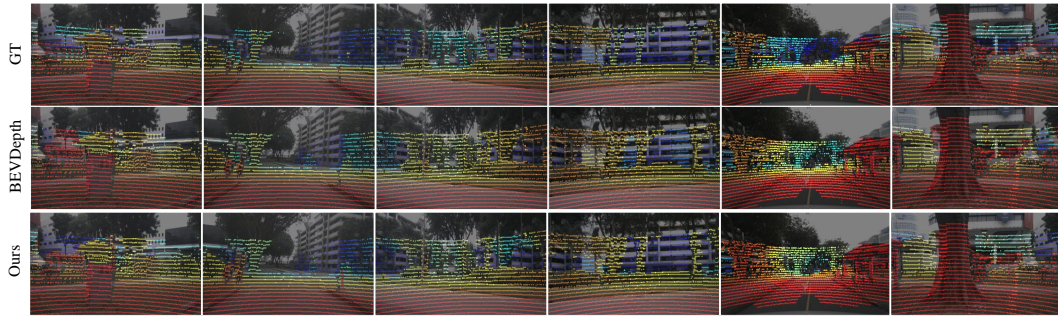
19

Figure 6: The paired sample of each evaluation set in Carla dataset.

consistency in occluded regions, but also show better spatial understanding for hard samples (*e.g.*, far and low distinguishable objects). Additionally, we note that our method effectively complement insufficient contextual recognition caused by sparse depth gt in Fig. 7 (b). Overall, we stably deploy Multi-View 3DOD by leveraging effective association between adjacent views.

D Broader Impacts.

Our framework is a practical AI algorithm that enhances its generalization ability to handle domain changes robustly, enabling us to effectively reduce data costs and computing resources required for adaptation. Practically, our method makes it suitable for deployment in mass-produced vehicles, where the algorithm can inherit the knowledge of well-trained pretrained weights while self-learning to adapt to each fleet environment. The adaptation learning process is also simplified, making it easier to transfer improved pretrained networks. Furthermore, by demonstrating superior performance compared to previous methods that relied on LiDAR for auxiliary depth networks, our approach reduces the dependency on lidar modality. This suggests the feasibility of excluding expensive LiDAR sensors from future autonomous vehicles.

20

(a) Lyft

(b) nuScenes

Figure 7: Multi-view visualization of the depth estimation of BEVDepth and Ours for (a)Lyft and (b)nuScenes samples. In general, our depth consistency was better in the Lyft dataset, while it was difficult to make a quantitative comparison in the case of nuScenes due to the sparseness of the LiDAR point clouds. The depth range is from 1m to 60m. Best viewed in color.

21

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have summarized the contributions of our work at the end of Sec. 1 (Introduction) as well as in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed and presented limitations in Sec. 5 (Conclusion).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

22

Justification:

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have provided detailed information about the implementations in Sec. 4 (Experiment) and in Appendix A, B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

23

Answer: [No]

Justification: We plan to release source codes upon acceptance. At current phase, we have provided details of implementations and necessary references to prior works in Sec. 4 (Experiment) and in Appendix A, B.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have provided detailed information about the implementations in Sec. 4 (Experiment) and in Appendix A, B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification:

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

24

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We have provided details of computing resources in Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We conform and follow the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We have discussed both impacts in Sec. 5 (Conclusion).

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

25

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We discussed the potential risks of the proposed method, which we judge to pose no notable risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have provided credits for the benchmark dataset and cited the baseline models.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

26

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

27