2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Depth Is All You Need for Monocular 3D Detection

Dennis Park*, Jie Li*, Dian Chen, Vitor Guizilini, Adrien Gaidon

***Abstract*— A key contributor to recent progress in 3D detection from single images is monocular depth estimation. Existing methods focus on how to leverage depth explicitly, by generating pseudo-pointclouds or providing attention cues for image features. More recent works leverage depth prediction as a pretraining task and fine-tune the depth representation while training it for 3D detection. However, the adaptation is limited in scale by manual labels. In this work, we propose further aligning the depth representation with the target domain in an unsupervised fashion. Our methods leverage commonly available LiDAR or RGB videos during training time to fine-tune the depth representation, which leads to improved 3D detectors. Especially when using RGB videos, we show that our two-stage training by first generating depth pseudo-labels is critical, because of the inconsistency in loss distribution between the two tasks. With either type of reference data, our multi-task learning approach improves over the state of the art on both KITTI and NuScenes, while matching the test-time complexity of its single-task sub-network. Source code and pretrained models are available on https://github.com/TRI-ML/DD3D.**

I. INTRODUCTION

Recognizing and localizing objects in 3D space is crucial for applications in robotics, autonomous driving, and augmented reality. Hence, in recent years monocular 3D detection has attracted substantial scientific interest [1], [2], [3], [4], because of its broad impact and the ubiquity of cameras. However, as quantitatively shown in [5], the biggest challenge in monocular 3D detection is the inherent ambiguity in depth caused by camera projection. Monocular depth estimation [6], [7], [8], [9] directly addresses this limitation by learning statistical models between pixels and their corresponding depth values, given monocular images.

One of the long-standing questions in 3D detection is how to leverage advances in monocular depth estimation to improve image-based 3D detection. Pioneered by [10], pseudo-LiDAR detectors [11], [12], [13] leverage monocular depth networks to generate intermediate pseudo point-clouds, which are then fed to a point-cloud-based 3D detection network. However, the performance of such methods is bounded by the quality of the pseudo point-clouds, which deteriorates drastically when facing domain gaps. Alternatively, Park et al. showed that by pretraining a network on a large-scale multi-modal dataset where point-cloud data serves as supervision for depth, the simple end-to-end architecture is capable of learning geometry-aware representation and achieving state-of-the-art detection accuracy on the target datasets [1].

However, in [1], the dataset used for pretraining exhibits a significant domain gap from the target data used for 3D detection. The source of this domain gap includes geographical locations (which affect scene density, weather, types of objects, etc.) and sensor configuration (e.g., camera extrinsics and intrinsics). This domain gap limits the scalable transfer of the depth-aware representation to the target domain; the performance in the target domain stops improving as the pretraining data grows large [1]. This work aims to push the boundaries of how much pretrained networks can be adapted for robust 3D detection using various types of unlabeled data available in the target domain.

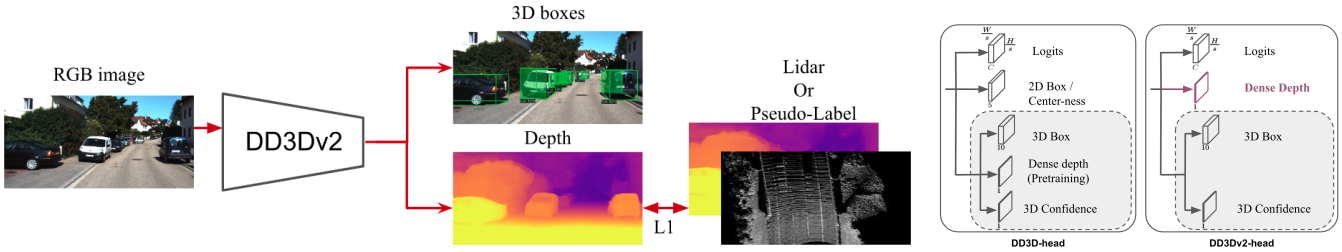
We first consider scenarios where in-domain point-cloud data is available at training time, sharing the assumptions with [8], [9]. In this case, we show that a simple multi-task framework supervised directly with projected depth maps along with 3D bounding boxes yields impressive improvements, compared with pseudo-LiDAR approaches [11], [12] or pretraining based methods [1]. Unlike pseudo-LiDAR methods, our methods entail no additional overhead at test time.

While it spawns insightful research ideas, the assumption that in-domain point-cloud data is available during training can be impractical. For example, most outdoor datasets for 3D detection assume either multi-modal settings [14], [15], [16] or a camera-only setting [17], [18] during both training and testing. Therefore, we propose an alternative variant to our method which adapts depth representations requiring only RGB videos.

Inspired by advances in self-supervised monocular depth estimation [6], [7], [19], we extend our method to using temporally adjacent video frames when LiDAR modality is not available. In this case, we observe that naively applying the same multi-task strategy with the two heterogeneous types of loss (2D photometric loss [7] and 3D box L1 distance) results in sub-par performance. To address this heterogeneity, we propose a two-stage method: first, we train a self-supervised depth estimator using raw sequence data to generate dense depth predictions or *pseudo-depth* labels. Afterward, we train a multi-task network supervised on these pseudo labels, using a distance-based loss akin to the one used to train the 3D detection. We show that this two-stage framework is crucial to effectively harness the learned self-supervised depth as a means for accurate 3D detection. In summary, our contributions are as follows:

- We propose a simple and effective multi-task network, DD3Dv2, to refine depth representation for more accu-

*Equal Contribution
DP: dennis.park@getcruise.com
JL: jie.li@tri.global
DC, VG, AG: firstname.lastname@tri.global
This work was done while DP and JL were at Toyota Research Institute.



(a) DD3Dv2: Depth Supervision Improves 3D Detection (b) Multi-task Head

Fig. 1: DD3Dv2. This paper proposes a simple and effective algorithm to improve monocular 3D detection through depth supervision. (a): The overall flowchart of our proposed system can be adapted to both LiDAR supervision or Camera videos through pseudo labels generated from self-supervision algorithms. (b) Our multi-task decoder head improves on top of the original DD3D by removing redundant information streams.

rate 3D detection. Our method uses depth supervision from unlabelled data in the target domain at training time *only*.

- We propose methods for learning depth representation under two practical scenarios of data availability: LiDAR or RGB video. For the latter scenario, we propose a two-stage training strategy to resolve the heterogeneity among the multi-task losses imposed by image-based self-supervised depth estimation. We show that this is crucial for performance gain with empirical experiments.
- We evaluate our proposed algorithms in two challenging 3D detection benchmarks and achieve state-of-the-art performance.

II. RELATED WORK

A. Monocular 3D detection

Early methods in monocular 3D detection focused on using geometry cues or pretrained 3D representations to predict 3D attributes from 2D detections and enforce 2D-3D consistency [20], [21], [22], [2], [23]. They often require additional data to obtain geometry information, such as CAD models or instance segmentation masks at training time, and the resulting performance was quite limited.

Inspired by the success of point-cloud-based detectors, a series of *Pseudo-LiDAR* methods were proposed [10], [24], [13], [25], [26], which first convert images into a point-cloud using depth estimators, and then apply ideas of point-cloud based detector. A clear advantage of such methods is that, in theory, a continuous improvement in depth estimation leads to more accurate detectors. However, the additional depth estimator incurs a large overhead in inference.

An alternative category is end-to-end 3D detection, in which 3D bounding boxes are directly regressed from CNN features [27], [4], [3], [1]. These methods directly regress 3D cuboid parameterizations from standard 2D detectors [28], [29]. While these methods tend to be simpler and more efficient, these methods do not address the biggest challenge of image-based detectors, the ambiguity in depth. DD3D [1] partially addresses this issue by pretraining the network on a large-scale image-LiDAR dataset.

Our work adopts the idea of end-to-end detectors, pushing the boundary of how far a good depth representation can help accurate 3D detection. Our key idea is to leverage raw data in the target domain, such as point-clouds or video frames, to improve the learning of geometry-aware representation for accurate 3D detection.

Other recent works try to leverage dense depth or its uncertainty as explicit information for 3D lifting [30], feature attention [31] or detection score [32]. MonoDTR [33] shares a similar spirit with us in leveraging in-domain depth through multitask network. However, MonoDTR focuses on the use of the predicted depth to help query learning in a Transfomer-style detector [34]. Compared to these methods, our method focuses on implicit learning of the depth information through proper supervision signal and training strategy. No additional module or test-time overhead is involved in the baseline 3D detector.

B. Monocular Depth Estimation

Monocular depth estimation is the task of generating per-pixel depth from a single image. Such methods usually fall within two different categories, depending on how training is conducted. *Supervised* methods rely on ground-truth depth maps, generated by projecting information from a range sensor (e.g., LiDAR) onto the image plane. The training objective aims to directly minimize the 3D prediction error. In contrast, *self-supervised methods* minimize the 2D reprojection error between temporally adjacent frames, obtained by warping information from one onto another given predicted depth and camera transformation. A photometric object is used to minimize the error between original and warped frames, which enables the learning of depth estimation as a proxy task.

Another aspect that differentiates these two approaches is the nature of learned features. Supervised methods optimize 3D quantities (i.e., the metric location of ground-truth and predicted point-clouds), whereas self-supervised methods operate in the 2D space, aiming to minimize reprojected RGB information. Because of that, most *semi-supervised* methods, that combine small-scale supervision with large-scale self-supervision, need ways to harmonize these two losses, to avoid task interference even though the task is the same. In [35], the supervised loss is projected onto the image plane in the form of a reprojected distance, leading to improved results relative to the naive combination of both losses. In

this work, we take the opposite approach and propose to revert the 2D self-supervised loss back onto the 3D space, through pseudo-label.

III. MULTI-TASK LEARNING FOR 3D DETECTION

In this section, we introduce our multitask framework to adapt geometry-ware features in the target domain during training. While our proposed approach can be generalized to any end-to-end 3D detector ([27], [3]), we build our model on top of DD3D [1] as a baseline. We briefly recapitulate DD3D and highlight our modifications to facilitate in-domain depth feature learning in our model, *DD3Dv2*, as also depicted in Figure 1b.

DD3D Baseline DD3D [1] is a fully convolutional network designed for 3D detection and pretraining supervised by point-cloud data. The backbone network transforms the input image to a set of CNN features with various resolutions. The CNN features are then processed by three different *heads*, each comprising 4 of 3×3 convolutional layers and compute logits and parameterizations of 2D / 3D boxes. We refer the readers to [1] for more detail on the architecture and decoding schemes.

Depth head. The design of a shared head for depth and 3D box prediction in DD3D is motivated by enhancing knowledge transfer between the (depth) pretraining and detection. However, in the scenario of multi-task, we found that excessive sharing of parameters causes unstable training. Therefore, we keep the parameters for depth prediction as an independent head with the same architecture of other heads, which consists of 4 of 3×3 convolution layers.

Removal of 2D box head. Adding an additional head incurs significant overhead in memory and hinders large-scale training with high-resolution images. Since we are only interested in 3D detection, we remove the 2D box head and center-ness. The 2D boxes used in non-maxima suppression are replaced by axis-aligned boxes that tightly contain the projected key points of 3D boxes. This results in a three-head network, with similar memory footprints of DD3D.

Improved instance-feature assignment. When training fully convolutional detectors, one must decide how to associate the ground-truth instances to the predicted candidates. DD3D adopts a CenterNet-style [36] strategy that matches the centers of ground-truth 2D boxes with the feature locations. However, applying this method to multi-resolution features (e.g., FPN [37]) causes a boundary effect between scale ranges. Instead of applying hard boundaries in scale space, we adopt a strategy of using *anchor boxes* (i.e., 2D boxes with various sizes and aspect ratios centered at a feature location) associated with features to determine the assignments. Given a feature location l and a ground-truth bounding box $\mathcal{B}_g = (x_1, y_1, x_2, y_2)$, the matching function $\mathcal{M}$ is defined as:

$$\mathcal{M}(l, \mathcal{B}_g) = I[\max_{\mathcal{B}_a \in \mathcal{A}(l)} v(\mathcal{B}_a, \mathcal{B}_g) > \tau] \quad (1)$$

where $\mathcal{A}(l)$ is a set of anchor boxes associated with the location l, $v(\cdot, \cdot)$ is an overlapping criteria (e.g., IoU), and τ is a threshold.

This effectively produces a *soft* boundary between the scale ranges and allows for many-to-one assignments. We observed that this leads to more stable training. On nuScenes validation split, this modification leads to a significant improvement in detection accuracy, from 38.9% to 41.2% mAP.

IV. LEARNING DEPTH REPRESENTATION

In this section, we describe how *DD3Dv2* can be trained under different in-domain data availability.

A. Using point cloud

When point cloud data is available, we directly use it as supervision for the depth head in our multi-task training. Following [1], we first project the point cloud onto the image plane and calculate smoothed L1 distance on the pixels with valid ground truth. Camera intrinsics are used to re-scale the depth prediction to account for variable input resolutions caused by data augmentation [1].

B. Using camera video

Given video frames instead of point cloud data, we adopt a two-stage pseudo-label framework. Concretely, as depicted in Figure 2(b), we first learn a depth network on the target data via self-supervised depth estimation ([6], [7]) in stage I, and then train our multi-task network using pseudo depth labels generated from the learned depth network. Stage II is similar to Sec. IV-A, but the target (pseudo) depth labels are dense compared to LiDAR point clouds.

Single-stage vs. Two-Stage Given video frames, the most direct and computationally efficient way to use it with DD3Dv2 is to adopt the same multi-task training, substituting the direct depth supervision with self-supervised photometric loss [38] (Fig. 2(a)). We refer to it as the single-stage strategy for the rest of the paper.

The photometric loss substitutes the direct depth estimation error with reprojection error in RGB space between two images: the target image on which the pixel-wise depth is estimated I_t, and the synthesized version of it formed by warping the neighboring frames $\hat{I}_t$. The difference in appearance is measured by SSIM [39] and L1 distance of (normalized) pixel values:

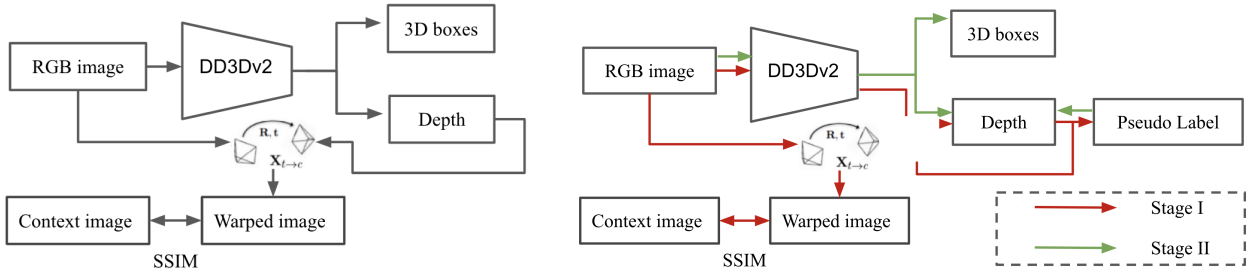
$$\mathcal{L}_p(I_t, \hat{I}_t) = \alpha \frac{1 - \text{SSIM}(I_t, \hat{I}_t)}{2} + (1 - \alpha)||I_t - \hat{I}_t|| \quad (2)$$

While photometric loss has been widely adopted in most of the self-supervised monocular depth estimation works [38], [6], we found that it does not work compatibly with direct 3D losses used in 3D detection, as demonstrated in Table IV (E3, E4 vs E1).

For 3D detection optimization, we apply disentangling 3D boxes loss computation [40] on 3D box loss to optimize 3D box components independently (orientation, projected center, depth, and size).

$$\mathcal{L}_{\text{3D}}(\mathbf{B}^*, \hat{\mathbf{B}}) = ||\mathbf{B}^* - \hat{\mathbf{B}}||_1, \quad (3)$$

where ground truth for other components is provided when the targeted component is being evaluated. In the case of depth, the 3D box loss equals a simple L1 loss.



(a) Single stage strategy

(b) Two stages strategy (Proposed)

Fig. 2: To use self-supervision techniques to guide depth supervision using context images from the video, we discuss two training strategies here. (a) The most straight forward and convenient strategy would be directly combine the self-supervised training paradigm as part of the multi-task network. (b) The second strategy would be first train a depth network that can be used to generate pseudo ground truth depth. Then apply multi-task training in the second stage using pseudo label the same way we use LiDAR. We found that the second strategy provide more significant improvement to the original 3D detection compared to the first one.

In the single-stage strategy, this heterogeneity of the two losses causes a large difference in the distribution of depth prediction and its error. In Figure 3, we visualize these losses to better illustrate this heterogeneity.

Compare to L1 loss, the photometric loss is correlated with the structure and the appearance of the scene. It exhibits different patterns depending on the distance of the object or structure in a scene. For example, objects further away or towards the vanishing point will be less sensitive to the depth error, due to a decrease in pixel resolution. A similar observation is also discussed in [35].

To address this inconsistency, we propose to use the self-supervised depth network in a similar fashion to show how we use point-cloud data. Namely, we apply the self-supervised network to training data to obtain *pseudo* depth labels, which are used in the same way as LiDAR point cloud to train the multi-task network with L1 loss. In this way, the depth loss shares the L1 nature (i.e., distance in 3D scenes) as detection loss. This yields improvement in 3D detection (Sec. VI).

V. BENCHMARK RESULTS

A. Datasets

nuScenes. The nuScenes dataset [14] contains 1000 videos divided into training, validation and test splits with 700, 150, and 150 *scenes*, respectively. Each sample is composed of 6 cameras covering the full 360-degree field of view around the vehicle, with corresponding annotations. The evaluation metric, *nuScenes detection score* (*NDS*), is computed as a linear combination of mean average precision *mAP* over four thresholds on center distance and five *true-positive* metrics. We report NDS and mAP, along with the three true-positive metrics that concern 3D detection, i.e., *ATE*, *ASE*, and *AOE*.

KITTI-3D. The KITTI-3D benchmark [16] contains a training set of 7481 images and a test set of 7518 images. For the 3D detection task, three object classes are evaluated on two average precision (AP) metrics: *3D AP* and *BEV AP*, which use intersection-over-union criteria on (projected) 3D boxes. The metrics are computed on three difficulty levels: Easy, Moderate, and Hard.

B. Implementation Details

In all experiments, we initiate our model using pretrained weights (V2-99) from [1]. We use the SGD optimizer with a learning rate of 2×10^{-3}, momentum of 0.9 and weight decay at 1×10^{-4}, and batch size of 64. For nuScenes, we train our model for $120K$ iterations with multi-step scheduler that decreases the learning rate by 10 at steps $100K$ and $119K$. For KITTI, we train for $35K$ iterations and similarly decrease the learning rate at $32K$ and $34K$ steps. Ground truth poses are used in self-supervised depth training.

C. Discussion

In Table I, we compare our model with published monocular approaches. (We exclude entries that use temporal cues at test time.) DD3Dv2, when trained using point-cloud supervision, yields higher accuracy than all other methods, including recent Transformer-based methods. When trained using video frames, it performs competitively with other methods, and shows impressive improvement over DD3D.

In Tables II and III, we show the results on KITTI-3D benchmark. We report our results with point-cloud supervision, since KITTI allows for only a single submission. (Comparison of self-supervised depth is provided in supplemental material.) DD3Dv2 achieves the state-of-the-art in most metrics across all three categories when compared with most published and concurrent works, including the ones that uses similar point-cloud supervision and Pseudo-LiDAR approaches. Our new representation significantly improves over end-to-end approaches like [1], especially on smaller objects.

VI. ABLATION ANALYSIS

Experimental setup. For ablative study, we use nuScenes dataset (*train* and *validation*). To cover a wide range of variations, we adopt a lightweight version of the full training protocol with half training steps and batch size. The reduced

(a) RGB image (b) L1 Loss on depth (c) Photometric loss

Fig. 3: Heterogeneity between photometric loss and L1 loss. We provide an illustration of the two loss distribution when depth prediction error is **1m** everywhere. While $L1$ loss is a direct measurement of depth estimation, photometric loss is also correlate with the structure and appearance in the scene. (This figure use perfect ground truth from vKITTI [41] dataset.)

Methods	Depth Sup.	Backbone	AP[%]↑	ATE[m]↓	ASE[1-IoU]↓	AOE[rad]↓	NDS↑
MonoDIS [40]	-	R34	30.4	0.74	0.26	0.55	0.38
FCOS3D [3]	-	R101	35.8	0.69	0.25	0.45	0.43
PGD[42]	-	R101	37.0	0.66	0.25	0.49	0.43
DD3D [1]	-	V2-99	41.8	0.57	0.25	0.37	0.48
DETR3D [**?**]	-	V2-99	41.2	0.64	0.26	0.39	0.48
BEVDet* [43]	-	V2-99	42.4	**0.52**	**0.24**	0.37	0.49
BEVFormer-S* [44]	-	V2-99	43.5	0.59	0.25	0.40	0.50
PETR* [45]	-	V2-99	44.1	0.59	0.25	0.38	0.50
DD3Dv2-selfsup	Video	V2-99	43.1	0.57	0.25	0.38	0.48
DD3Dv2	LiDAR	V2-99	**46.1**	**0.52**	**0.24**	**0.36**	**0.51**

TABLE I: **nuScenes detection *test* set evaluation.** We present summary metrics of the benchmark. * denotes results reported on the benchmark that do not have associated publications at the time of writing. The **bold** and underline denote the best of all and the best excluding concurrent work, respectively. Note that PointPillars [46] is a LiDAR-based detector.

Methods	Depth Sup.	Car					
		BEV AP			3D AP		
		Easy	Med	Hard	Easy	Med	Hard
SMOKE [27]	-	20.83	14.49	12.75	14.03	9.76	7.84
MonoPair [47]	-	19.28	14.83	12.89	13.04	9.99	8.65
AM3D [26]	LiDAR	25.03	17.32	14.91	16.50	10.74	9.52
PatchNet† [12]	LiDAR	22.97	16.86	14.97	15.68	11.12	10.17
RefinedMPL [48]		28.08	17.60	13.95	18.09	11.14	8.96
D4LCN [49]	LiDAR	22.51	16.02	12.55	16.65	11.72	9.51
Kinematic3D [50]	Video	26.99	17.52	13.10	19.07	12.72	9.17
Demystifying [5]	LiDAR	-	-	-	23.66	13.25	11.23
CaDDN [30]	LiDAR	27.94	18.91	17.19	19.17	13.41	11.46
MonoEF [51]	Video	29.03	19.70	17.26	21.29	13.87	11.71
MonoFlex [52]	-	28.23	19.75	16.89	19.94	13.89	12.07
GUPNet [53]	-	-	-	-	20.11	14.20	11.77
PGD [42]	-	30.56	23.67	20.84	24.35	18.34	16.90
DD3D [1]	-	30.98	22.56	20.03	23.22	16.34	14.20
MonoDTR⋆ [33]	LiDAR	28.59	20.38	17.14	21.99	15.39	12.73
PS-fld⋆† [54]	LiDAR	32.64	23.76	20.64	23.74	**17.74**	15.14
MonoDDE⋆ [55]	-	33.58	23.46	20.37	23.74	17.14	15.10
Ours	LiDAR	**35.70**	**24.67**	**21.73**	**26.36**	17.61	**15.32**

TABLE II: **KITTI-3D *test* set evaluation on *Car*.** We report $\text{AP}|_{R_{40}}$ metrics. ⋆ indicates concurrent works. † indicates the usage of the **KITTI-depth** dataset, with a known information leakage between training and validation splits [5]. **Bold** and underline denote the best of all and the best excluding concurrent work.

training schedule causes degradation in detection accuracy of baseline detection-only DD3Dv2 model from 41.1% to 35.8% mAP. To understand the interplay between detection and depth accuracy, we also report depth metrics computed *only* on foreground regions.

a) Is supervised depth using point-cloud data effectively?: With direct supervision for the depth estimation task, E2 achieves clear improvement compared to E1. This supports our argument that even without a significant change of architecture or explicit use of depth prediction, the representation for 3D detection can be significantly improved by adapting to a good depth representation.

b) Are pseudo-labels necessary for self-supervised depth?: When the supervision of depth is replaced by the self-supervision from video frames, we observe a clear loss in accuracy (E3/E4 compared to E1), and it only yields a mediocre improvement over the DD3Dv2 single-task baseline. This gap is noticeably closed by training on the pseudo-

Methods	Pedestrian						Cyclist					
	BEV AP			3D AP			BEV AP			3D AP		
	Easy	Med	Hard	Easy	Med	Hard	Easy	Med	Hard	Easy	Med	Hard
M3D-RPN [**?**]	5.65	4.05	3.29	4.92	3.48	2.94	1.25	0.81	0.78	0.94	0.65	0.47
MonoPSR [23]	7.24	4.56	4.11	6.12	4.00	3.30	9.87	5.78	4.57	8.37	4.74	3.68
CaDDN [30]	14.72	9.41	8.17	12.87	8.14	6.76	9.67	5.38	4.75	7.00	3.41	3.30
DD3D	15.90	10.85	8.05	13.91	9.30	8.05	3.20	1.99	1.79	2.39	1.52	1.31
MonoDTR⋆ [33]	16.66	10.59	9.00	15.33	10.18	8.61	5.84	4.11	3.48	5.05	3.27	3.19
MonoDDE* [55]	12.38	8.41	7.16	11.13	7.32	6.67	6.68	4.36	3.76	5.94	3.78	3.33
PS-fld⋆† [54]	**19.03**	**12.23**	**10.53**	**16.95**	**10.82**	**9.26**	**12.80**	**7.29**	**6.05**	**11.22**	**6.18**	**5.21**
Ours	17.74	12.16	10.49	16.25	10.82	9.24	10.67	7.02	5.78	8.79	5.68	4.75

TABLE III: **KITTI-3D *test* set evaluation on *Pedestrian* and *Cyclist***. ⋆ indicates concurrent works. † indicates the usage of the **KITTI-depth** dataset. **Bold** and underline denote the best of all and the best excluding concurrent work.

ID	Approach	Extra Data	Pseudo Labels	Depth Loss	Detection Accuracy NDS ↑ (mAP [%]↑)	Depth Accuracy Abs. Rel ↓
E1	Detection Only	-	-	L1	41.2 (35.8)	-
E2	DD3Dv2	LiDAR	-	L1	**45.6** (39.1)	0.20
E3	Self-supervised	Video	-	SSIM	42.8 (36.4)	0.51
E4	+ ignore close	Video	-	SSIM	42.9 (37.5)	0.54
E5	DD3Dv2-selfsup	Video	√	L1	43.2 (37.7)	0.51 → 0.52
E6	+ ignore close	Video	√	L1	43.7 (36.9)	0.54 → 0.54

TABLE IV: We provide an ablation analysis on crucial design choices of both architecture and training strategies. We show how LiDAR supervision improves on top of single-task training (E2 vs. E1). In E3 and E4, we employ a single-stage training strategy using video frames as depicted in Figure 2(a). In E5 and E6, we employ a two-stage training strategy by generating pseudo-labels first as depicted in Figure 2(b). "ignore close" indicate a small trick to ignore closest depth estimation in self-supervised training. All methods start from a single initial model pretrained by large-scale depth supervision available from [1].

Backbone	Multi-task	Pretrained Dataset	Pretrained Task	NDS ↑	mAP [%]↑
V2-99	-	DDAD15M	Depth Est.	41.2	35.8
V2-99	√	DDAD15M	Depth Est.	45.6 (+**4.4**)	39.1 (+**3.3**)
V2-99	-	COCO	2D Det.	40.8	34.0
V2-99	√	COCO	2D Det.	43.1 (+2.3)	36.2(+2.2)

TABLE V: We analyzed the relationship between the pretraining backbone and proposed the in-domain multi-task representation learning using depth supervision. We compare the same backbone training on COCO [56] on 2D detection. (Released by [57].) The multi-task training paradigm is consistently improving over the detection-only case. It is also noticeable that geometry-aware backbones (pretrained on depth estimation) achieve more significant improvement than object-aware backbones (COCO).

labels (E5 vs. E3, E6 vs. E4). The pseudo-labels significantly reduce the gap from the naive multi-task training. We argue that removing the heterogeneity in the combined loss results in a better adaptation.

c) When does depth-supervised Multi-task work?: To better understand and evaluate the generalizability of the proposed training paradigm, we analyze the effectiveness of LiDAR supervision against different pretraining conditions in Table V. We compare the geometry-aware backbone (DD3D15M [1]) and objectness-aware backbone (COCO [56] released by [57]). From both of the pretraining weights, multi-task learning with dense depth supervision can improve 3D detection by a clear margin. The geometry-aware model sees a higher improvement (4.4 over 2.3 NDS), which further verifies our intuition that the multi-task training improves the adaptation of the geometry information in the pretrained weights into the target domain.

VII. CONCLUSION

In this paper, we explore the use of in-domain depth estimation for end-to-end monocular 3D detection through implicit depth representation learning. We propose to leverage depth estimation as a proxy task through a multitasking network that encourages representation alignment when either LiDAR data or RGB videos are available in the target domain during training. Our approach focuses on strengthening representation learning, which is generalizable and complementary to other advances in end-to-end 3D detection algorithms.

REFERENCES

[1] D. Park, R. Ambrus, V. Guizilini, J. Li, and A. Gaidon, "Is pseudo-lidar needed for monocular 3D object detection?" in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[2] F. Manhardt, W. Kehl, and A. Gaidon, "ROI-10d: Monocular lifting of 2D detection to 6d pose and metric shape," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2069–2078.

[3] T. Wang, X. Zhu, J. Pang, and D. Lin, "FCOS3D: Fully convolutional one-stage monocular 3D object detection," *arXiv preprint arXiv:2104.10956*, 2021.

[4] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1991–1999.

[5] A. Simonelli, S. R. Bulò, L. Porzi, P. Kontschieder, and E. Ricci, "Demystifying pseudo-lidar for monocular 3D object detection," *arXiv preprint arXiv:2012.05796*, 2020.

[6] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3D packing for self-supervised monocular depth estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.

[8] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2002–2011.

[9] C. Godard, O. Mac Aodha, and G. Brostow, "Digging into self-supervised monocular depth estimation," *arXiv preprint arXiv:1806.01260v3*, 2018.

[10] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[11] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-LiDAR++: Accurate depth for 3D object detection in autonomous driving," *arXiv preprint arXiv:1906.06310*, 2019.

[12] X. Ma, S. Liu, Z. Xia, H. Zhang, X. Zeng, and W. Ouyang, "Rethinking pseudo-LiDAR representation," in *European Conference on Computer Vision*. Springer, 2020, pp. 311–327.

[13] R. Qian, D. Garg, Y. Wang, Y. You, S. Belongie, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao, "End-to-end pseudo-LiDAR for image-based 3D object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5881–5890.

[14] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "NuScenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[15] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[16] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[17] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[18] A. Ahmadyan, L. Zhang, A. Ablavatski, J. Wei, and M. Grundmann, "Objectron: A large scale dataset of object-centric videos in the wild with pose annotations," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[19] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *arXiv preprint arXiv:1805.09817*, 2018.

[20] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau, "Deep MANTA: A coarse-to-fine many-task network for joint 2D and 3D vehicle analysis from monocular image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[21] J. A. Ansari, S. Sharma, A. Majumdar, J. K. Murthy, and K. M. Krishna, "The earth ain't flat: Monocular reconstruction of vehicles on steep and graded roads from a moving camera," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018*. IEEE, 2018, pp. 8404–8410. [Online]. Available: https://doi.org/10.1109/IROS.2018.8593698

[22] I. Barabanau, A. Artemov, E. Burnaev, and V. Murashkin, "Monocular 3D object detection via geometric reasoning on keypoints," *arXiv preprint arXiv:1905.05618*, 2019.

[23] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3D object detection leveraging accurate proposals and shape reconstruction," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 867–11 876.

[24] X. Weng and K. Kitani, "Monocular 3D object detection with pseudo-LiDAR point cloud," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[25] Y. Wang, X. Chen, Y. You, L. E. Li, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. 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*, 2020, pp. 11 713–11 723.

[26] X. Ma, Z. Wang, H. Li, P. Zhang, W. Ouyang, and X. Fan, "Accurate monocular 3D object detection via color-embedded 3D reconstruction for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6851–6860.

[27] Z. Liu, Z. Wu, and R. 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*, 2020, pp. 996–997.

[28] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9627–9636.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[30] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distributionnetwork for monocular 3D object detection," *CVPR*, 2021.

[31] R. Zhang, H. Qiu, T. Wang, X. Xu, Z. Guo, Y. Qiao, P. Gao, and H. Li, "MonoDETR: Depth-aware transformer for monocular 3D object detection," *arXiv preprint arXiv:2203.13310*, 2022.

[32] Y. Lu, X. Ma, L. Yang, T. Zhang, Y. Liu, Q. Chu, J. Yan, and W. Ouyang, "Geometry uncertainty projection network for monocular 3D object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 3111–3121.

[33] K.-C. Huang, T.-H. Wu, H.-T. Su, and W. H. Hsu, "MonoDTR: Monocular 3D object detection with depth-aware transformer," in *CVPR*, 2022.

[34] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.

[35] V. Guizilini, J. Li, R. Ambrus, S. Pillai, and A. Gaidon, "Robust semi-supervised monocular depth estimation with reprojected distances," in *Conference on Robot Learning (CoRL)*, October 2019.

[36] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[37] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.

[38] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the*

IEEE conference on computer vision and pattern recognition, 2017, pp. 1851–1858.

[39] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, pp. 600 – 612, 05 2004.

[40] A. Simonelli, S. R. Bulo, L. Porzi, M. L. Antequera, and P. Kontschieder, "Disentangling monocular 3D object detection: From single to multi-class recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[41] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4340–4349.

[42] T. Wang, Z. Xinge, J. Pang, and D. Lin, "Probabilistic and geometric depth: Detecting objects in perspective," in *Conference on Robot Learning*. PMLR, 2022, pp. 1475–1485.

[43] J. Huang, G. Huang, Z. Zhu, and D. Du, "BEVDet: High-performance multi-camera 3D object detection in bird-eye-view," *arXiv preprint arXiv:2112.11790*, 2021.

[44] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "BEVformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," *arXiv preprint arXiv:2203.17270*, 2022.

[45] Y. Liu, T. Wang, X. Zhang, and J. Sun, "PETR: Position embedding transformation for multi-view 3D object detection," *arXiv preprint arXiv:2203.05625*, 2022.

[46] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 697–12 705.

[47] Y. Chen, L. Tai, K. Sun, and M. Li, "MonoPair: Monocular 3D object detection using pairwise spatial relationships," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 093–12 102.

[48] J. M. U. Vianney, S. Aich, and B. Liu, "RefinedMPL: Refined monocular pseudoLiDAR for 3D object detection in autonomous driving," *arXiv preprint arXiv:1911.09712*, 2019.

[49] M. Ding, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu, and P. Luo, "Learning depth-guided convolutions for monocular 3D object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 1000–1001.

[50] G. Brazil, G. Pons-Moll, X. Liu, and B. Schiele, "Kinematic 3D object detection in monocular video," in *European Conference on Computer Vision*. Springer, 2020, pp. 135–152.

[51] Y. Zhou, Y. He, H. Zhu, C. Wang, H. Li, and Q. Jiang, "Monocular 3D object detection: An extrinsic parameter free approach," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7556–7566.

[52] Y. Zhang, J. Lu, and J. Zhou, "Objects are different: Flexible monocular 3D object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3289–3298.

[53] Y. Lu, X. Ma, L. Yang, T. Zhang, Y. Liu, Q. Chu, J. Yan, and W. Ouyang, "Geometry uncertainty projection network for monocular 3D object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3111–3121.

[54] Y.-N. Chen, H. Dai, and Y. Ding, "Pseudo-Stereo for monocular 3D object detection in autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[55] Z. Li, Z. Qu, Y. Zhou, J. Liu, H. Wang, and L. Jiang, "Diversity matters: Fully exploiting depth clues for reliable monocular 3D object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2791–2800.

[56] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "MS-COCO: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[57] Y. Lee and J. Park, "CenterMask: Real-time anchor-free instance segmentation," 2020.