2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Every Dataset Counts: Scaling up Monocular 3D Object Detection with Joint Datasets Training

Fulong Ma, Xiaoyang Yan, Guoyang Zhao, Xiaojie Xu, Yuxuan Liu, Jun Ma, and Ming Liu

***Abstract*— Monocular 3D object detection is essential for autonomous driving. However, current monocular 3D detection algorithms rely on expensive 3D labels from LiDAR scans, making it difficult to use in new datasets and unfamiliar environments. This study explores training a monocular 3D object detection model using a mix of 3D and 2D datasets. The proposed framework includes a robust monocular 3D model that can adapt to different camera settings, a selective-training strategy to handle varying class annotations in datasets, and a pseudo 3D training method using 2D labels to improve detection ability in scenes with only 2D labels (as shown in Fig. 1). By utilizing this framework, we can train models on a combination of 3D and 2D datasets to improve generalization and performance on new datasets with only 2D labels. Extensive experiments on KITTI, nuScenes, ONCE, Cityscapes, and BDD100K datasets showcase the scalability of our proposed approach. Here is our project page: https://sites.google.com/view/fmaafmono3d.**

I. Introduction

Precise 3D understanding of the surrounding environment is the cornerstone of fields such as robotics and autonomous driving. In recent years, 3D detection algorithms using LiDAR point clouds have demonstrated outstanding performance, attributed to the precise ranging capabilities of LiDAR [1]. However, LiDAR is costly and not conducive to large-scale practical applications. In comparison to LiDAR, cameras offer advantages such as low cost, energy efficiency, rich color information, and compact size, providing greater flexibility in installation. These advantages have led to their widespread use in the fields of robotics and autonomous driving, making 3D detection using monocular cameras an increasingly promising research area in robotics and computer vision.

This work was supported by the Guangzhou-HKUST(GZ) Joint Funding Scheme under Grant 2024A03J0618. *(Corresponding Author: Jun Ma.)*

Fulong Ma, Guoyang Zhao, Xiaojie Xu, and Ming Liu are with The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China. (e-mail:{fmaaf, gzhao492, xxu763, eelium}@connect.hkust-gz.edu.cn).

Xiaoyang Yan and Yuxuan Liu are with The Hong Kong University of Science and Technology, Hong Kong SAR, China. (e-mail: {xyanaq, yliuhb}@ust.hk).

Jun Ma is with the Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China, and also with the Division of Emerging Interdisciplinary Areas, The Hong Kong University of Science and Technology, Hong Kong SAR, China. (e-mail: jun.ma@ust.hk).

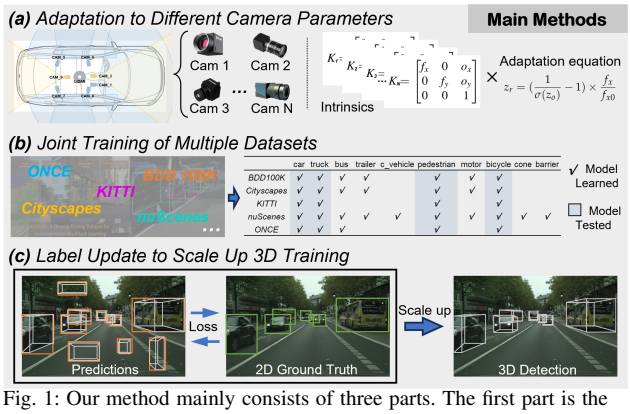


Fig. 1: Our method mainly consists of three parts. The first part is the camera parameter adaptation module, which handles different camera parameters to mitigate their impact. The second part is multi-dataset joint training, where we pre-train the model using as many datasets as possible to enhance its feature extraction capability. The third part involves leveraging 2D annotation information to assist the training of the 3D detection model, enabling good detection performance even in the absence of 3D annotation information.

In recent years, there has been a notable advancement in monocular 3D object detection [2]. Models utilizing Bird's Eye View (BEV) representation have shown increased effectiveness in scenarios requiring multi-sensor fusion, commonly found in autonomous driving applications. On the other hand, models using front-view representation, which is the natural representation of camera images, are not only quicker but also easier to implement.

Despite the progress made, the implementation of monocular 3D object detection still faces significant obstacles, with data being a primary issue [3]. When attempting to deploy a 3D detection model on a robot equipped with a single camera in a new setting, acquiring 3D data labeled with LiDAR points poses a challenge, and refining the model with data collected from the robot is not feasible. As such, it is crucial to rely on the pretrained model's ability to generalize effectively with minimal zero-shot fine-tuning.

Given these obstacles, we propose strategies aimed at enhancing the implementation of monocular 3D detection models by making efficient use of available data. Initially, we investigate the complexities of training vision-based 3D detection models using a diverse range of public 3D datasets with different camera settings.

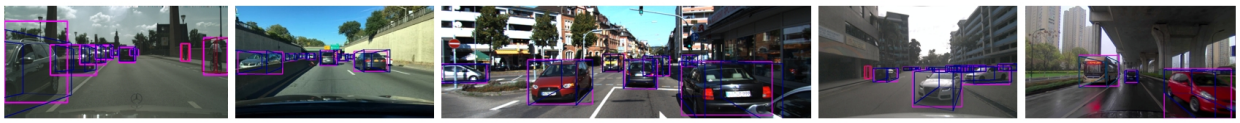


Fig. 2: Visualizations of the detection results of our method on five different datasets: BDD100K, Cityscapes, KITTI, nuScenes, and ONCE.

Through this process, we develop an output representation for the monocular 3D detection model that is unaffected by variations in camera settings and simultaneously create a framework for training models on datasets with diverse annotations. This method has the potential to significantly increase the scalability of models for individual researchers and developers.

Following this, we propose an approach for training 3D models using 2D labels. Various existing techniques, such as MonoFlex [4], annotate objects on the heatmap based on the projection of the 3D center onto the image rather than the center of 2D bounding boxes. We develop a training methodology that allows these models to be fine-tuned using 2D labels. This strategy enables us to refine existing models using more cost-effective 2D-labeled data collected from on-site robots, facilitating the transfer of 3D knowledge from public 3D datasets to the target environment. In this paper, we will present experiments conducted to fine-tune a pre-trained model using KITTI 2D data and Cityscapes 2D data as a demonstration of the potential effectiveness of our proposed approach.

Furthermore, through the integration of the two aforementioned methods, we can greatly enhance the scope of our monocular 3D detection models by training them on a combination of diverse public 3D/2D datasets. This approach significantly increases the volume of data utilized during training, leading to a notable improvement in the model's generalization capabilities. We commence by pre-training a model on a merged collection of KITTI, nuScenes, ONCE, Cityscapes, and BDD100K datasets, followed by fine-tuning it on the target dataset using solely 2D labels. Subsequently, we evaluate the model's ability to generalize on the target dataset. Our algorithm's qualitative results on the five well-known datasets are depicted in the Fig. 2.

Our main contributions are as follows:

- A robust output representation for models like MonoFlex [4] was developed to accommodate different camera intrinsic parameter settings lays the foundation for training models on diverse datasets.
- We proposed a novel method for training and fine-tuning monocular 3D detection models on mixed 2D and 3D datasets. Enhanced the model's generalization performance and reduced the dependence on costly 3D labels.
- Comprehensive experiments were conducted on the joint dataset comprising KITTI, nuScenes, ONCE, Cityscapes, and BDD100K. The results of the experiments demonstrate the effectiveness of our method. Compared to zero-shot settings, our approach achieved a significant performance improvement.

II. Related Works

A. Monocular 3D Object Detection

Bird-Eye-View Methods: These techniques focus on performing monocular 3D detection directly in 3D spaces, simplifying the output representation design. However, the main difficulties lie in converting perspective-view images to 3D coordinates features. In [5]–[8], researchers first predict depth information from monocular images, and then perform 3D object detection based on the predicted depth information. Another method directly conducts differentiable feature transformation, creating 3D features from image features, enabling end-to-end training of 3D detection in 3D spaces [9]. These methods often address the scale-ambiguity issue through depth prediction sub-networks or attention modules. However, inference speed in the BEV space heavily depends on 3D labels from LiDAR or direct supervision from LiDAR data, making it difficult to leverage existing 2D datasets or cost-effective 2D labeling tools. As a result, researchers lacking access to 3D labeled data may face challenges when deploying or fine-tuning networks in new environments.

Perspective-View Methods: These methods perform monocular 3D detection directly in the original perspective view, which is more intuitive. Many single-stage monocular 3D object detection methods are built upon existing 2D object detection frameworks. The main challenge here lies in designing robust and accurate encoding/decoding methods to bridge 3D predictions and dense perspective view features. Various techniques have been proposed, such as SS3D [10], which adds additional 3D regression parameters; ShiftRCNN [11], which introduces an optimization scheme. Other works like M3DRPN [12], D4LCN [13], and YoloMono3D [14] use statistical priors in anchors to improve the accuracy of 3D regression. Additionally, SMOKE [15], RTM3D [16], Monopair [17], and KM3D [18] leverage heatmap-based keypoint predictions, combined with anchor-free object detection frameworks like CenterNet [19].

Despite impressive progress in monocular 3D detection, most research still focuses on training within

a single homogeneous dataset, leading to overfitting and poor generalization of models in specific camera settings. In this study, we propose a more robust output representation based on MonoFlex [4], enabling the network to be trained on different datasets. We also introduce a training strategy for performing 3D object detection on 2D datasets, enhancing the model's generalization and reducing the annotation cost for 3D object detection.

B. Weakly Supervised 3D Object Detection

The weakly-supervised approach is also one of the methods aimed at reducing dependency on annotated data. Autolabels [20] present an automatic annotation pipeline to recover 9D cuboids and 3D shapes from pre-trained off-the-shelf 2D detectors and sparse LIDAR data. WS3D [21] introduced a weakly supervised approach for 3D LiDAR object detection in two stages. Initially, cylindrical object proposals are generated by manual annotations in the bird's eye-view. Subsequently, the network refines these proposals using a small set of precisely labeled object instances to produce the final 3D object bounding boxes. WeakM3D [22] first detected objects in images and combined them with 3D point cloud data to obtain object-LiDAR-points. Furthermore, it proposed a method to estimate object orientation θ by determining the orientation of each point pair in the object-LiDAR-points. WeakMono3D [23] introduce a new labeling method called 2D direction label, replacing the 3D rotation label in point clouds data and a direction consistency loss based on the new labels. Compared to the aforementioned weakly supervised methods, our approach does not require point clouds assistance, which reduces the complexity of sensors. It also does not rely on multi-view images, avoiding errors in the spatial position of objects caused by imprecise poses between multi-view images.

III. Methods

A. Camera Aware Monoflex Detection Baseline

Monocular 3D object detection involves estimating the 3D location center (x, y, z), dimensions (w, h, l) and planar orientation θ of objects of interest with a single image. Since most SOTA detectors perform prediction in the camera's front-view, we generally predict the projection of the object center on the image plane (c_x, c_y) instead of the 3D position (x, y). The orientation θ is further replaced with the observation angle

$$\alpha = \theta - \arctan(\frac{x}{z}), \tag{1}$$

which better conforms with the visual appearance of the object [4], [12].

MonoFlex [4] is an anchor-free method. It extracts feature maps from the input images with a DLA [24] backbone, similar to CenterNet [19] and KM3D [18]. As an anchor-free algorithm, MonoFlex predicts the center positions of target objects with a heat map. Monoflex primarily consists of the following components: 2D detection, dimension estimation, orientation estimation, keypoint estimation and depth estimation ensemble. The depth prediction z is simultaneously estimated from both geometry and direct prediction, and these predictions are adaptively ensembled to obtain the final result.

Since we aim to perform joint training on diverse datasets, where data collection involves different cameras with distinct camera settings, our method needs to overcome the challenges posed by varying camera parameters in order to facilitate effective knowledge transfer across these datasets. Given that MonoFlex exhibits insensitivity to camera parameters, we build upon MonoFlex as the foundation of our approach.

Overall, our approach is similar to MonoFlex, with the main difference being that our method has modifications in the depth prediction component, which takes camera parameters into account. Specifically, in the original MonoFlex paper, the direct regression part of depth prediction assumes an absolute depth of

$$z_r = \frac{1}{\sigma(z_o)} - 1, \tag{2}$$

where z_o is the unlimited network output following [17], [19] and

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{3}$$

While in our method, we have improved it by taking camera's parameters into account as follows:

$$z_r = (\frac{1}{\sigma(z_o)} - 1) \times \frac{f_x}{f_{x0}}. \tag{4}$$

where f_x is epresents the focal length of the camera used in the training dataset, while f_{x0} is a hyperparameter, we set its value to 500 in our experiments.

B. Selective Training for Joint 3D Dataset Training

During dataset training, some parts of the dataset are incompletely annotated. For instance, in the case of KITTI, certain categories like "Tram" are not labeled. However, it is not appropriate to treat the data from KITTI as negative samples for the "Tram" category. Therefore, each data point, in addition to its own annotations, is associated with the categories labeled in the respective dataset. This association provides supervision for the network's classification output and is applied only to the categories with annotations. Specifically, for different 3D datasets with varying labeled categories, each data frame stores the categories currently labeled in the dataset. When calculating the loss, we do not penalize (suppress) the detection predictions for unannotated

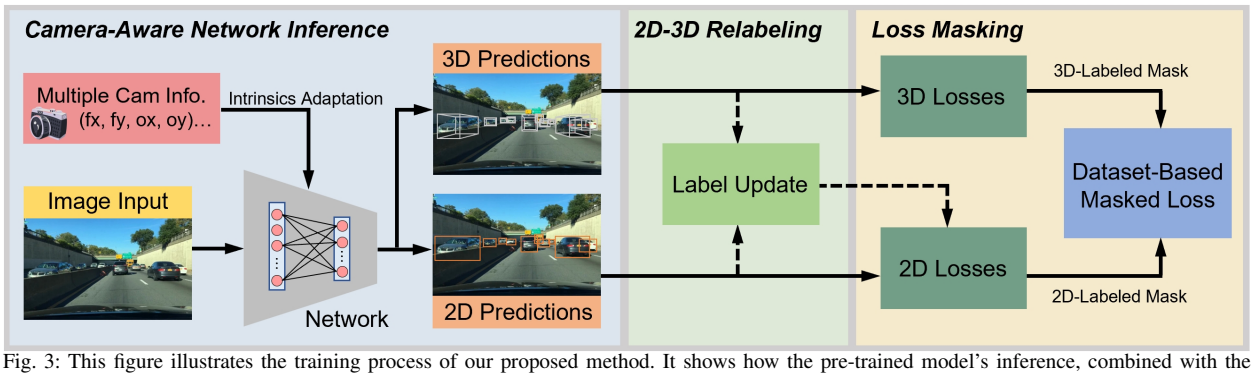


Fig. 3: This figure illustrates the training process of our proposed method. It shows how the pre-trained model's inference, combined with the 2D annotations from the dataset, facilitates the training of a 3D detection model on datasets that lack 3D training labels.

categories. In summary, this approach effectively handles the issue of incomplete annotations during dataset training. It ensures that only annotated categories influence the model's training, and unannotated categories do not lead to erroneous model behavior. The training procedure is shown in Fig. 3.

C. Regulating 2D Labels of 2D Datasets for Pseudo 3D training

In the Monoflex method, the center of the heatmap is determined by projecting the 3D center. For data with only 2D annotations, we have no direct means of generating supervision signals for the object's 3D center. Therefore, we need to find a way to generate 3D detection supervision information from 2D labels. We propose a novel method to train 3D detection algorithms solely relying on 2D detection labels. Specifically, we start by feeding data with only 2D annotations into a pre-trained 3D detection model and set a very low score threshold to enable the model to produce multiple detection results (including both 2D and 3D detection results). This step may include some erroneous or inaccurate detections. Next, we use the 2D training labels from the new dataset to match them with the 2D detection results obtained in the previous step, filtering out erroneous or inaccurate detections to obtain pseudo 3D training labels. Finally, we reconstruct the ground truth heatmap and 2D detection map (as shown in Fig. 4), and ultimately, we calculate the loss between the model predictions and the pseudo 3D labels only on the heatmap and 2D detection map. The method of training a monocular 3D model using 2D annotated data is illustrated in Algorithm 1, we refer to it as *Pseudo 3D Training with 2D Labels*.

Algorithm 1 Pseudo 3D training with 2D Labels

Input: Dense Detection Maps F, Labeled 2D boxes B_t
Output: Loss l

Initialization:
B_p : Detection results from dense detection maps
B_t' : Pseudo ground truth
M : IoU matrix
Main Loop:
Retrieve detection results B_p from dense detection maps F.
Compute IoU matrix M between B_p and B_t.
Compute the matching with the minimum cost, take 3D centers of B_p as ground truth for B_t'.
for each matched b_p, b_t **do**
    if $cost_i > eps$ **then**
        Remove mis-detection box from B_t'.
    end if
end for
Reconstruct ground truth heatmaps and 2D detection maps F_t from B_t'.
Compute Loss l with pseudo ground truth B_t' only on heatmaps and 2D detection maps.

IV. Experiments

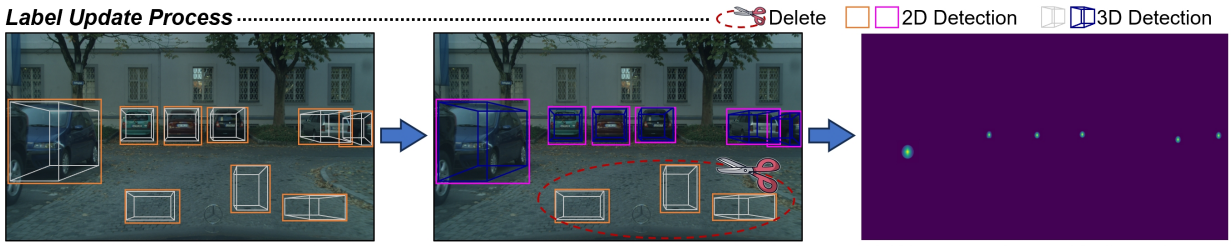
A. Datasets Reviews

We evaluate the proposed networks on the KITTI 3D object detection benchmark [25] and Cityscapes dataset [26]. The KITTI dataset consists of 7,481 training frames and 7,518 test frames. Chen *et al.* [27] split the training set into 3,712 training frames and 3,769 validation frames. The Cityscapes dataset contains 5000 images split into 2975 images for training, 500 images for validation, and 1525 images for testing.

B. Evaluation Metrics

1) KITTI 3D: All the testing and validation results, are evaluated with 40 recall positions (AP_{40}), following Simonelli *et al.* [28] and the KITTI team. Such a protocol is considered to be more stable than the AP_{11} proposed in the Pascal VOC benchmark [29].

2) Cityscapes 3D: Following [26], we use these five metrics: 2D Average Precision (AP), Center Distance ($BEVCD$), Yaw Similarity ($YawSim$), Pitch-Roll Similarity ($PRSim$), Size Similarity ($SizeSim$) and Detection Score (DS) to evaluate the performance on Cityscapes 3D dataset. Among them, DS is a com-



(a) Retrieve detection results from dense detection map.

(b) Filter out the misdetections.

(c) Reconstruct ground truth heatmaps.

Fig. 4: The figure depicts the training label update process. In the left image, the pre-trained 3D detection model's predictions on new data are shown, which may include some erroneous detections, as indicated by the green boxes. The middle image illustrates the process of identifying and filtering out these erroneous detections, marking them in gray based on the matching results. The right image represents the reconstruction of the ground truth heatmap using the pseudo 3D labels.

TABLE I: Detection results on the "Car", "Pedestrian", and "Cyclist" categories on the KITTI dataset.

	Methods	Car			Pedestrain			Cyclist		
		Easy (%)	Moderate (%)	Hard (%)	Easy (%)	Moderate(%)	Hard (%)	Easy (%)	Moderate(%)	Hard (%)
$2D$	Zero-shot	93.94	81.92	71.75	57.53	50.24	44.36	53.65	46.24	40.18
	Ours	99.54	96.32	88.60	82.62	76.59	67.54	87.67	79.45	72.59
	Improvement	+5.60	+14.40	+16.85	+26.09	+26.35	+23.18	+34.02	+33.21	+32.41
$3D$	Zero-shot (3D)	30.44	23.19	20.21	6.80	5.82	5.07	0.64	0.37	0.35
	Ours (3D)	48.99	33.29	28.41	14.09	12.95	10.79	1.57	0.84	0.81
	Improvement	+18.55	+10.10	+8.20	+7.29	+7.13	+5.72	+0.93	+0.47	+0.46
	Zero-shot (BEV)	38.20	28.85	25.42	7.50	6.44	5.43	1.29	0.82	0.53
	Ours (BEV)	56.87	39.17	33.79	15.40	13.72	11.37	1.86	1.06	1.01
	Improvement	+18.67	+10.32	+8.37	+7.90	+7.28	+5.94	+0.57	+0.24	+0.48

TABLE II: Object detection results of category "Car" on the Cityscapes dataset.

Methods	Metrics					
	DS (%)	AP (%)	BEVCD (%)	YawSim (%)	PRSim(%)	SizeSim (%)
Zero-shot	32.92	36.44	95.73	90.12	99.98	75.52
Ours	56.94	61.49	96.42	92.27	99.98	81.70
Improvement	+24.02	+25.05	+0.69	+2.15	0	+6.18

TABLE III: Object detection results of category "Truck" on the Cityscapes dataset.

Methods	Metrics					
	DS (%)	AP (%)	BEVCD (%)	YawSim (%)	PRSim(%)	SizeSim (%)
Zero-shot	10.26	11.47	93.64	99.87	99.98	64.55
Ours	23.38	25.18	94.49	99.93	99.98	77.05
Improvement	+13.12	+13.71	+0.85	+0.06	0	+12.50

TABLE IV: Object detection results of category "Bicycle" on the Cityscapes dataset.

Methods	Metrics					
	DS (%)	AP (%)	BEVCD (%)	YawSim (%)	PRSim(%)	SizeSim (%)
Zero-shot	0.02	0.03	93.14	72.42	99.98	52.91
Ours	2.37	2.80	96.64	77.63	99.98	64.65
Improvement	+2.35	+2.77	+3.50	5.21	0	+11.74

bination of the first five metrics and computed as:

$$DS = AP \times \frac{BEVCD + YawSim + PRSim + SizeSim}{4}. \quad (5)$$

For details, please refer to the paper [26].

C. Experiment Setup

To demonstrate the effectiveness of our method, we conducted extensive experiments on the KITTI and Cityscapes datasets. When testing on the KITTI dataset, we designed our experiments as follows:

- We initially pre-trained our model on four datasets: BDD100K, nuScenes, ONCE, and Cityscapes.
- With the pre-trained model, we evaluated its zero-shot detection performance on the KITTI dataset.
- Subsequently, we fine-tuned the model using our method, which involves training a 3D detection model using 2D training labels from KITTI.
- Finally, we obtained detection results on the KITTI dataset based on our method.

When testing on the Cityscapes dataset, we followed

TABLE V: Comparison with weakly-supervised methods on the "Car" category on KITTI dataset.

Methods	AP_{BEV} / AP_{3D}		
	Easy (%)	Moderate (%)	Hard (%)
VS3D [21]	31.59/22.62	20.59/14.43	16.28/10.91
WeakM3D [22]	58.20/50.16	38.02/29.94	30.17/23.11
Autolabels [20]	50.51/38.31	30.97/19.90	23.72/14.83
WeakMono3D [23]	54.32/**49.37**	**42.83/39.01**	**40.07/36.34**
Ours	**56.87**/48.99	39.17/33.29	33.79/28.41

Fig. 5: The qualitative results on the Cityscapes dataset. The leftmost column contains the original images, the middle column displays the zero-shot results, and the rightmost column shows the results obtained using our method. The pink boxes represent 2D detection results.

the same experimental setup on the KITTI dataset.

D. Experiment Results and Comparison

The quantitative results of the "Car", "Pedestrain" and "Cyclist" category on KITTI dataset are reported in Table I, while quantitative results of "Car", "Truck" and "Bicycle" category on Cityscapes dataset are shown in Table II, Table III, and Table IV respectively.

From Table I, we can observe that our method has achieved significant improvements in both 3D and 2D detection tasks compared to zero-shot learning. Specifically, for the 3D detection task, our method has shown an improvement in AP_{3D}/AP_{BEV} for the "Car" category in the Easy, Moderate, and Hard difficulty levels by $18.55/18.67$, $10.10/10.32$ and $8.20/8.37$, respectively. In the "Pedestrian" category, we achieved a progress in AP_{3D}/AP_{BEV} in the Easy, Moderate, and Hard difficulty levels by $7.29/7.90$, $7.13/7.28$ and $5.72/5.94$, respectively. And for the "Cyclist" category, we improved the AP_{3D}/AP_{BEV} by $0.93/0.57$, $0.47/0.24$ and $0.46/0.48$ in Easy, Moderate and Hard level, respectively. For the "Cyclist" class, although there is not a significant increase in the number of 3D detection points, the improvement in AP3D/APBEV across three different difficulty levels ranges from 29.27% to 145.32%. Therefore, the improvement from our method is significant. In summary, the experimental results demonstrate the effectiveness of our method on the KITTI dataset.

Table II, Table III and Table IV present the detection results for the "Car", "Truck" and "Bicycle" categories on the Cityscapes dataset. Quantitative results indicate that our method, when compared to the zero-shot approach, has shown significant improvements in various metrics, except for the PRSim metric (as we only focused on the yaw angle, considering pitch and roll angles to be 0). Specifically, in the "Car" category, we observed an increase of 25.05 in AP, a 0.69 improvement in BEVCD, a 2.15 improvement in YamSim, a 6.18 improvement in SizeSim, and a remarkable 24.02 enhancement in DS. In the "Truck" category, AP, BEVCD, YawSim, SizeSim, and DS have improved by 13.71, 0.85, 0.06, 12.50, and 13.12, respectively. And for the "Bicycle" class,AP, BEVCD, YawSim, SizeSim, and DS have improved by 2.77, 3.50, 5.21, 11.74, and 2.35, respectively. From the above experimental results, it can be seen that our method has also achieved significant performance improvement on the Cityscapes dataset. Fig. 5 provides a visual comparison between our method and the zero-shot approach. It can be observed that our method exhibits significantly improved detection capability compared to the zero-shot approach.

Table V illustrates the comparison between our method and several weakly-supervised methods on the "Car" class in the KITTI dataset. It is apparent that overall, our method falls short of the latest state-of-the-art algorithm, WeakMono3D [23]. However, it outperforms other weakly-supervised algorithms. It is worth mentioning that our approach relies solely on easily accessible 2D annotations, without requiring any 3D annotation information or additional sensor data such as point clouds or multi-view images for assistance.

V. Conclusion

In this paper, we initially conducted research on models such as MonoFlex [4] and developed strategies that are resilient to changes in camera intrinsics. These strategies allow the models to be trained on diverse datasets. Additionally, we designed a learning approach that enables monocular 3D detection models to acquire 3D detection knowledge based solely on 2D labels, even in datasets that only provide 2D training labels. Lastly, we carried out extensive experiments on a combination of datasets, including KITTI, nuScenes, Cityscapes, and others. The experimental results demonstrated the efficacy of our approach. Despite its success, our work does have limitations. First, our method is currently applicable only to algorithms that are insensitive to camera parameters, such as MonoFlex. For models where

the influence of camera parameters cannot be disregarded. Moreover, when encountering new categories in a novel dataset, the lack of relevant supervision from previous datasets may result in suboptimal detection performance for these new categories. In future work, we will explore how to reduce the algorithm's sensitivity to sensor parameters to make our method more versatile. Additionally, we will investigate open-vocabulary object detection to enhance the algorithm's detection performance on new categories.

REFERENCES

[1] Peng Yun, Lei Tai, Yuan Wang, and Ming Liu. Focal loss in 3d object detection. *CoRR*, abs/1809.06065, 2018.

[2] Xinzhu Ma, Wanli Ouyang, Andrea Simonelli, and Elisa Ricci. 3d object detection from images for autonomous driving: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[3] Liang Peng, Fei Liu, Zhengxu Yu, Senbo Yan, Dan Deng, Zheng Yang, Haifeng Liu, and Deng Cai. Lidar point cloud guided monocular 3d object detection. In *European Conference on Computer Vision*, pages 123–139. Springer, 2022.

[4] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3289–3298, June 2021.

[5] Xinshuo Weng and Kris Kitani. Monocular 3d object detection with pseudo-lidar point cloud. *CoRR*, abs/1903.09847, 2019.

[6] Jean Marie Uwabeza Vianney, Shubhra Aich, and Bingbing Liu. Refinedmpl: Refined monocular pseudolidar for 3d object detection in autonomous driving, 11 2019.

[7] Xinzhu Ma, Zhihui Wang, Haojie Li, Wanli Ouyang, and Pengbo Zhang. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. *CoRR*, abs/1903.11444, 2019.

[8] Jason Ku, Alex D. Pon, and Steven L. Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. *CoRR*, abs/1904.01690, 2019.

[9] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distributionnetwork for monocular 3d object detection. *CVPR*, 2021.

[10] Eskil Jörgensen, Christopher Zach, and Fredrik Kahl. Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss. *CoRR*, abs/1906.08070, 2019.

[11] Andretti Naiden, Vlad Paunescu, Gyeongmo Kim, ByeongMoon Jeon, and Marius Leordeanu. Shift R-CNN: deep monocular 3d object detection with closed-form geometric constraints. *CoRR*, abs/1905.09970, 2019.

[12] Garrick Brazil and Xiaoming Liu. M3D-RPN: monocular 3d region proposal network for object detection. *CoRR*, abs/1907.06038, 2019.

[13] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. 12 2019.

[14] Yuxuan Liu and Ming Liu. Yolostereo3d: A step back to 2d for efficient stereo 3ddetection. In *arXiv preprint arXiv:2102.15072*, 2021.

[15] Zechen Liu, Zizhang Wu, and Roland Tóth. SMOKE: Single-stage monocular 3d object detection via keypoint estimation. *arXiv preprint arXiv:2002.10111*, 2020.

[16] Pei-Xuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. *ArXiv*, abs/2001.03343, 2020.

[17] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] Peixuan Li. Monocular 3d detection with geometric constraints embedding and semi-supervised training, 2020.

[19] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019.

[20] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. Autolabeling 3d objects with differentiable rendering of sdf shape priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12224–12233, 2020.

[21] Zengyi Qin, Jinglu Wang, and Yan Lu. Weakly supervised 3d object detection from point clouds. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4144–4152, 2020.

[22] Liang Peng, Senbo Yan, Boxi Wu, Zheng Yang, Xiaofei He, and Deng Cai. Weakm3d: Towards weakly supervised monocular 3d object detection. *arXiv preprint arXiv:2203.08332*, 2022.

[23] Runzhou Tao, Wencheng Han, Zhongying Qiu, Cheng-zhong Xu, and Jianbing Shen. Weakly supervised monocular 3d object detection using multi-view projection and direction consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17482–17492, 2023.

[24] F. Yu, D. Wang, E. Shelhamer, and T. Darrell. Deep layer aggregation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018.

[25] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[26] Nils Gählert, Nicolas Jourdan, Marius Cordts, Uwe Franke, and Joachim Denzler. Cityscapes 3d: Dataset and benchmark for 9 dof vehicle detection. *arXiv preprint arXiv:2006.07864*, 2020.

[27] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 424–432. Curran Associates, Inc., 2015.

[28] Andrea Simonelli, Samuel Rota Bulò, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. *CoRR*, abs/1905.12365, 2019.

[29] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.