DINO-SD for Robust Multi-View Supervised Depth Estimation

Yifan Mao* Ming Li* Jian Liu* Jiayang Liu Zihan Qin Chunxi Chu
Jialei Xu Wenbo Zhao Junjun Jiang Xianming Liu†
Harbin Institute of Technology
maoyf1105@163.com 2021111347@stu.hit.edu.cn hitcslj@stu.hit.edu.cn
1190200503@stu.hit.edu.cn 1201021818@stu.hit.edu.cn 3045357158@qq.com
xujialei@stu.hit.edu.cn wbzhao@hit.edu.cn jiangjunjun@hit.edu.cn csxm@hit.edu.cn

Abstract

This technical report summarizes the champion solution for the RoboDepth Challenge, which is held in the ICRA 2024 RoboDrive Workshop. DINO-SD is a multi-view supervised depth estimation model. Our model primarily focuses on addressing robustness issues in corrupted environments of autonomous driving. We use pretrained DINOv2 [1] as the backbone, M-DPT, and DPT [2] as a decoder. To enhance the robustness of the model, we have chosen AugMix [3] as our data augmentation strategy. Additionally, in the testing pipeline, we have implemented denoising [4] and equalization for the images. In Track 4 of the 2024 RoboDrive Challenge [5], our model achieved an Abs Rel of 0.187, which is the SOTA result on this dataset. The other six metrics also achieved SOTA performance.

1. Overview

Depth estimation, which aims to estimate the distance of every point in the image, is a crucial task in 3D vision with important applications such as autonomous driving [6–8], augmented reality [9], virtual reality [10], and 3D reconstruction [11, 12]. Compared to acquiring depth using depth sensors such as Lidar [13–15], estimating depth from images can effectively reduce hardware costs and produce dense depth maps. This makes depth estimation algorithms the primary choice in these fields.

Depth estimation tasks can be categorized into monocular depth estimation and multi-view depth estimation according to the number of cameras used. Monocular depth estimation is inherently limited by its reliance on a single viewpoint, which compromises its robustness. In contrast, multi-view depth estimation provides a comprehensive 360° view of the surroundings, which can more accurately estimate depth and is robust to changes in scene geometry. As sensor technologies advance and manufacturing costs decrease, multi-view depth estimation has progressively replaced monocular depth estimation as the industry standard. Notable works in this field include MVSNet [16], SurroundDepth [17], and S3Depth [18].

However, existing multi-view depth estimation methods still do not perform satisfactorily in real-world scenarios [19]. The main reason is that real-world sensor data often contains corruptions, such as adverse weather and sensor noise, and most autonomous driving training datasets primarily consist of clean data. Existing methods [17, 18] lack robustness to noise. Several studies have focused on enhancing the robustness of depth estimation [20–23] through the use of additional training data in different scenes. However, the introduction of additional data still cannot cover all situations in real-world scenarios. Given the prohibitive cost of acquiring large volumes of corrupted data, and considering that expanding training datasets may not encompass all real-world corruptions, developing a robust multi-view depth estimation model capable of performing well on out-of-distribution (OoD) data is imperative.

Inspired by the Depth Anything [24] framework, we introduce DINO-SD, a novel approach aimed at improving the robustness of surround-view depth estimation models. This framework is designed to handle a variety of environmental conditions and sensor imperfections, enhancing the reliability of depth estimation in autonomous driving and other critical applications.

2. Technical Approach

2.1. Overview

Given 6 surrounding views $I_s \in \mathbb{R}^{6\times3\times H\times W}$. The goal of the proposed DINO-SD is to output 6 corresponding depth maps $D_s \in \mathbb{R}^{6\times1\times H\times W}$. As shown in Fig. 1, the proposed

*Equal contribution.
†Corresponding author.
Technical Report of the 2024 RoboDrive Challenge.
Track 4: Robust Depth Estimation.

DINO-SD encompasses three principal phases: feature extracting, fuse and decoding, and depth estimation. In the following, we will introduce the three phases in detail.

2.2. DINO-SD

Our DINO-SD uses the pretrained DINOv2 [1] as the encoder, M-DPT, and DPT [2] as a decoder. The reason we use Dinov2 is that Dinov2 can extract robust image features compared with other encoders. It helps to improve the model performance when processing the OoD data.

Furthermore, we choose DPT [2] as our decoder. We also modify the structure of DPT to adapt for surround-view depth estimation and propose the Multiview-DPT (M-DPT) as shown in figure 1. We introduce the adjacent-view attention into DPT.

In the previous surround-view depth estimation task, SurroundDepth [17] uses the cross-view self-attention while S3Depth[18] uses the adjacent-view cross-attention.

Let $F_i \in \mathbb{R}^{N \times C \times \frac{H}{n} \times \frac{W}{n}}, i = 1, 2, 3, 4, 5, 6$ be the feature maps obtained from *i*-th view, where H and W indicate the height and width of the input images, N represents the batch size and C stands for the dimensions of the feature map. For self attention, the feature maps F_i are concatenated and reshaped into $F \in \mathbb{R}^{N \times \frac{6HW}{n^2} \times C}$ and then used to compute the *Q*, *K*, *V* from F. The formula 1 shows the calculation process.

$$\begin{aligned} Q &= W_Q F \\ K &= W_K F \\ V &= W_V F \\ F &= sofrmax(\frac{Q^T K}{\sqrt{C}})V \end{aligned} \tag{1}$$

For adjacent-view cross attention, *K* and *V* are computed from the adjacent-view feature maps $F_j, j \in (i-1, i+1)$. The feature maps F_i are shaped into $F_i \in \mathbb{R}^{N \times \frac{HW}{n^2} \times C}$ and then used to computed the *Q*, *K*, *V*. The formula 2 shows the calculation process.

$$\begin{aligned} Q &= W_Q F_i \\ K &= W_K F_j \\ V &= W_V F_j \\ F_i &= softmax(\frac{Q^T K}{\sqrt{C}})V \end{aligned} \tag{2}$$

S3Depth's performance is better than SurroundDepth, so we use the adjacent-view cross attention. We also try the cross-view self attention but the results show that adjacent-view cross attention is better than cross-view self attention. More details can be found in section 3. To introduce self-attention or cross-attention into DPT, we perform a self-attention or cross-attention operation before the feature maps are fed into the Fusion module of DPT. More details for DPT structure can be found in [2].

Our depth head is very simple, containing only two convolutional layers and a sigmoid head.

2.3. Training Pipeline

Our model training pipeline is shown in the figure 2a. For surrounding-view images $I_s \in \mathbb{R}^{6 \times 3 \times H \times W}$, we first use DINOv2 to extract the feature maps $F \in \mathbb{R}^{6 \times \frac{HW}{n^2} \times C}$ and then we use M-DPT and DPT to decode the feature maps. Finally, we use a depth head to get the depth map $D \in \mathbb{R}^{6 \times 1 \times H \times W}$. We use the LiDAR ground truth to offer supervision for depth maps. We adopt the silog loss to depth supervision:

$$L_{silog} = \frac{1}{n}\sum_i d_i^2 - \frac{\lambda}{n^2}(\sum_i d_i)^2, \tag{3}$$

where $d_i = \log y_i - \log y_i^*$, y_i represents the predicted depth of i-th pixel and y_i^* represents the ground truth depth of i-th pixel. We set $\lambda = 0.85$.

Furthermore, we use the AugMix loss to make the model able to process OoD data. AugMix data augmentation method [3] which mixs the augmented images. We first use AugMix to process the surrounding-view images I_s and get the augmented images I_a. Then, we use our DINO-SD to estimate the depth maps D_a of the augmented images I_a, and use the AugMix loss to make D_s and D_a have similar distributions. In our experiments, we perform two separate AugMix operations on I_s to obtain D_{a1} and D_{a2}, and calculated the difference between the D_s, D_{a1}, D_{a2} distributions by JS divergence:

$$\begin{aligned} D_{mix} &= \frac{1}{3}(D_s + D_{a1} + D_{a2}), \\ L_{AugMix} &= \frac{1}{3}(KL(D_s || D_{mix}) + KL(D_{a1} || D_{mix}) \\ &\quad + KL(D_{a2} || D_{mix})), \end{aligned} \tag{4}$$

where D_{mix} represents the mixed depth map and KL represents the KL divergence.

We also use the smooth loss to maintain the depth map consistency:

$$L_{smooth} = \sum_i |\partial_x d_i^*| e^{-|\partial_x I_i|} + |\partial_y d_i^*| e^{-|\partial_y I_i|}, \tag{5}$$

where $d_i^* = d_i / \overline{d_i}$.

Our loss is the combination of silog loss, augmix loss, and smooth loss:

$$L = L_{silog} + \alpha L_{smooth} + \beta L_{AugMix}, \tag{6}$$

we set $\alpha = 10^{-3}$ and $\beta = 10^{-2}$.

2.4. Testing Pipeline

Our testing pipeline is shown in figure2b. For surrounding-view OoD images, we perform image denoise and equalize

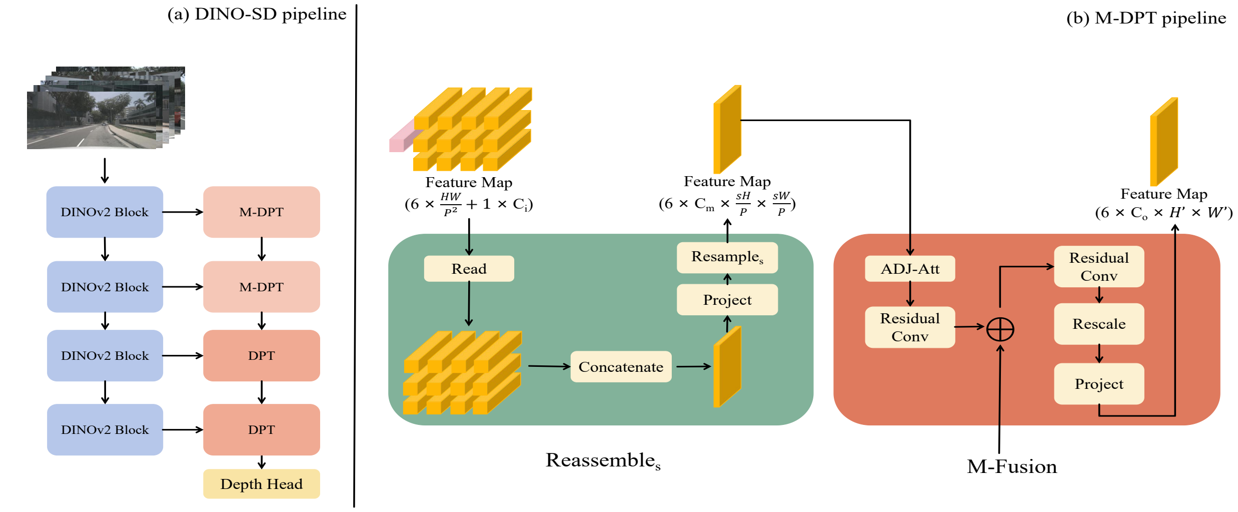
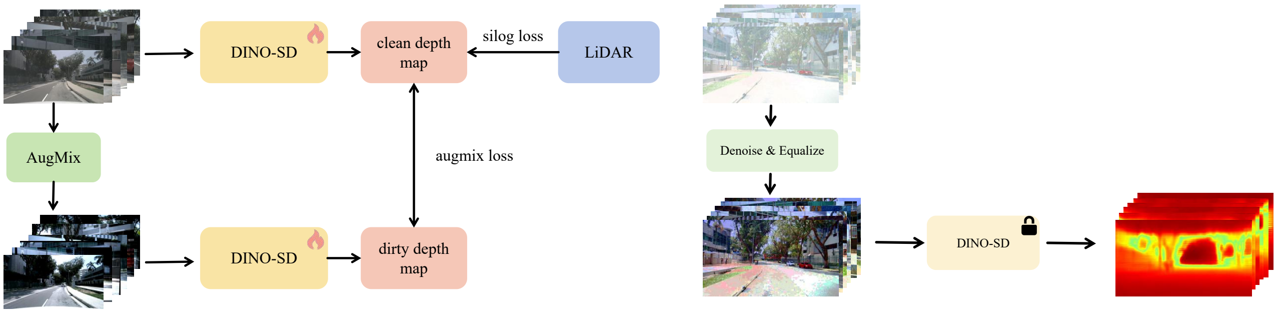


Figure 1. **Our DINO-SD model**: Our DINO-SD model use the pretrained DINOv2 as encoder, M-DPT and DPT as decoder.



(a) Training pipeline.

(b) Testing pipeline.

Figure 2. Our training and testing pipeline.

operations on the OoD images and then input into our trained DINO-SD. We adopt donoho *et al*. image denoise method[4]. We did not use model ensemble methods to improve the performance of our method.

3. Experiments

3.1. Datasets

This work follows the protocol in the 2024 RoboDrive Challenge [5] when preparing the training and test data. Specifically, the model is trained on the official *train* split of the nuScenes dataset [25] and tested on the held-out competition evaluation sets. The evaluation data was created following RoboDepth [19, 26, 27], RoboBEV [28–30], and Robo3D [31, 32]. The corruption types are mainly from three sources, namely camera corruptions, camera failures, and LiDAR failures. For more details, please refer to the corresponding GitHub repositories.

Implementation Details The model is implemented based on the MMDetection3D codebase [33]. The DINO-SD framework is implemented using PyTorch. It utilizes four NVIDIA GTX 3090 GPUs for model training, each configured with a batch size of 1x6 (for six views). On the leaderboard, our method secured first place across all six metrics. We use the DINOv2 [1] as the backbone, which is pretrained on a massive dataset named LVD-142M, comprising 142 million images. This dataset is assembled from ImageNet-22k, the training split of ImageNet-1k, Google Landmarks, and several fine-grained datasets. In our DINO-SD model, we use the last four blocks of the DINOv2 encoder to extract image features, the M-DPT and DPT to decode feature maps. For the penultimate third and penultimate fourth blocks, we processed the features using M-DPT, with the resample scale set to 1 and 0.5 respectively. For the penultimate and penultimate blocks, we processed the features using DPT with the resample scale set to 2 and 4, respectively. The learning rate

Table 1. Ablation results of DinoSurDepth on the RoboDrive cimpetition leaderboard.The symbol × indicates that the module was not used, the symbol ✓ indicates that the module was used, and the best results are highlighted in **bold**.

Method	attention	denoise	equalize	Abs Rel↓	Sq Rel↓	RMSE↓	log RMSE↓	$a1$ ↑	$a2$ ↑	$a3$ ↑
SurroundDepth(Baseline)	self attention	×	×	0.3039	3.0596	8.5285	0.4003	0.5439	0.7839	0.8911
DINO-SD	×	×	×	0.2654	2.5803	8.2702	0.3738	0.5873	0.8220	0.9128
DINO-SD	self attention	×	×	0.2162	1.9451	7.6721	0.3289	0.6702	0.8512	0.9279
DINO-SD	self attention	✓	×	0.2074	1.8087	7.4020	0.3145	0.6887	0.8588	0.9340
DINO-SD	self attention	×	✓	0.2078	1.7107	7.0786	0.3091	0.6844	0.8611	0.9370
DINO-SD	self attention	✓	✓	0.2052	1.7246	7.1974	0.3075	0.6900	0.8651	0.9395
DINO-SD	adjacent-view cross attention	✓	✓	**0.1870**	**1.4683**	**6.2365**	**0.2760**	**0.7339**	**0.8952**	**0.9531**

Table 2. Quantitative results on the Robodrive competition(Track 4).The **best** scores of each metric are highlighted in **bold**.

Team	Abs Rel↓	Sq Rel↓	RMSE↓	log RMSE↓	$a1$ ↑	$a2$ ↑	$a3$ ↑
HIT-AIIA[1]	**0.187**	**1.468**	**6.236**	**0.276**	**0.734**	**0.895**	**0.953**
twolones[2]	0.211	1.655	6.327	0.294	0.686	0.880	0.946
CUSTZS[3]	0.264	2.320	7.961	0.363	0.578	0.816	0.913

for the encoder is 5e-6 and the learning rate for the decoder is 2e-5. We employed the CosineAnnealingWarmRestarts method for our model's optimizer. This scheduler adjusts the learning rate following a cosine annealing pattern, periodically resetting to enhance convergence. We spent 18 hours training for 5 epochs, but ultimately found that the weights from the first epoch achieved the best results on the test set (corrupted images). We believe this was mainly due to the following reasons: Firstly, our batch size was relatively large (4 GPUs × 6 views), which led to the model converging quickly; Secondly, there was a significant distribution shift between the corrupted images and clean images, and learning too much from the clean images easily led to overfitting.

Comparative Study The benchmark uses the original depth of corruption images as ground truth for evaluation purposes. Corruptions are simulated through algorithms encompassing 18 types, namely: darkness, brightness, defocus blur, contrast, JPEG compression, impulse noise, motion blur, snow, zoom blur, frost, pixelation, color, quantization, elastic transformation, Gaussian noise, fog, ISO noise, shot noise, and glass blur. Table 2 compares the model's performance with that of other teams on the RoboDrive competition leaderboard.

Ablation Study In Table 1, we assess the impact of various repair algorithms, decoders, and backbones. The results demonstrate that all configurations yield improvements in the depth estimation task.

The first line is the official baseline SurroundDepth. We first try DINOv2 as encoder and DPT as decoder. The second line shows that although our DINO-SD do not use any attention mechanism, our DINO-SD performance is better than baseline. In line 3, we try to introduce self attention into DPT and the model performance improved a lot. In lines 4-6, we explore the impact of image denoise and equalization on the results. The results show that both image denoise and equalization help to improve the performance of the model, and the improvement effect is greater when used together. In line 7, we change the self attention into adjacent view cross attention and the results show that the adjacent-view cross attention is better than self attention. The reason adjacent-view cross attention performance better than self attention has explained in S3Depth[18], the adjacent views can offer direct environment information compared with non-adjacent views.

4. Summary

In this work, we propose the DINO-SD, a novel framework that focuses on improving the robustness of surround-view depth estimation model. We add attention mechanism into DPT and improved the performance of model. The results show that our method is able to handle various corruptions.

References

[1] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[2] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[3] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[4] David L Donoho and Iain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *biometrika*, 81(3):425–455, 1994.

[5] Lingdong Kong, Shaoyuan Xie, Hanjiang Hu, Yaru Niu, Wei Tsang Ooi, Benoit R. Cottereau, Lai Xing Ng, Yuexin Ma, Wenwei Zhang, Liang Pan, Kai Chen, Ziwei Liu, Weichao Qiu, Wei Zhang, Xu Cao, Hao Lu, Ying-Cong Chen, Caixin Kang, Xinning Zhou, Chengyang Ying, Wentao Shang, Xingxing Wei, Yinpeng Dong, Bo Yang, Shengyin Jiang, Zeliang Ma, Dengyi Ji, Haiwen Li, Xingliang Huang, Yu Tian, Genghua Kou, Fan Jia, Yingfei Liu, Tiancai Wang, Ying Li, Xiaoshuai Hao, Yifan Yang, Hui Zhang, Mengchuan Wei, Yi Zhou, Haimei Zhao, Jing Zhang, Jinke Li, Xiao He, Xiaoqiang Cheng, Bingyang Zhang, Lirong Zhao, Dianlei Ding, Fangsheng Liu, Yixiang Yan, Hongming Wang, Nanfei Ye, Lun Luo, Yubo Tian, Yiwei Zuo, Zhe Cao, Yi Ren, Yunfan Li, Wenjie Liu, Xun Wu, Yifan Mao, Ming Li, Jian Liu, Jiayang Liu, Zihan Qin, Cunxi Chu, Jialei Xu, Wenbo Zhao, Junjun Jiang, Xianming Liu, Ziyan Wang, Chiwei Li, Shilong Li, Chendong Yuan, Songyue Yang, Wentao Liu, Peng Chen, Bin Zhou, Yubo Wang, Chi Zhang, Jianhang Sun, Hai Chen, Xiao Yang, Lizhong Wang, Dongyi Fu, Yongchun Lin, Huitong Yang, Haoang Li, Yadan Luo, Xianjing Cheng, and Yong Xu. The robodrive challenge: Drive anytime anywhere in any condition. *arXiv preprint arXiv:2405.08816*, 2024.

[6] Markus Schön, Michael Buchholz, and Klaus Dietmayer. Mgnet: Monocular geometric scene understanding for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15804–15815, 2021.

[7] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 228–240, 2023.

[8] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030, 2023.

[9] Mehmet Kerim Yucel, Valia Dimaridou, Anastasios Drosou, and Albert Saa-Garriga. Real-time monocular depth estimation with sparse supervision on mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2428–2437, 2021.

[10] Yuyan Li, Zhixin Yan, Ye Duan, and Liu Ren. Panodepth: A two-stage approach for monocular omnidirectional depth estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 648–658. IEEE, 2021.

[11] Xingbin Yang, Liyang Zhou, Hanqing Jiang, Zhongliang Tang, Yuanbo Wang, Hujun Bao, and Guofeng Zhang. Mobile3drecon: Real-time monocular 3d reconstruction on a mobile phone. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3446–3456, 2020.

[12] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[13] Lingdong Kong, Jiawei Ren, Liang Pan, and Ziwei Liu. Lasermix for semi-supervised lidar semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21705–21715, 2023.

[14] Lingdong Kong, Xiang Xu, Jiawei Ren, Wenwei Zhang, Liang Pan, Kai Chen, Wei Tsang Ooi, and Ziwei Liu. Multimodal data-efficient 3d scene understanding for autonomous driving. *arXiv preprint arXiv:2405.05258*, 2024.

[15] Lingdong Kong, Niamul Quader, and Venice Erin Liong. Conda: Unsupervised domain adaptation for lidar segmentation via regularized domain concatenation. In *IEEE International Conference on Robotics and Automation*, pages 9338–9345, 2023.

[16] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. *European Conference on Computer Vision (ECCV)*, 2018.

[17] 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. In *Conference on Robot Learning*, pages 539–549. PMLR, 2023.

[18] Xianda Guo, Wenjie Yuan, Yunpeng Zhang, Tian Yang, Chenming Zhang, Zheng Zhu, and Long Chen. A simple baseline for supervised surround-view depth estimation. *arXiv preprint arXiv:2303.07759*, 2023.

[19] Lingdong Kong, Shaoyuan Xie, Hanjiang Hu, Lai Xing Ng, Benoit Cottereau, and Wei Tsang Ooi. Robodepth: Robust out-of-distribution depth estimation under corruptions. *Advances in Neural Information Processing Systems*, 36, 2024.

[20] Stefano Gasperini, Patrick Koch, Vinzenz Dallabetta, Nassir Navab, Benjamin Busam, and Federico Tombari. R4dyn: Exploring radar for self-supervised monocular depth estimation of dynamic scenes. In *2021 International Conference on 3D Vision (3DV)*, pages 751–760. IEEE, 2021.

[21] Ukcheol Shin, Jinsun Park, and In So Kweon. Deep depth estimation from thermal image. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 1043–1053, June 2023.

[22] Stefano Gasperini, Nils Morbitzer, HyunJun Jung, Nassir Navab, and Federico Tombari. Robust monocular depth estimation under challenging conditions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8177–8186, 2023.

[23] Yifan Mao, Jian Liu, and Xianming Liu. Stealing stable diffusion prior for robust monocular depth estimation. *arXiv preprint arXiv:2403.05056*, 2024.

[24] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[25] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[26] Lingdong Kong, Shaoyuan Xie, Hanjiang Hu, Benoit Cottereau, Lai Xing Ng, and Wei Tsang Ooi. The robodepth benchmark for robust out-of-distribution depth estimation under corruptions. https://github.com/ldkong1205/RoboDepth, 2023.

[27] Lingdong Kong, Yaru Niu, Shaoyuan Xie, Hanjiang Hu, Lai Xing Ng, Benoit Cottereau, Ding Zhao, Liangjun Zhang, Hesheng Wang, Wei Tsang Ooi, Ruijie Zhu, Ziyang Song, Li Liu, Tianzhu Zhang, Jun Yu, Mohan Jing, Pengwei Li, Xiaohua Qi, Cheng Jin, Yingfeng Chen, Jie Hou, Jie Zhang, Zhen Kan, Qiang Lin, Liang Peng, Minglei Li, Di Xu, Changpeng Yang, Yuanqi Yao, Gang Wu, Jian Kuai, Xianming Liu, Junjun Jiang, Jiamian Huang, Baojun Li, Jiale Chen, Shuang Zhang, Sun Ao, Zhenyu Li, Runze Chen, Haiyong Luo, Fang Zhao, and Jingze Yu. The robodepth challenge: Methods and advancements towards robust depth estimation. *arXiv preprint arXiv:2307.15061*, 2023.

[28] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Benchmarking and improving bird's eye view perception robustness in autonomous driving. *arXiv preprint arXiv:2405.17426*, 2024.

[29] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robobev: Towards robust bird's eye view perception under corruptions. *arXiv preprint arXiv:2304.06719*, 2023.

[30] Shaoyuan Xie, Lingdong Kong, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. The robobev benchmark for robust bird's eye view detection under common corruption and domain shift. https://github.com/Daniel-xsy/RoboBEV, 2023.

[31] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. Robo3d: Towards robust and reliable 3d perception against corruptions. In *IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023.

[32] Lingdong Kong, Youquan Liu, Xin Li, Runnan Chen, Wenwei Zhang, Jiawei Ren, Liang Pan, Kai Chen, and Ziwei Liu. The robo3d benchmark for robust and reliable 3d perception. https://github.com/ldkong1205/Robo3D, 2023.

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