Robust Monocular Depth Estimation under Challenging Conditions

Stefano Gasperini*,[1,2] Nils Morbitzer*,[1] HyunJun Jung[1]
Nassir Navab[1] Federico Tombari[1,3]

[1] Technical University of Munich [2] VisualAIs [3] Google

Abstract

While state-of-the-art monocular depth estimation approaches achieve impressive results in ideal settings, they are highly unreliable under challenging illumination and weather conditions, such as at nighttime or in the presence of rain. In this paper, we uncover these safety-critical issues and tackle them with md4all: a simple and effective solution that works reliably under both adverse and ideal conditions, as well as for different types of learning supervision. We achieve this by exploiting the efficacy of existing methods under perfect settings. Therefore, we provide valid training signals independently of what is in the input. First, we generate a set of complex samples corresponding to the normal training ones. Then, we train the model by guiding its self- or full-supervision by feeding the generated samples and computing the standard losses on the corresponding original images. Doing so enables a single model to recover information across diverse conditions without modifications at inference time. Extensive experiments on two challenging public datasets, namely nuScenes and Oxford RobotCar, demonstrate the effectiveness of our techniques, outperforming prior works by a large margin in both standard and challenging conditions. Source code and data are available at: https://md4all.github.io.

1. Introduction

Estimating the depth of a scene is a fundamental task for autonomous driving and robotics navigation. While supervised monocular depth estimation approaches have achieved remarkable results, they rely on ground truth data which is expensive and time-consuming to produce [19, 12]. This requires costly 3D sensors (e.g., LiDAR) and significant additional data processing [19, 12].

To circumvent these issues, geometrical constraints on stereo pairs or monocular videos have been widely explored to learn depth estimation in a self-supervised manner [11, 25, 34, 7, 10]. Monocular training solutions are the most inexpensive and rely on the smallest amount of assumptions on the sensor setup, as they require only image sequences captured by a single camera.

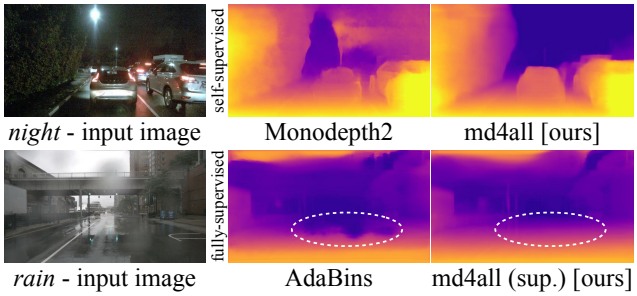

Figure 1. Predictions in challenging settings [3] for self-supervised [11] and supervised [1] methods. Standard approaches fail due to training assumptions or sensor artifacts. Under both supervisions, our md4all makes the same models robust in all conditions.

Self-supervised methods rely on photometric assumptions and pixel correspondences [11, 34]. State-of-the-art approaches [11, 41, 32] deliver sharp and accurate estimates in standard conditions (i.e., sunny and cloudy), but suffer from a variety of inherent issues, such as scale ambiguity and difficulties with dynamic objects. While prior works have already proposed robust methods to address these problems [12, 8], there is still a major issue preventing the wide applicability of self-supervised depth estimators in safety-critical settings, such as autonomous driving. Darkness and adverse weather conditions (e.g., night, rain, snow, and fog) introduce noise in the pixel correspondences. As displayed in Figure 1, this is detrimental to the effectiveness of such methods, thereby requiring ad hoc solutions.

As shown in Figure 2, this problem is particularly severe at nighttime due to reflections (e.g., caused by streetlights and vehicle headlights), noise, and the general inability of the embedded cameras to capture details in dark areas. This leads to wrong depth estimates, which can be dangerous in safety-critical settings. A few pioneering works have already explored this problem, albeit with highly-

* The authors contributed equally.
Contact author: Stefano Gasperini (*stefano.gasperini@tum.de*).

complex pipelines and significant architecture changes affecting inference as well [38, 37, 22, 33, 36], such as illumination-specific branches. Additionally, prior methods that can operate both at night- and daytime introduce a significant trade-off concerning the standard daytime performance [22, 37], highlighting the need for a new solution.

In adverse weather conditions such as rain, monocular models are similarly fooled by reflections and decreased visibility. However, rain introduces another problem. While radars are robust in such conditions, LiDARs become unreliable, as they introduce multi-path and the so-called blooming effects (Figure 2). In autonomous driving, since supervised depth estimation approaches learn from LiDAR data, this causes them to learn also such erroneous measurements, rendering them unreliable in rainy settings (Figure 1). Analogous issues occur with snow and fog. These problems are relatively unexplored, demanding new solutions.

Alarmingly, no general solution currently allows an image-based depth estimator to work reliably under all conditions. Since LiDAR can constitute a misleading training signal in adverse weather, and pixel correspondences are problematic too (e.g., at night), neither existing supervised [24] nor self-supervised [11, 34] techniques work well in such challenging settings. A straightforward solution for the supervised case would be using synthetic data [40, 30], as by simply not modeling the sensor issues, a simulator could produce perfect ground truth in adverse weather. However, this is not only unexplored, but it would introduce a series of problems, such as a substantial syn2real gap due to the difficulty of modeling challenging conditions realistically (requiring, e.g., domain adaptation).

In this paper, we address these open issues with a simple and effective solution that works reliably in a variety of conditions and for multiple types of supervision. We approach this challenging problem by considering the success of existing methods in standard illumination and weather settings [11, 13, 12, 8]. This motivated us to find a way for them to work also under challenging scenarios, exploiting what makes them learn depth effectively in ideal conditions. Our core idea is based on training the model by providing always valid training signals as if it was sunny or cloudy, even when samples with adverse conditions are given. We apply this general principle to both supervised and self-supervised depth estimation via a set of techniques to improve the model robustness and reduce the performance gap between standard and hard conditions. The main contributions of this paper can be summarized as follows:

- We show how estimating depth in adverse conditions (e.g., night and rain) is problematic for both self- and fully-supervised approaches, requiring new solutions.
- We propose md4all: a simple and effective technique to make standard models robust in diverse conditions.

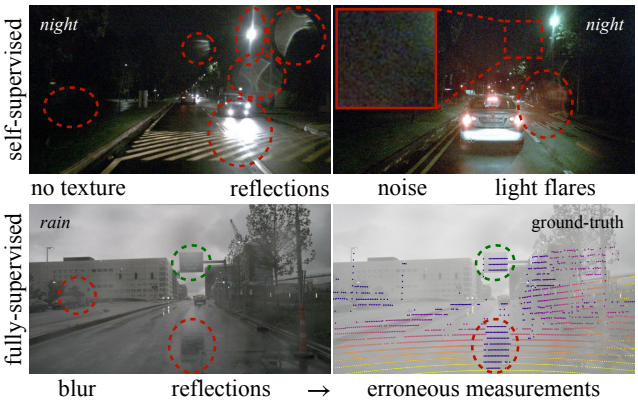

Figure 2. Detrimental factors to monocular depth estimation in difficult settings from nuScenes [3]. Self-supervised works have issues with textureless areas, reflections, and noise. Supervised ones learn artifacts from the ground truth sensor (LiDAR is shown).

- We apply our generic method to both fully- and self-supervised monocular settings.
- We generate and share open-source images in adverse conditions corresponding to the sunny and cloudy samples of nuScenes [3] and Oxford Robotcar [23].

With md4all, we substantially outperform prior solutions delivering robust estimates in a variety of conditions.

2. Related Work

2.1. Supervised Monocular Depth Estimation

The problem of estimating depth from a single color image is challenging due to the countless 3D scenarios that can produce the same 2D projection, making it an ill-posed problem. Nevertheless, significant progress has been made, thanks to the introduction of CNN-based architectures by Eigen et al. [5] and fully-convolutional networks with residual connections by Laina et al. [20] to estimate dense depth maps from monocular inputs. While many supervised methods have focused on directly regressing to depth measurements from LiDAR sensors (as in KITTI [9]) or RGB-D cameras (as in NYU-Depth v2 [31]), DORN [6] tackles the task in an ordinal manner. AdaBins [1] extended DORN via a linear combination of predictions across adaptive bins. Moreover, BTS uses a multi-stage local planar guidance [21] and P3Depth exploits coplanar pixels [24]. Others investigated the benefit of depth estimation while tackling other tasks, such as 3D object detection [17].

Issues While the supervision signal from 3D sensors is reliable in ideal conditions (e.g., sunny, cloudy), it severely degrades in photometrically challenging scenarios [19]. Outdoor, LiDAR sensors deliver erroneous measurements in adverse weather conditions, such as rain, snow and fog. As Jung et al. demonstrated indoor [19], training on an inexact ground truth leads depth models to learn the sensor ar-

tifacts and deliver wrong outputs. This problem is relatively unexplored outdoors, e.g., with rain. A few works investigated depth completion in simulated settings with LiDAR and radar in input [40] or event cameras and RGB [30]. In this paper, we explore this issue on AdaBins [1] and provide a simple solution to estimate depth reliably in diverse conditions, regardless of the sensor artifacts.

2.2. Self-Supervised Monocular Depth Estimation

To bypass the need for expensive LiDAR data, self-supervised methods employ view reconstruction constraints through stereo pairs [7, 10] or monocular videos [11, 46]. The latter utilizes motion parallax from a moving camera in a static environment [35] and requires simultaneous depth and camera pose transformation prediction. Significant advancements have been made since Zhou et al.'s pioneering video-based approach [46], including novel loss terms [11], network architectures that preserve details [12], the use of cross-task dependencies [18, 13], pseudo labels [25], vision transformers [44], uncertainty estimation [26], and 360 degrees depth predictions [15].

2.2.1 Solutions to Inherent Issues

Scale ambiguity Video-based methods predict depth up to scale, requiring median-scaling with ground truth data at test time [11]. Guizilini et al. [12] used the readily available odometry information to achieve scale awareness via weak velocity supervision on the pose transformation.

Dynamic scenes Due to the moving camera in a static world assumption [35], video-based methods have issues with dynamic objects, e.g., cars. To address this, Monodepth2 [11] uses an auto-masking loss on the static pixels, R4Dyn [8] adds weak radar supervision on the objects, and DRAFT [14] combines optical and scene flows.

Darkness Low visibility is detrimental to the losses used to learn depth because noise and lack of details prevent establishing pixel correspondences across the frames. DeFeat-Net [33] was among the first to mitigate this, with a cross-domain dense feature representation. ADFA [36] uses a generative adversarial network (GAN) to adapt nighttime features to daytime ones. R4Dyn [8] shows that radar is beneficial not only for dynamic objects but also at nighttime as a byproduct. RNW [38] reduces the irregularities at nighttime via, e.g., image enhancement and a GAN-based regularizer. ADIDS [22] uses separate networks for day and night images, partially sharing weights. ITDFA [43] is similar to ADFA, doing feature adaptation from night to day, with images generated with a GAN. WSGD [37] combines denoising with a lighting change decoder to predict per-pixel changes. While these works made significant steps towards solving the problem, they either have complex pipelines with dedicated branches for day and night [22, 43], use additional sensors [8], suffer from a significant trade-off on the daytime performance [37], or are not meant to operate on multiple conditions, such as both day and night [36, 38, 43]. Therefore, an effective solution without inference complications is yet to be found.

Adverse weather As at nighttime, in adverse weather such as rain, fog, and snow, the limited visibility prevents establishing correct correspondences. Even fully-supervised approaches have issues in these settings [19]. So far, only a handful of works have explored depth estimation with adverse weather. ITDFA [43] requires an encoder for each condition and was not shown to work in both standard and adverse settings. R4Dyn [8] and MonoViT [44] are robust methods that delivered improvements also in adverse conditions as a side effect. Thus, this problem is largely unexplored, demanding a general solution.

Unlike prior works, in this paper, we propose a simple and effective solution enabling a standard monocular model to estimate depth in diverse conditions (e.g., day, night, and rain) without any difference at inference time compared to a common encoder-decoder pipeline [11]. Additionally, ours does not degrade the output quality in standard settings.

3. Method

In this paper, we enable a model to estimate depth reliably in diverse conditions (e.g., day, night, and rain). Displayed in Figures 3 and 4, our techniques exploit the effectiveness of existing approaches in standard conditions (e.g., daytime in good weather) to increase their robustness in adverse settings. Towards this end, we perform day-to-adverse image translation, train on the generated adverse samples, and learn only from valid training signals from the original day inputs. This simple idea is suitable to both self-supervised (Section 3.1) and supervised (Section 3.2) frameworks and is general to operate under various weather and illumination settings (including fog and snow).

3.1. md4all - Self-Supervised

We build upon a scale-aware video-based monocular method (Section 3.1.1). As described in Section 2.2.1, night and bad weather cause issues to self-supervised approaches. We address this with md4allby computing the losses only on the ideal samples corresponding to the hard ones given as input (Section 3.1.2). We then take this concept even further by distilling knowledge from a frozen self-supervised model trained only on the ideal samples (Section 3.1.3).

3.1.1 Self-Supervised Baseline

We build on a standard video-based monocular depth baseline equivalent to the framework shown in Figure 4 when considering $x = 0$ (i.e., no translation). We predict both the depth $\hat{D}_t$ of a target frame and the pose transformations

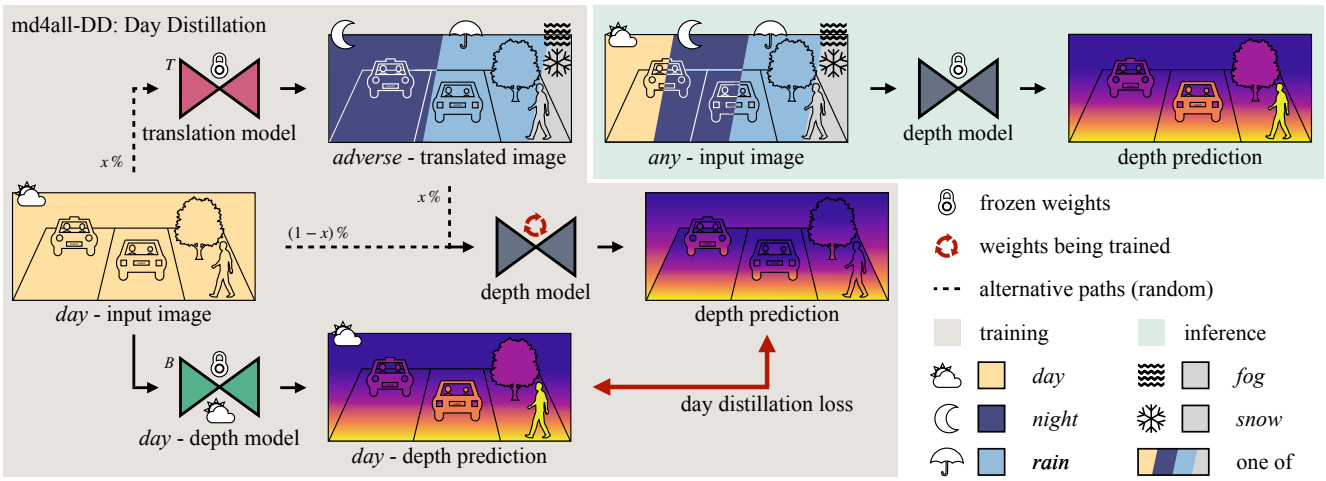

Figure 3. Our md4all-DD framework. The frozen *day* - depth model estimates on easy samples and provides guidance to another model fed with a mix of easy and translated inputs. Inference is done with a simple single model for both fully- and self-supervised md4all.

between the target I_t and source frames $I_{s\in\{t-1,t+1\}}$, with which we warp the source into a reconstructed target view. As in [11, 12], a loss is computed on the appearance shift between I_t and the reconstruction [46], alongside the structural similarity [39]. Following [11], we account for partial occlusions via the minimum reprojection error $\mathcal{L}_p$, and we ignore static pixels. Another loss $\mathcal{L}_s$ promotes smoothness and preserves edges [10]. $\mathcal{L}_p$ and $\mathcal{L}_s$ are calculated at all decoder scales, upsampled to the input size [11].

So far, this is equivalent to Monodepth2 [11]. Then, we add the weak velocity supervision $\mathcal{L}_v$ to achieve scale-awareness [12] and allow consistent predictions, beneficial when distilling knowledge between different models.

Architecture Unlike previous works having specialized branches [22, 43], we leave the architecture unchanged (e.g., [11]). Instead, we act on the training process. Our approach is general and not bound to a specific architecture.

3.1.2 md4all-AD: Always Daytime, No Bad Weather

Our md4all-AD configuration is shown in Figure 4. The core idea is learning from easy samples, even when given challenging ones (e.g., night) as if it was always daytime with good visibility (i.e., sunny or cloudy). This allows using the same established losses described in Section 3.1.1, which would otherwise fail with difficult inputs.

Day-to-adverse translation To achieve the above, we need easy samples corresponding to the challenging ones. This means having paired images (e_i, h_i^c), with $e_i \in E$ and E being the set of easy samples (i.e., sunny or cloudy), $h_i^c \in H$ with H the set of the difficult samples from the conditions of interest $c \in C$ (e.g., snow). While an image translation method could convert the training H into easy ones, removing information is easier than adding it. Therefore, we generate H from E (e.g., turning e_i into nighttime). Specifically, for each e_i and each condition c we aim to improve (e.g., night and rain), we obtain $h_i^c = T^c(e_i)$. We do this with c image translation models T^c trained at an earlier stage, increasing the training set size by $C \times E$.

Training scheme We then train depth and pose models as shown in Figure 4. During training, we feed to the depth model m_i, which is either h_i^c (for $x\%$ of the inputs, as a random mix of c) or e_i from the pre-existing training data. Additionally, we normalize the inputs depending on the recording time (i.e., day/night) to learn robust features agnostic of the input condition. The Appendix shows how performing this step only during training delivers similar results. Then, in the case of particularly noisy night samples (e.g., nuScenes [3]), we augment the inputs with heavy noise. The pose model always takes the sequence $[e_{i-1}, e_i, e_{i+1}]$, corresponding to m_i. If fed h_i^c, the pose network would have issues assessing the pixel correspondences.

Learning in all conditions Computing the losses $\mathcal{L}_p$ and $\mathcal{L}_s$ on h_i^c would lead to issues because of the difficulty of establishing correspondences in adverse conditions (Section 2.2.1). For this reason, training on E and deploying on H is more effective than training on both (Section 4.2), proving the limitations of standard methods. Our solution to this challenging problem is relatively simple: as shown in the figure, we provide a reliable training signal by always calculating the losses on E. Specifically, they are always computed on e_i, even when the depth model is fed with h_i^c ($x\%$). This constructed setting constitutes the ideal condition in which the losses $\mathcal{L}_p$ and $\mathcal{L}_s$ are already proven successful [11], eliminating the source of the issues. This leads the depth model to learn to extract robust features, regardless of whether the input belongs to E or H.

Inference After training depth and pose models, the latter is discarded, while our depth model is a simple encoder-decoder capable of estimating depth in multiple conditions.

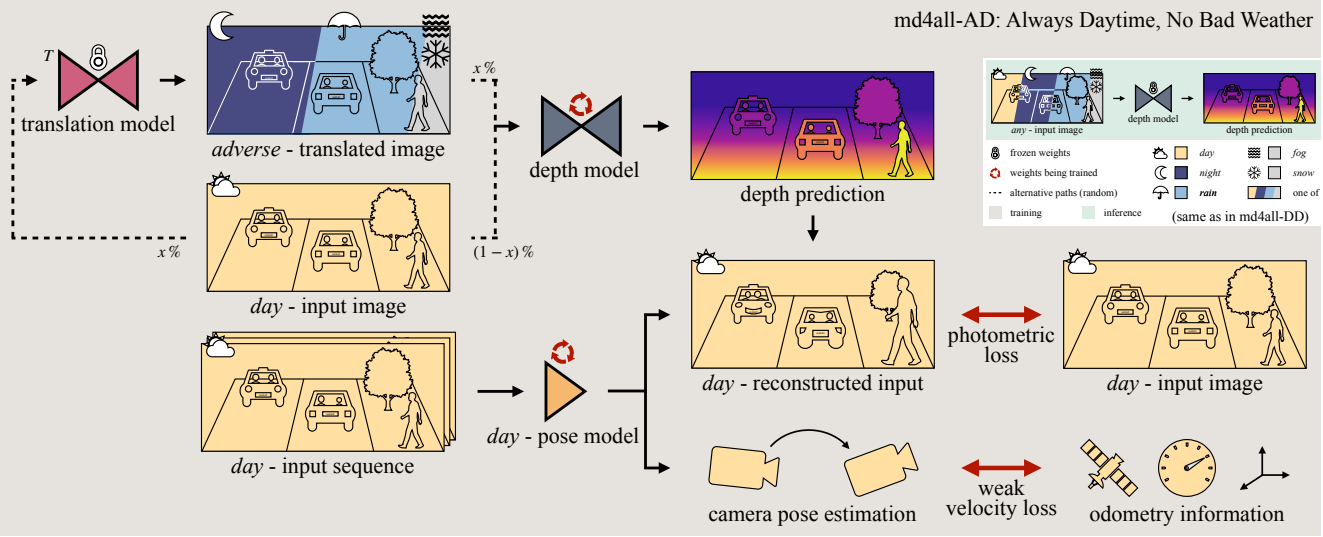

Figure 4. Our self-supervised md4all-AD framework. With $x = 0$, it is equivalent to the *day* - depth model in Figure 3 and the baseline. The depth model is trained with a mix of easy and translated samples, while the training signal is always from the easy ones.

As shown at the top of Figure 3, since we do not apply any architectural modification, at inference time, we predict depth with the same model parameters, regardless of the input condition. While dedicated models or branches may lead to better performance, switching between them is not always trivial, e.g., at dusk or with light rain. Therefore, we opted for a single monocular model, which does not penalize inference time compared to the same model trained only on E.

3.1.3 md4all-DD: Day Distillation

We take md4all-AD (Section 3.1.2) to the next level by simplifying the training scheme with md4all-DD. The core idea of md4all-DD is the same as for md4all-AD: we aim to learn depth only from E, pretending that the conditions C detrimental for the losses never occur.

Our md4all-DD framework mimics model estimates in ideal settings E, regardless of the difficulty of the input. As shown in Figure 3, we achieve this via knowledge distillation from a depth network B (baseline) trained at an earlier stage on E to a new depth model DD for both easy and adverse scenarios (i.e., E and H). The latter is fed m_i, i.e., the same mix of e_i and h_i^c as in md4all-AD (Section 3.1.2), while the former is given only e_i. DD is optimized solely through the following objective:

$$\mathcal{L}_d = \frac{1}{N} \sum_{j=1}^{N} \frac{|DD(m_i)_j - B(e_i)_j|}{DD(m_i)_j} \qquad (1)$$

where N is the number of pixels, $DD(m_i)$ is DD's depth prediction on m_i (i.e., an easy or hard sample), and $B(e_i)$ is B's estimation on e_i (i.e., an easy sample). DD learns to follow B at the output level, even when fed the problematic h_i^c, without being affected by the detrimental factors occurring in adverse settings. Inference is unchanged.

3.2. md4all - Supervised

Learning depth from a 3D sensor in adverse conditions exposes issues inherent to the sensor and the way it measures depth [19]. With bad weather (e.g., rain), LiDARs provide erroneous measurements (Figure 2), so learning from their signal means copying their artifacts as well (Figure 1). This has been ignored so far for monocular depth.

Regardless of the input, we address the sensor issues by learning from E. Analogously to the self-supervised setting, we use image pairs (e_i, h_i^c) and specify our method as md4all-AD following the self-supervised definition (Section 3.1.2), except for the supervision signal. Thus, we train the depth model with m_i and learn from the LiDAR signal of e_i. Thus, the supervision is from artifact-free data in ideal conditions E, such that the models never experiences the sensor issues. As in the self-supervised setup, the inference is unchanged. While md4all-DD (Section 3.1.3) also applies to the supervised case, using AD is more reasonable since reliable ground truth data from e_i is available.

4. Experiments and Results

4.1. Experimental Setup

Datasets and metrics We used two public driving datasets containing various illumination and weather conditions: nuScenes [3] and Oxford RobotCar [23]. **nuScenes** is a challenging large-scale dataset with 15h of driving in Boston and Singapore, diverse scenes, and difficult conditions. We distinguished good visibility (i.e., *day-clear*),

Method	sup.	tr.data	*day-clear* – nuScenes absRel	RMSE	δ_1	*night* – nuScenes absRel	RMSE	δ_1	*day-rain* – nuScenes absRel	RMSE	δ_1
Monodepth2 [11]	M*	*a*: *dnr*	0.1477	6.771	85.25	2.3332	32.940	10.54	0.4114	9.442	60.58
Monodepth2 [11]	M*	*d*	0.1374	6.692	85.00	0.2828	9.729	51.83	0.1727	7.743	77.57
PackNet-SfM [12]	Mv	*d*	0.1567	7.230	82.64	0.2617	11.063	56.64	0.1645	8.288	77.07
R4Dyn w/o r in [8]	Mv**r**	*d*	0.1296	6.536	85.76	0.2731	12.430	52.85	0.1465	7.533	80.59
R4Dyn [8] (radar)	Mv**r**	*d*	**0.1259**	**6.434**	**86.97**	0.2194	10.542	62.28	**0.1337**	**7.131**	**83.91**
RNW [38]	M*	*dn*	0.2872	9.185	56.21	0.3333	10.098	43.72	0.2952	9.341	57.21
[ours] baseline	Mv	*d*	0.1333	6.459	85.88	0.2419	10.922	58.17	0.1572	7.453	79.49
[ours] md4all-AD	Mv	*dT(nr)*	0.1523	6.853	83.11	0.2187	9.003	68.84	0.1601	7.832	78.97
[ours] **md4all-DD**	Mv	*dT(nr)*	0.1366	6.452	84.61	**0.1921**	**8.507**	**71.07**	0.1414	7.228	80.98
AdaBins [1]	GT	*a*: *dnr*	0.1384	5.582	81.31	0.2296	7.344	63.95	0.1726	6.267	76.01
[ours] **md4all-AD**	GT	*dnT(r)*	**0.1206**	**4.806**	**88.03**	**0.1821**	**6.372**	**75.33**	**0.1562**	**5.903**	**82.82**

Table 1. Evaluation of self- and GT-supervised methods on the nuScenes [3] validation set. Supervisions (sup.): M: via monocular videos, *: test-time median-scaling via LiDAR, v: weak velocity, r: weak radar, GT: via LiDAR data. Training data (tr.data): *d*: *day-clear*, *T*: translated in, *n*: *night* (incl. *night-rain*), *r*: *day-rain*, *a*: *all*. Visual support: **1st**, 2nd, 3rd best. More conditions and metrics in the Appendix.

night (including *night-rain*), and *day-rain*. We used the official split following R4Dyn [8], with 15129 training images (with synced sensors), and 6019 validation ones (of which 4449 *day-clear*, 602 *night*, and 1088 *rain*). **RobotCar** was collected in Oxford, UK, by traversing the same route multiple times in a year. It features a mix of *day* and *night* scenes. We followed the split and setup of WSGD [37], with 16563 *day* training samples and 1411 test ones (with synced sensors, of which 709 *night*). While we focused on night, rain, sun, and overcast, the Appendix shows preliminary results with fog and snow from the **DENSE** dataset [2]. We report on the standard metrics and errors up to 50m for RobotCar as in [37], and 80m for nuScenes as in [8]. More results can be found in the Appendix.

Implementation details Our self-supervised models use a ResNet-18 backbone [16] and learn from an image triplet sized 576x320 for nuScenes and 544x320 for RobotCar. The supervised model and md4all-DD are given only one keyframe. At inference time, all models take a single RGB input. We set $x = |C|/(|C|+1)$ %, with $|C|$ being the number of the adverse conditions of interest C, e.g., $x = 66\%$ for a model to work with *rain*, *night* and *day*, and within $x\%$ we used equally distributed data among C. So, our models see an equal amount of inputs for each condition. We used the same hyperparameters as Monodepth2 [11] and AdaBins [1] for self- and fully-supervised models, respectively. All models were trained on a single 24GB GPU.

Image translation We translated each e_i image to h_i^c. Diffusion models [27, 28] are not suitable due to the lack of already paired images. Datasets with multiple drives on the same roads [4, 29, 23] do not solve this issue due to the lack of synchronization and environmental changes. So we opted for GANs. For each condition c, we used a ForkGAN model [45] T^c to translate all *day-clear* training samples E of nuScenes, with $c \in C = \{night, rain\}$. We trained ForkGAN on BDD100K [42] and fine-tuned it on the nuScenes training set. For RobotCar, we used T^c to translate all *day* samples E into *night* ones. RobotCar contains more *night* samples than nuScenes, so we trained T^c directly on its training set. We share publicly all generated h_i^c images.

Prior works and baselines We compared ours with a variety of works [8, 11, 12, 38, 37, 22, 33, 1]. We applied ours on the self-supervised Monodepth2-based baseline of Section 3.1.1 and the fully-supervised AdaBins [1].

4.2. Quantitative Results

***Night* – nuScenes** In Table 1, we report results for nuScenes [3] across various settings. Night samples present strong noise levels and reflections that are detrimental for self-supervised models (Figure 2), causing the absRel errors of most methods to double from ideal conditions (i.e., *day-clear*) to *night*. The difficulty of learning from night inputs is evident comparing Monodepth2 [11] trained only on *day-clear* (*d*) against *all* conditions (*a*), with the latter severely underperforming. PackNet [12] improved at *night* and *rain*, albeit doing worse in standard settings, possibly due to its large model and the relatively small dataset. PackNet's velocity supervision also helped over Monodepth2 (md2) with our baseline. Thanks to the extra radar signal, R4Dyn [8] delivered significant improvements, although at *night*, only adding radar in input was beneficial over md2. md2 trained only on *day-clear* data outperformed RNW's complex pipeline [38]. We retrained RNW on the official split (the authors reported an absRel of 0.3150 at *night* on their split [38]). Remarkably, despite being based on the same model as md2, at *night*, our simple techniques reduced absRel by 32% and relatively increased δ_1 by 37% (DD). Our md4all also outperformed the radar-based R4Dyn at *night*. This is thanks to the ability of our method to extract robust features from monocular data even in the dark.

Method	source	sup.	tr.data	*day* – RobotCar absRel	sqRel	RMSE	δ_1	*night* – RobotCar absRel	sqRel	RMSE	δ_1
Monodepth2 [11]	[ours]	M*	*d*	0.1196	0.670	**3.164**	86.38	0.3029	1.724	5.038	45.88
DeFeatNet [33]	[37]	M*	*a*: *dn*	0.2470	2.980	7.884	65.00	0.3340	4.589	8.606	58.60
ADIDS [22]	[37]	M*	*a*: *dn*	0.2390	2.089	6.743	61.40	0.2870	2.569	7.985	49.00
RNW [38]	[37]	M*	*a*: *dn*	0.2970	2.608	7.996	43.10	0.1850	1.710	6.549	73.30
WSGD [37]	[37]	M*	*a*: *dn*	0.1760	1.603	6.036	75.00	0.1740	1.637	6.302	75.40
[ours] baseline	[ours]	Mv	*d*	0.1209	0.723	3.335	86.61	0.3909	3.547	8.227	22.51
[ours] **md4all-DD**	[ours]	Mv	*dT(n)*	**0.1128**	**0.648**	3.206	**87.13**	**0.1219**	**0.784**	**3.604**	**84.86**

Table 2. Evaluation of self-supervised works on the RobotCar [23] test set. Trailing 0 added to the values from [37]. Notation from Table 1.

***Night* – RobotCar** In Table 2, we report results for RobotCar [23]. Here we compare with various approaches that also target depth estimation in challenging conditions [33, 22, 38, 37]. They all focus on *night* issues, tested here. Our md4all outperforms them all across the board, with substantially better estimates at *night* than theirs during the *day*: the previous best WSGD [37]'s *day* absRel error is 45% higher than ours at *night*. This is thanks to the simplicity of our approach, which does not rely on complex architectures, but makes existing models robust in adverse conditions by changing their input and training signals.

***Rain* – nuScenes** Rain is less problematic than darkness due to the lack of cues in the latter. Results are shown in Table 1, with all methods performing better with *rain* than at *night*. Our self-supervised monocular md4all-DD significantly improved over Monodepth2 and the baseline, performing close to the radar-based R4Dyn [8].

Fully-supervised Table 1 reports also results in supervised settings. LiDAR data is reliable in the dark, so *night* scenes are less of an issue. Instead, *rain* inputs are particularly interesting for supervised works due to the reflection issues shown in Figures 1 and 2. For supervised settings, we applied our method on AdaBins [1]. It is to be considered that LiDAR artifacts may have an impact on the *rain* values, such that perfect estimates would not score perfectly because the ground truth is wrong (Figure 2). So, while we can assess the improvements of md4all at handling the blur caused by raindrops, we cannot correctly quantify its impact on eliminating the artifacts. Therefore, these comparisons are more meaningful when considered alongside qualitative outputs (Figure 5). Our supervised md4all performed better than AdaBins both quantitatively and qualitatively, eliminating the dependency on the sensor artifacts. Additionally, thanks to the strong regularization introduced by the translated samples, our model generalizes significantly better than the standard AdaBins, leading to vast improvements across the board, also at *night*. Training on the sparse LiDAR signal of nuScenes [3] (Figures 2 and 5) can lead to overfitting. Ours is a beneficial data augmentation technique, adding diversity to the training, as the model is shown $|C| + 1$ variations of each *day-clear* input.

Day(-clear) While we do not include any modification addressing standard conditions, we still see improvements over the baselines across both datasets and supervision types (Tables 1 and 2). This is due to the training mix of easy and translated samples acting as a strong data augmentation and regularization technique. Since the same weights are optimized on all conditions, they learn to extract robust features which are beneficial also with good visibility. Instead, RNW [38] is meant for operating only at night.

All conditions Remarkably, across all tested conditions, md4all significantly improves over Monodepth2 and AdaBins on which we applied it, without the need for specialized branches (Tables 1 and 2). There is no trade-off introduced when training our unique md4all model for multiple conditions, as the scores and errors remain equivalent or even improve compared to training only in ideal settings. This proves the effectiveness and generality of our simple ideas. The Appendix includes preliminary results with *snow* and *fog* on the challenging DENSE dataset [2].

AD and DD Our md4all delivers improvements both as DD and AD (Tables 1 and 2). While the two are applicable under both supervisions, available and reliable ground truth alongside the *day-clear* data makes AD more suitable for supervised setups. DD works better than AD in self-supervised settings thanks to the simplified training scheme and the guidance of our strong baseline.

Robustness against translations In Table 3, we assess the impact of the quality of image translation on our method. While the selected ForkGAN [45] translates better than CycleGAN [47], it does not give perfect outputs either (Appendix). Since we use the translations to learn robust features, their imperfections even help our model's ro-

Method	*avg/all*	*day*	*night*
[ours] w/ CycleGAN [47]	0.1244	0.1159	0.1328
[ours] w/ ForkGAN [45]	**0.1174**	**0.1128**	**0.1219**
[ours] w/ degraded ForkGAN	0.1213	0.1159	0.1266

Table 3. Robustness of md4all-DD against translations from different GANs. Evaluation of absRel on the RobotCar [23] test set.

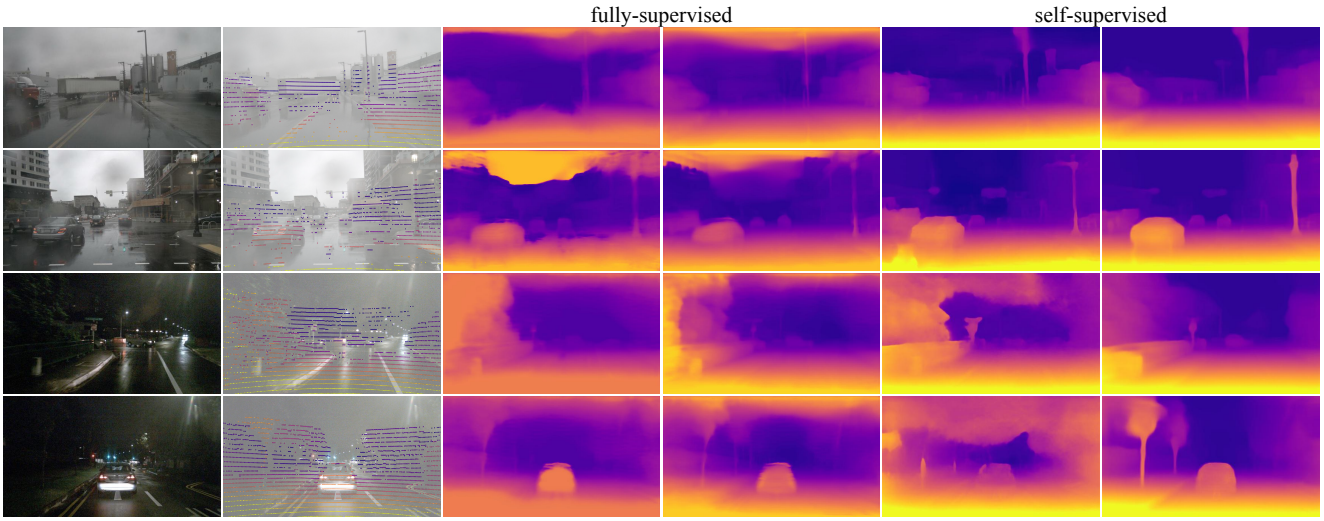

Figure 5. Comparison on nuScenes [3] between fully-sup. AdaBins [1] w/o and w/ ours, and self-sup. Monodepth2 [11] w/o and w/ ours.

bustness by making it harder to recover information for the depth task, as the translations act as data augmentation and regularization. The table confirms the robustness of md4all, performing similarly regardless of which GAN is used, even when degrading 10% of the inputs via random erasing.

4.3. Qualitative Results

Qualitative comparisons in Figures 5 and 6 confirm the quantitative findings, with our md4all delivering improved estimates in both adverse and standard conditions. On nuScenes [3] (Figure 5), unlike the baselines, both our models correctly identified the truck in the first rainy sample. As shown in Figure 2, rain leads to artifacts in the LiDAR ground truth, which cause the standard fully-supervised AdaBins [1] to learn them and estimate the road wrongly. Our supervised md4all exhibits no such artifacts as it was not trained with the problematic rainy samples but rather on our translated ones, which have reliable ground truth. Instead, self-supervised methods have issues at *night*. While Monodepth2 [11] could identify critical elements of the scenes (e.g., car and sign), its difficulties in extracting information in the dark are evident. Monodepth2 had fewer issues with brighter *night* samples, as shown in the Appendix. Our self-supervised md4all delivered sharp estimates, identifying even the two trees on the left side of the bottom input, which are particularly hard to see. For RobotCar [23] (Figure 6), we compared on the same samples displayed by WSGD in their paper [37]. As in Table 2, our md4all delivered better and sharper estimates in both conditions, correctly estimating the people's distance.

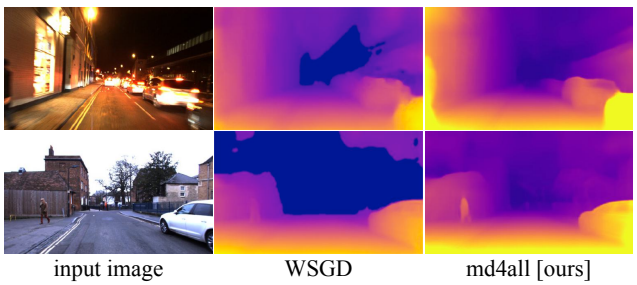

Figure 6. Comparison on RobotCar [23] samples between ours self-supervised and WSGD [37]. Ouputs of WSGD are from [37].

Limitations md4all improves in all tested conditions, but DD may propagate errors from the baseline. Thus, a stronger baseline would help. Despite the robustness against translations (Table 3), GANs [45] could be problematic. Better translations would help eliminate the domain gap, as seen with RobotCar (Table 2). GANs require many adverse images for training. Hard-to-distinguish data distributions (e.g., light snow vs. overcast) may create problems. md4all is applicable to stereo-based models too, but only given consistent translations for the stereo images. Future work may focus on eliminating the dependency on the GAN. Furthermore, md4all does not address the issue of dynamic objects, so flow [14] or weak radar supervision [8] may be beneficial, albeit adding complexity. The core ideas of this work can be extended to other tasks.

The **Appendix** includes a variety of extra results, e.g., experiments with *snow* and *fog*, and sample translations.

5. Conclusion

We presented the simple and effective md4all, enabling a single monocular model to estimate depth robustly in both standard and challenging conditions (e.g., night, rain). We showed md4all delivering significant improvements under both fully- or self-supervised settings, overcoming the detrimental factors that make adverse conditions problematic.

References

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. AdaBins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 1, 2, 3, 6, 7, 8

[2] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020. 6, 7

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 1, 2, 4, 5, 6, 7, 8

[4] Carlos A Diaz-Ruiz, Youya Xia, Yurong You, Jose Nino, Junan Chen, Josephine Monica, Xiangyu Chen, Katie Luo, Yan Wang, Marc Emond, et al. Ithaca365: Dataset and driving perception under repeated and challenging weather conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21383–21392, 2022. 6

[5] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, 27, 2014. 2

[6] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 2

[7] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *Proceedings of the European Conference on Computer Vision*, pages 740–756. Springer, 2016. 1, 3

[8] 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 *Proceedings of the International Conference on 3D Vision (3DV)*, pages 751–760. IEEE, 2021. 1, 2, 3, 6, 7, 8

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2

[10] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 1, 3, 4

[11] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 1, 2, 3, 4, 6, 7, 8

[12] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3D packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494, 2020. 1, 2, 3, 4, 6

[13] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *International Conference on Learning Representations*, 2020. 2, 3

[14] Vitor Guizilini, Kuan-Hui Lee, Rareş Ambruş, and Adrien Gaidon. Learning optical flow, depth, and scene flow without real-world labels. *IEEE Robotics and Automation Letters*, 7(2):3491–3498, 2022. 3, 8

[15] Vitor Guizilini, Igor Vasiljevic, Rares Ambrus, Greg Shakhnarovich, and Adrien Gaidon. Full surround monodepth from multiple cameras. *IEEE Robotics and Automation Letters*, 7(2):5397–5404, 2022. 3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6

[17] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H Hsu. MonoDTR: Monocular 3d object detection with depth-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4012–4021, 2022. 2

[18] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European Conference on Computer Vision*, pages 53–69, 2018. 3

[19] HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou, Anil Armagan, Slobodan Ilic, Aleš Leonardis, Nassir Navab, and Benjamin Busam. On the importance of accurate geometry data for dense 3D vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 780–791, 2023. 1, 2, 3, 5

[20] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 239–248. IEEE, 2016. 2

[21] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 2

[22] Lina Liu, Xibin Song, Mengmeng Wang, Yong Liu, and Liangjun Zhang. Self-supervised monocular depth estimation for all day images using domain separation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12737–12746, 2021. 2, 3, 4, 6, 7

[23] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. 2, 5, 6, 7, 8

[24] Vaishakh Patil, Christos Sakaridis, Alexander Liniger, and Luc Van Gool. P3Depth: Monocular depth estimation with a piecewise planarity prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1610–1621, 2022. 2

[25] Andra Petrovai and Sergiu Nedevschi. Exploiting pseudo labels in a self-supervised learning framework for improved monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1578–1588, 2022. 1, 3

[26] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3227–3237, 2020. 3

[27] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 6

[28] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *Proceedings of the ACM SIGGRAPH Conference*, pages 1–10, 2022. 6

[29] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10765–10775, 2021. 6

[30] Peilun Shi, Jiachuan Peng, Jianing Qiu, Xinwei Ju, Frank Po Wen Lo, and Benny Lo. EVEN: An event-based framework for monocular depth estimation at adverse night conditions. *arXiv preprint arXiv:2302.03860*, 2023. 2, 3

[31] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. *Proceedings of the European Conference on Computer Vision*, 7576:746–760, 2012. 2

[32] Xibin Song, Wei Li, Dingfu Zhou, Yuchao Dai, Jin Fang, Hongdong Li, and Liangjun Zhang. MLDA-Net: Multi-level dual attention-based network for self-supervised monocular depth estimation. *IEEE Transactions on Image Processing*, 30:4691–4705, 2021. 1

[33] Jaime Spencer, Richard Bowden, and Simon Hadfield. DeFeat-Net: General monocular depth via simultaneous unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14402–14413, 2020. 2, 3, 6, 7

[34] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9799–9809, 2019. 1, 2

[35] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979. 3

[36] Madhu Vankadari, Sourav Garg, Anima Majumder, Swagat Kumar, and Ardhendu Behera. Unsupervised monocular depth estimation for night-time images using adversarial domain feature adaptation. In *Proceedings of the European Conference on Computer Vision*, pages 443–459. Springer, 2020. 2, 3

[37] Madhu Vankadari, Stuart Golodetz, Sourav Garg, Sangyun Shin, Andrew Markham, and Niki Trigoni. When the Sun Goes Down: Repairing photometric losses for all-day depth estimation. In *Proceedings of the Conference on Robot Learning*, pages 1992–2003. PMLR, 2023. 2, 3, 6, 7, 8

[38] Kun Wang, Zhenyu Zhang, Zhiqiang Yan, Xiang Li, Baobei Xu, Jun Li, and Jian Yang. Regularizing Nighttime Weirdness: Efficient self-supervised monocular depth estimation in the dark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16055–16064, 2021. 2, 3, 6, 7

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

[40] Mengchen Xiong, Xiao Xu, Dong Yang, and Eckehard Steinbach. Robust depth estimation in foggy environments combining RGB images and mmWave radar. In *International Symposium on Multimedia*, pages 34–41. IEEE, 2022. 2, 3

[41] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. Channel-wise attention-based network for self-supervised monocular depth estimation. In *International Conference on 3D vision (3DV)*, pages 464–473. IEEE, 2021. 1

[42] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020. 6

[43] Chaoqiang Zhao, Yang Tang, and Qiyu Sun. Unsupervised monocular depth estimation in highly complex environments. *Transactions on Emerging Topics in Computational Intelligence*, 6(5):1237–1246, 2022. 3, 4

[44] Chaoqiang Zhao, Youmin Zhang, Matteo Poggi, Fabio Tosi, Xianda Guo, Zheng Zhu, Guan Huang, Yang Tang, and Stefano Mattoccia. MonoViT: Self-supervised monocular depth estimation with a vision transformer. In *Proceedings of the International Conference on 3D Vision (3DV)*, pages 668–678. IEEE, 2022. 3

[45] Ziqiang Zheng, Yang Wu, Xinran Han, and Jianbo Shi. ForkGAN: Seeing into the rainy night. In *Proceedings of the European Conference on Computer Vision*, pages 155–170. Springer, 2020. 6, 7, 8

[46] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 3, 4

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 7