Self-supervised Monocular Depth Estimation: Let's Talk About The Weather

Kieran Saunders
Aston University
Birmingham, UK
190229315@aston.ac.uk

George Vogiatzis
Loughborough University
Leicestershire, UK
g.vogiatzis@lboro.ac.uk

Luis J. Manso
Aston University
Birmingham, UK
l.manso@aston.ac.uk

Abstract

Current, self-supervised depth estimation architectures rely on clear and sunny weather scenes to train deep neural networks. However, in many locations, this assumption is too strong. For example in the UK (2021), 149 days consisted of rain. For these architectures to be effective in real-world applications, we must create models that can generalise to all weather conditions, times of the day and image qualities. Using a combination of computer graphics and generative models, one can augment existing sunny-weather data in a variety of ways that simulate adverse weather effects. While it is tempting to use such data augmentations for self-supervised depth, in the past this was shown to degrade performance instead of improving it. In this paper, we put forward a method that uses augmentations to remedy this problem. By exploiting the correspondence between unaugmented and augmented data we introduce a pseudo-supervised loss for both depth and pose estimation. This brings back some of the benefits of supervised learning while still not requiring any labels. We also make a series of practical recommendations which collectively offer a reliable, efficient framework for weather-related augmentation of self-supervised depth from monocular video. We present extensive testing to show that our method, Robust-Depth, achieves SotA performance on the KITTI dataset while significantly surpassing SotA on challenging, adverse condition data such as DrivingStereo, Foggy CityScape and NuScenes-Night. The project website can be found at https://kieran514.github.io/Robust-Depth-Project/.

1. Introduction

Depth estimation has been a pillar of computer vision for decades and has many applications, such as self-driving, robotics, and scene reconstruction. While multiple-view geometry is a well-understood computer vision problem, the advent of deep learning has enabled depth estimation from a single image. The first such methods used a supervised approach to estimate depth and required expensive ground truth data collected using LIDAR and Radar sensors. Recently, self-supervised monocular methods have been introduced, using photometric loss [57] to achieve view synthesis on consecutive images as a form of self-supervision. These methods have received wide interest because of their low cost and ability to generalize to multiple scenarios [22]. However, despite its potential, self-supervised Monocular Depth Estimation (MDE) has been hampered by adverse weather conditions and nighttime driving [46, 44]. The KITTI dataset [15], used extensively for MDE, only contains daytime, dry, sunny images rather than varied realistic conditions. Previous attempts have shown that training these methods on different domains, like nighttime data, lead to worse performance [29]. Figure 1 illustrates Monodepth2 [17], trained on the KITTI dataset, estimating depth accurately on sunny data but struggles when shifting domains to different weather conditions. Our method **Robust-Depth**, trained with the KITTI dataset and augmentations, is more robust to such changes.

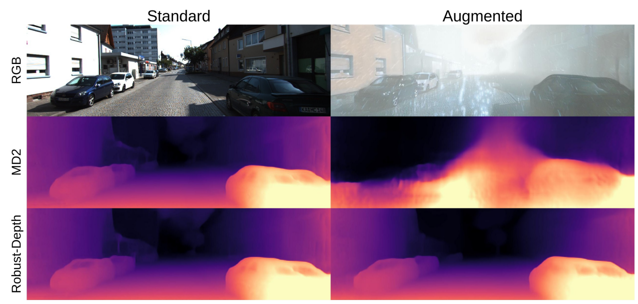

Figure 1. Monodepth2 demonstrates impressive performance on sunny scenes but struggles in different weather scenarios. Our method is more robust to environmental changes.

Unlike in other fields of computer vision [55, 35], dataset augmentation has led to worse performance for MDE (see Table 1). One possible explanation is that augmentations lead to texture shifts, and we know that CNNs are poor at generalising to texture shifts [3]. It is known [10, 34, 46] that depth networks become reliant on vertical cues (e.g., the output for a pixel being dependent on its vertical position in an image). Pixels at the bottom of the image are assumed to be close and pixels at the top are assumed to be far from the camera. While [46] suggests this is beneficial overall, these methods rely on potentially false cues. For example, a cliff-side image would pose difficulties to a system over-relying on vertical cues. In this work, we put forward a further observation that has to do with self-supervision itself: When performing data augmentation in supervised learning, we introduce some form of noise on the input image while maintaining a noise-free label. On the other hand, in self-supervised methods, the labels come from the data themselves. So augmenting a data point in a self-supervised method introduces noise in both data and target labels, leading to a much harder machine learning problem.

To overcome this challenge we propose a different formulation of the self-supervised loss, which exploits the alignment between the unaugmented and augmented data. Under that scheme, we are able to treat depth maps obtained from the unaugmented data as soft targets for the augmented depth estimations. Furthermore, with minimal extra effort, we can also treat depth maps of augmented images as labels for the unaugmented depth predictions leading to a fully symmetric bi-directional consistency constraint. We call this *pseudo-supervision loss* because it is an attempt to leverage the benefits of supervised learning (faster learning rates) in the self-supervised domain. For more detail see section 3.2. The paper also puts forward a number of recommendations for creating a robust, data-augmentation framework for MDE, each of which helps overcome the reliance on simplistic cues for depth. These include:

- Using the unaugmented images when warping the target image with the current depth map (sec. 3.2 eq. (6))
- Training in (unaugmented, augmented) pairs (sec. 3.2 eq. (7))
- Applying a one-way pseudo-supervision loss for pose estimation (sec. 3.2 eq. (13))

Finally, we propose a wide-ranging set of weather-related data augmentations together with vertical cropping and tilling, the effect of which is to move the focus of the network away from simplistic depth cues and towards deeper semantic cues. The paper contains an extensive experimental analysis, including an ablation study of the various algorithmic components as well as a comparison to State-of-the-Art (SotA). The analysis shows that our method significantly surpasses SotA on adverse weather data and performs as well or better on sunny weather data.

2. Related Work

2.1. Supervised Depth

Initially, depth estimation was cast as a simple regression task using ground truth depth data collected via one or multiple sensors. CNNs were first used for MDE in [12], and there have been many architectural and network developments over time [41, 26, 1]. Furthermore, supervised methods made the regression task a classification-regression task [13, 5, 28] as the regression tasks lead to substandard results. These networks relied on high-cost ground truth data and struggled with generalizing to other datasets.

2.2. Self-supervised Depth

2.2.1 Stereo

Garg *et al.* [14] used view synthesis as a method for self-supervised learning for stereo depth between stereo pairs. Later, Monodepth [16] used photometric loss, made from L_1 loss and SSIM [47], to demand left-right consistency between left-right reconstruction.

2.2.2 Monocular

Our work focuses on monocular cameras, as stereo setups are more expensive and require more space. The first technique to use view synthesis for MDE was SfM-learner [57]. This and the methods to follow use a depth network with pose estimations to warp images and maximize photometric uniformity. Monodepth2 (MD2) [17] introduced the per-pixel minimization of the photometric error to avoid occlusion issues while also introducing auto-masking to handle texture-less regions and dynamic objects.

One issue with MDE is that pose, and therefore depth, can only be estimated up to an unknown scale causing Monodepth2 to struggle with maintaining consistent scales between frames. To counter this, SC-SfM-Learner [6] created a differentiable geometric loss and masked out pixels with geometric inconsistency. This led to higher inter-frame depth consistency but is computationally more expensive for each iteration. Other approaches make use of cost volumes for their efficiency, specifically, Manydepth [48] and MonoRec [49] use temporal image sequences that allow for geometrical reasoning during inference. They use multiple pre-defined depths hypotheses to warp reference frame features to the target image frame, and their differences create a cost volume. The depth hypothesis closest to the true depth leads to low values in the cost volume.

Moreover, RA-depth [20] developed a useful augmentation that leads to more scale consistency and better depth estimation when changing resolutions. This is crucial as these self-supervised methods have no reference for scale and, at the very least, should have consistent depth in the same image at different scales.

In parallel to new training methodologies, there is work on improving architectures, and some recent research has been focused on increasing efficiency while increasing accuracy [30, 18, 56, 52]. We will focus on applying our contributions to DepthNet from Monodepth2 [17] and the current SotA depth network [54] to show improvements. As demonstrated in [45, 3] transformer networks are more robust than their CNN equivalent and self-supervised methods are more robust and generalisable than supervised methods. An explanation for this is that transformer networks are more shape-biased versus the texture bias of CNNs, allowing transformers to be more accurate on out-of-distribution data and texture variations (such as style transfers). This inspires us to use MonoViT [54] as our final model because it is lightweight, accurate and more robust than a pure CNN network. We must note however that the methodology proposed in this paper is agnostic to the specifics of the neural model used for depth or pose estimation.

2.3. Data Augmentation

Many previous methods have attempted to solve the lack of robustness problem with MDE [46, 44, 42, 19, 2, 29]. A common theme with these papers is that they can only handle few variations in environments and require large and complex architectural modifications to achieve realistic depth for the augmentations. Some also heavily rely on synthetic data, which leads to the models having a domain bias (from real to synthetic) [2, 19]. To our knowledge, there has been a limited use of augmentations in MDE [23, 24, 4], especially self-supervised MDE. Simple colour jitters and horizontal flips have been used in the past but, more complicated augmentations have been avoided as they are typically found to degrade depth accuracy.

ADDS-DepthNet [29] is the most similar to our method as they create night image pairs from day images using GANs. In doing so, they train their network to be effective in two domains, night and day. They use the day depth estimation as pseudo-supervision for their depth estimation for night scenes. This method is one-directional and limits the depth estimation to only be as good as the day depth estimation. Also, they ignore pose estimation and focus on creating a reconstruction of the GAN-generated night image. We avoid reconstruction of the augmented images as the GAN-generated augmentations lack consistency between frames, especially for the night domain. Furthermore, our model is unique in that it is able to handle multiple domains with greater accuracy.

Another approach, ITDFA [53], uses a trained fixed depth decoder from Monodepth2 [17] and changes the encoders for each domain to force consistency between features in different domains. Also, ITDFA enforces consistency between depth images for both day and night. This has obvious drawbacks compared to our model; firstly, we use the same encoder and decoder for all different augmentations, which can be trained end-to-end. Secondly, the depth estimations from these augmented images are limited to the accuracy of Monodepth2. Our method avoids this issue by only updating the augmented depth where the unaugmented depth has a lower photometric loss (sec. 3.2).

Furthermore, [10] highlighted the fact that MDE tends to learn only a small number of cues and shows great dependency on the position of the pixel to estimate depth. EPC-depth [34] demonstrated that the use of data grafting for input images leads to greater performance with respect to stereo depth. Inspired by this and to help MDE learn new and more varied cues, we apply vertical cropping to the input images. Also, we apply tile cropping to the input images, to enforce equivariance for depth estimation. Similarly, RA-depth [20] used augmentation to change the scales of the input image, allowing the depth estimation to be more scale consistent. A modified version of this idea has been employed in our method.

3. Method

Overview: In this paper, we put forward **Robust-Depth**, a self-supervised architecture that is able to estimate depth and ego-motion while being robust to many different augmentations. We highlight SotA's limited robustness in other weather scenarios, over-dependence on pixel position, and its lack of ability to generalise to other datasets. All of this is achieved with a minor increase in computation and memory requirements while using the standard DepthNet backbone from Monodepth2 [17] and the current SotA network MonoViT [54]. The applications of our novel contributions are shown in Figure 2.

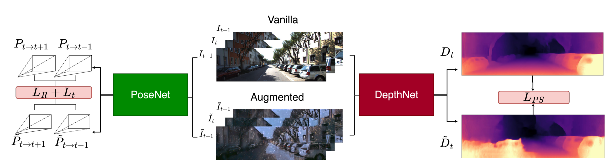

Figure 2. The original image (top-middle) and the augmented image (bottom-middle) are each fed into the pose and depth estimation networks. Our pseudo-supervision loss function contains terms that encourage consistency between the depth map and pose estimates of the original and augmented image.

3.1. Preliminaries

Following Zhou et al. [57] we simultaneously train an ego-motion network with a depth network to execute view-synthesis between consecutive frames. We use the target depth estimation $D_t = DepthNet(I_t)$ and camera pose estimations $P_{t \to t'} = PoseNet(I_t, I'_t)$ to synthesise the target image where $I_{t'} \in \{I_{t-1}, I_{t+1}\}$, only using source frames. PoseNet and DepthNet denote the pose and depth estimation networks respectively. The synthesised image obtained is projected using inverse warping as shown below.

$$I_{t' \to t} = I_{t'} \langle Proj(D_t, P_{t \to t'}, K) \rangle \tag{1}$$

Proj() outputs the resulting 2D coordinates of the depths after projecting into the camera of frame $I_{t'}$, and $\langle\rangle$ is the sampling operator. We will be using photometric loss (pe) which is defined below,

$$pe(I_a, I_b) = \frac{\alpha}{2}(1 - SSIM(I_a, I_b)) + (1 - \alpha)||I_a - I_b||, \tag{2}$$

where $\alpha = 0.85$. In Monodepth2, the per-pixel photometric loss used for training the pose and depth network is simply:

$$L_p = \min_{t'} pe(I_t, I_{t' \to t}) \tag{3}$$

Where for each pixel we decide whether to use the next or the previous frames to reproject, based on the minimization of reprojection error. Moreover, unlike some methods that advocate using only one layer of the depth network to increase learning speed [27, 20], we use the multi-scale proposal from Monodepth2.

3.2. Bi-directional pseudo-supervision Loss

Previous methods that used augmented images for self-supervised MDE attempted to create reconstructions of the augmented images [29], as shown below.

$$\tilde{I}_{t' \to t} = \tilde{I}_{t'} \langle Proj(\tilde{D}_t, \tilde{P}_{t \to t'}, K) \rangle \tag{4}$$

Where $\tilde{I}_{t'}$ represents the augmented input image, $\tilde{D}_t$ is the depth estimation of the augmented image, and $\tilde{P}_{t \to t'}$ is the pose estimation obtained from the augmented images. They would then proceed with the standard loss function;

$$L_p = \min_{t'} pe(\tilde{I}_t, \tilde{I}_{t' \to t}) \tag{5}$$

The augmented input images are generally created using GANs and lack consistency between consecutive frames. This will cause the reconstructed augmented images $\tilde{I}_{t' \to t}$ to be dissimilar to the true target image as there may be changes between frames that can not be accounted for, e.g. illumination changes and rain streaks. Furthermore, the pose estimation will suffer from the same problem as it requires consecutive frame inputs, which in the case of augmented images, will not be consistent. To counter these issues we introduce semi-augmented warping;

$$\overset{*}{I}_{t' \to t} = I_{t'} \langle Proj(\tilde{D}_t, P_{t \to t'}, K) \rangle. \tag{6}$$

Semi-augmented warping exploits the consistency between the unaugmented frames while using the depth estimations from the augmented images. Furthermore, we use the pose estimation from the unaugmented input to ensure that the depth estimation is isolated from the errors of pose estimation under the various augmentations.

Simply optimising the photometric loss for the augmented reprojected image from equation (4) can lead to catastrophic forgetting. This happens because the network shifts its focus onto the augmented data and forgets the original data (see Table 1). To prevent this we always train images in pairs (augmented, unaugmented) as follows.

$$L_p = \min_{t'}(pe(I_t, I_{t' \to t})) + \min_{t'}(pe(I_t, \overset{*}{I}_{t' \to t})). \tag{7}$$

There are several possible formulations of the loss function that achieve the same objective, however, we found this simple pair-training scheme works well in practice.

The photometric loss given above is responsible for enforcing the self-supervision constraint. However, the relationship between the augmented and unaugmented data gives us an opportunity to exploit the faster learning rate of supervised learning. To see this, assume we had access to a fully trained network that can correctly predict depth in the unaugmented, original images. We note that each augmented image should correspond to the same underlying depth map as the original image it was derived from since all augmentations considered do not change the underlying depth map. We could therefore obtain that depth map using the fully trained network on the original image, and then use that depth map as a label in a supervised regression loss.

In practice, we do not have access to such a fully trained network on the original images. During training, we can expect that our depth network will either perform better in the augmented or the unaugmented image at any given time. However, using the photometric reprojection loss, we can easily find out which of the two depths gives a better reprojection and use that depth as a label for the other depth map. Furthermore, this analysis can be done per-pixel leading to an efficient scheme which we term pseudo-supervision loss.

We define the following per-pixel mask that picks out pixels where the unaugmented image gives a better depth than the augmented one

$$M_v = [\min_{t'} pe(I_t, I_{t' \to t}) < \min_{t'} pe(I_t, \overset{*}{I}_{t' \to t})] \odot \mu, \tag{8}$$

where [ ] are the Iverson brackets and $\odot$ is the pixel-wise product. We define μ as auto-masking from Monodepth2 [17], which is applied to our mask M_v to remove stationary pixels. We apply this same concept in the opposite direction for pixels where the augmented depth leads to a smaller reprojection loss than the unaugmented depth estimation.

$$M_a = [\min_{t'} pe(I_t, \overset{*}{I}_{t' \to t}) < \min_{t'} pe(I_t, I_{t' \to t})] \odot \mu \tag{9}$$

Note, we cannot simply find the inverse of this mask, as we have to account for the auto-masked pixels $M_a \neq 1 - M_v$. We can now introduce the components of bi-directional pseudo-supervision depth loss;

$$L_a = log(|\tilde{D}_t - \underline{D_t}| + 1) \odot M_v, \tag{10}$$

$$L_v = log(|D_t - \underline{\tilde{D}_t}| + 1) \odot M_a. \tag{11}$$

The underlining of $\underline{D_t}$ and $\underline{\tilde{D}_t}$ signifies that the gradients of D_t and $\tilde{D}_t$ are cut off during backpropagation. This gives us the bi-directional pseudo-supervision loss:

$$L_{PS} = L_a + L_v. \tag{12}$$

L_a is the loss caused by a worse augmented depth estimation compared to the unaugmented depth estimation, and L_v is the loss caused by a worse unaugmented depth estimation than augmented depth estimation, summing to make the bi-direction pseudo-supervision loss L_{PS}. We show the contribution of each component to the bi-direction pseudo-supervision loss in the supplementary materials. The analysis above also applies to the pose estimation part of the algorithm. The rotation/translation that links two consecutive frames remains the same after these images are augmented. Once again, we can introduce a bi-directional consistency loss using reprojection quality to judge which pose to use as a label. However in experiments, this was found to have negligible improvement, so we assume that the pose estimated from the original images will be closer to the correct one. This leads to the pseudo-supervised pose loss,

$$L_R = |\tilde{R} - \underline{R}|, \quad L_t = |\tilde{t} - \underline{t}|, \tag{13}$$

where $\tilde{R}$ and $\tilde{t}$ are the rotation and translation obtained from the augmented images and $\underline{R}$, $\underline{t}$ from the original. The underlining again denotes that the gradients of R and t are not used when updating the pose network weights.

Final Loss: Our final loss is a combination of photometric loss, bi-directional pseudo-supervision loss, pseudo-supervision pose loss, and edge-aware smoothness loss (L_s) from [38]. Where edge-aware smoothness loss is applied to both the unaugmented depth and augmented depth. Giving a final loss, which averages over each pixel, scale and batch;

$$L = \mu L_p + \omega L_{PS} + \beta(L_R + L_t) + \gamma L_s. \tag{14}$$

3.3. Data preparation & augmentations

Previous depth estimation methods struggled with variations in weather. Fog, rain, extreme brightness, night time and motion blur have all been a challenge for all such depth networks. There have been few attempts at solving some of these issues, but they suffer from poor results [46, 44, 42, 53], or they use simulation data that struggles with the domain shift to real-world application [2, 19].

We use a physics-based renderer (PBR) [43] to create realistic rain and fog augmentations on the KITTI dataset and store these prior to training the presented architecture. Similarly, we use CoMoGAN [36] to augment the KITTI dataset with realistic night-time, dawn and dusk. Furthermore, we make all the combinations of these weather-based augmentations, for example, rain+fog or rain+fog+dawn. Separately, we also add ground snow as it is a great source of corruption and motion blur using Automold [40]. We use grey-scale images and break the image down into its red, green and blue components to remove the effects of colour on depth estimation. This is important for large shifts in image colour, for example, between autumn and winter. To achieve extreme brightness, we increase the brightness capabilities on the already implemented colour jitter.

To further improve the robustness of self-supervised MDE, we took inspiration from robustness testing on other tasks [32, 21]. [45] demonstrated the robustness of monocular depth using transformer depth networks with corruption augmentations on the test sets. Rather than solely relying on the underlying network for robustness, we implement these augmentations prior to training. Specifically, we add Gaussian noise, shot noise, impulse noise, defocus blur, frosted glass blur, zoom blur, snow, frost, elastic, pixelated, and jpeg augmentation variants to the KITTI dataset. The percentage and distribution of the augmentations are provided in the supplementary materials.

During training, we use a modified scale augmentation from RA-Depth [20], vertical cropping augmentation, and tile augmentation created especially for this architecture so that the augmentations can be reverted later. Furthermore, standard random erase is used as an augmentation. Also, note that due to the lack of variety of data used for self-supervised monocular depth, all of these augmentations will reduce the chance that MDE overfits naive cues such as vertical dependency, texture bias and apparent size of objects.

Vertical cropping augmentation: This augmentation was inspired by EPC-Depth [34] grafting augmentation. We randomly select value τ between $\{0.2,0.4,0.6,0.8\}$ with uniform probability. We then select this percent of the input image from the top down and crop it to be moved underneath the section below it, all of this occurs on each minibatch.

Tile augmentation: For tile cropping, we split the width of the image by two or four, and the height of the image by two or three randomly. We crop each section and then randomly shuffle the tiles to a new position, all of this occurs on each minibatch.

4. Results

Experimental set-up: We train the model starting with pretrained ImageNet weights [9] using PyTorch [33] and train on an NVIDIA A6000 GPU. We use the Adam optimiser [25] for 30 epochs, with an input size of 640×192 and set a starting learning rate of 10^{-4}. Progressively reducing the learning rate using multi-step learning rate decay at epoch $[20, 25, 29]$ by 0.1. We set the hyperparameters ω, β and γ to 0.01, 0.01 and 0.001, respectively. In all tables, Robust-Depth† represents Robust-Depth trained with just weather and time-specific augmentations. Finally, Robust-Depth* represents Robust-Depth using MonoViTs transformer backbone instead of the baseline (Monodepth2)

Ablation	Eq. (7)	Eq. (6) Depth	Eq. (6) Pose	L_{PS}	L_R+ L_t	Test	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 (*Sunny*) [17]						*Sunny*	0.115	0.903	4.863	0.193	0.877	0.959	0.981
						Bad w.	0.249	2.477	7.962	0.347	0.612	0.824	0.918
Monodepth2 (*Bad w.*) [17]						*Sunny*	0.118	0.920	4.814	0.193	0.869	0.958	0.981
						Bad w.	0.140	1.069	5.408	0.220	0.821	0.940	0.975
Robust-Depth	✓					*Sunny*	0.116	0.948	4.884	0.194	0.875	0.959	0.981
						Bad w.	0.155	1.623	5.677	0.232	0.812	0.933	0.970
Robust-Depth	✓	✓				*Sunny*	0.116	0.881	4.912	0.196	0.871	0.958	0.980
						Bad w.	0.135	1.048	5.351	0.216	0.830	0.944	0.976
Robust-Depth	✓		✓			*Sunny*	0.116	0.890	4.872	0.192	0.869	0.959	0.982
						Bad w.	0.138	1.066	5.481	0.219	0.821	0.941	0.976
Robust-Depth	✓			✓		*Sunny*	0.116	0.931	4.816	0.193	0.877	0.959	0.981
						Bad w.	0.139	1.136	5.439	0.219	0.827	0.942	0.975
Robust-Depth	✓				✓	*Sunny*	0.116	0.942	4.895	0.194	0.874	0.958	0.981
						Bad w.	0.146	1.249	5.533	0.226	0.818	0.937	0.972
Robust-Depth All	✓	✓	✓	✓	✓	*Sunny*	0.115	0.937	4.840	0.193	0.873	0.959	0.981
						Bad w.	0.133	1.115	5.259	0.211	0.842	0.948	0.977

Table 1. **Ablation Study:** We demonstrate the addition of each component to the baseline Monodepth2. Monodepth2 (*Sunny*) represents Monodepth2 trained on the original KITTI dataset and (*Bad w.*) represents training with augmentations. Each method is tested in the *sunny* and *bad weather* datasets (white and grey rows respectively).

ResNet18.

4.1. Datasets

KITTI [15]: We use the official Zhou split for validation data consisting of 4,424 images and train on the entire 39,810 images from the training set. For testing, we use 697 images from the Eigen *et al.* test set [11]. An in-depth discussion of how we created the variants of this dataset is provided in the supplementary material. This includes all the generation parameters that determine how to reproduce the augmentations when replicating the experiments. All methods results to follow are trained on the KITTI dataset, unless specified. By *sunny* we denote the raw, unaugmented KITTI dataset while *bad weather* denotes the KITTI dataset augmented with adverse weather effects and other image modifications as laid out in sec. 3.3.

DrivingStereo [51]: As the aim of this paper is to demonstrate the capacity to infer depth in different domains, the DrivingStereo dataset provides a selection of foggy, cloudy, rainy and sunny images for testing, all containing 500 images.

Foggy Cityscape [39, 8]: To test methods on more severe foggy scenes, we use the Foggy CityScape dataset as we use the data provided with the beta parameter equal to 0.02 (most severe fog) consisting of 1525 test images.

NuScenes [7]: Nighttime scenarios are a challenging domain for many self-supervised monocular depth methods, and we can use NuScenes to explicitly test the ability of each method to infer depth in this domain. We select the nighttime splits from [46] test selection and test on 500 real nighttime images.

4.2. Ablation Study:

This subsection provides a detailed discussion of the results of the ablation study (Table 1). We test using the *sunny* and *bad weather* datasets to demonstrate quantitatively how well a model can handle sunny weather as well as image degradation and varying weather conditions, including time of day. Initially, when testing Monodepth2 with the *bad weather* test set (row 1), we obtain a disappointing performance, indicating a lack of ability to handle weather conditions and image degradation.

Optimising for augmented data (Row 2): When attempting to train Monodepth2 with augmented data, in the naive way we outline in equations (4) and (5), we see a significant improvement in robustness but worse performance for the *sunny* test. Here we train with augmented data in the traditional sense, including unaugmented images. These results inspired our method, which by design, attempts to maintain a balance between focus on sunny and weather-augmented data.

Optimising for unaugmented & augmented data (Row 3): Optimising for both the *sunny* and *bad weather* data simultaneously (see equation (7)) leads to the best of both worlds, a form of trade-off. Here we emphasise that we are using the augmented reprojection from equation (4) and not semi-augmented warping from equation (6). This formulation achieves *sunny* test results almost as good as the standard Monodepth2 and yet only slightly worse performance for robustness compared to naively training Monodepth2 with augmented data. The worse performance for the *bad weather* testing is because the network trained on only augmented data sufferers from catastrophic forgetting. All rows to follow will optimise for both *sunny* and *bad weather* data.

Semi-augmented Warping (Row 4 & 5): We explore the effect of equation (6) into two different parts. Firstly, we investigate the effects of warping the original image while using the pose and depth obtained from the augmented image (row 4). From this, we see that, due to the inconsistency of the augmented images, there is a significant improvement from exploiting the consistency of unaugmented images. Secondly, we evaluate the effect of using the unaugmented pose estimation while warping the augmented images (row

Method	W × H	Test	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ranjan *et al*. [38]	832 × 256	*Sunny*	0.148	1.149	5.464	0.226	0.815	0.935	0.973
		Bad w.	0.262	2.263	7.757	0.3560	0.570	0.811	0.919
Monodepth2 [17]	640 × 192	*Sunny*	0.115	0.903	4.863	0.193	0.877	0.959	0.981
		Bad w.	0.249	2.477	7.962	0.347	0.612	0.824	0.918
Mono-Uncertainty[37]	640 × 192	*Sunny*	0.111	0.863	4.756	0.188	0.881	0.961	0.982
		Bad w.	0.243	2.287	7.806	0.341	0.613	0.831	0.923
HR-Depth [30]	640 × 192	*Sunny*	0.109	0.792	4.632	0.185	0.884	0.962	0.983
		Bad w.	0.273	2.544	8.408	0.380	0.552	0.793	0.901
CADepth [50]	640 × 192	*Sunny*	0.105	0.769	4.535	0.181	0.892	0.964	0.983
		Bad w.	0.266	2.530	8.145	0.365	0.575	0.801	0.908
DIFFNet [56]	640 × 192	*Sunny*	0.102	0.749	<u>4.445</u>	0.179	<u>0.897</u>	0.965	0.983
		Bad w.	0.202	1.724	7.198	0.304	0.680	0.870	0.940
MonoViT [54]	640 × 192	*Sunny*	**0.099**	**0.708**	**4.372**	**0.175**	**0.900**	**0.967**	**0.984**
		Bad w.	0.167	1.347	6.385	0.258	0.751	0.909	0.961
Robust-Depth	640 × 192	*Sunny*	0.115	0.937	4.840	0.193	0.873	0.959	0.981
		Bad w.	<u>0.133</u>	<u>1.115</u>	<u>5.259</u>	<u>0.211</u>	<u>0.842</u>	<u>0.948</u>	<u>0.977</u>
Robust-Depth*	640 × 192	*Sunny*	<u>0.100</u>	<u>0.747</u>	4.455	<u>0.177</u>	0.895	<u>0.966</u>	**0.984**
		Bad w.	**0.114**	**0.891**	**4.878**	**0.193**	**0.868**	**0.958**	**0.981**

Table 2. **Quantitative Results for the KITTI Eigen Test Dataset.** Note that we do not use any test time refinement and only require one frame to do inference for depth estimation. All methods have been pretrained on ImageNet [9] and **bold** text represents the best result for the metric for each test. Whereas, the <u>underlined</u> text represents the second best for each metric for each test.

5). Again, a notable reduction is shown in *bad weather* test metrics. This affirms our assumption that the unaugmented pose estimations will result in the most accurate pose, as the pose network relies heavily on consistency between two frames.

Pseudo-supervision depth loss (Row 6): We aim for consistency between unaugmented and augmented depth estimations. This also leads to significant loss reductions in the *bad weather* test set, which supports the use of bi-directional pseudo-supervision, using unaugmented image depth estimations to improve the depth estimations of augmented images and vice versa.

Pseudo-supervision pose loss (Row 7): Although using unaugmented pose estimation for inverse warping augmented images is beneficial, we can still try to improve the robustness of the pose network by using the pseudo-supervised scheme for improving pose estimation. This again shows a reduction in *bad weather* metrics, and we would expect this is making the pose network more accurate. The reasoning behind this is that the GANs inconsistency will cause the pose of the augmented images to be worse than the unaugmented pose, and the pose loss will help the pose network be more robust.

All (Row 8): Our model trained with all of the contributions leads to *sunny* test results close to Monodepth2 and the best *bad weather* performance. We use Monodepth2 as our baseline, but we expect to see the same pattern when using any network. The pseudo-supervised depth loss leads to a 10.3% reduction. On the other hand, semi-augmented warping (equation (6)) leads to a 12.9% reduction in the absolute relative error for the *bad weather* test. Therefore, out of all the algorithmic components, warping with unaugmented images leads to the most significant reduction in the *bad weather* test metrics. In the final row, we confirm that our method of training with augmented data leads to the best performance on the augmented data, significantly outperforming the baseline model trained with augmented data, while maintaining *sunny* data depth performance.

4.3. Comparison with SotA

Baseline models in Table 2 have not been trained on augmented data, but, just like we demonstrated in Table 1 when applying our contribution to any of these methods as a baseline we would expect to see the same patterns; improved *bad weather* metrics and maintained *sunny* metrics.

Table 2 shows our model's results on the KITTI Eigen test set [11] compared with previous and current state-of-the-art methods/architectures for self-supervised MDE. There is strong evidence that Robust-Depth is able to perform at the same level as Monodepth2 for the original KITTI dataset, but performs significantly better than all previous state-of-the-art for the *bad weather* test set. This demonstrates the depth networks' robustness to a variety of image degradation, weather and time of day changes.

Note that the previous state-of-the-art, along with improved methods, have mainly developed more sophisticated architectures which have led to improvements in depth. Many recent works have demonstrated better generalisability to domain changes from transformer networks. To make this comparison fair we have trained our model with the standard Monodepth2 (ResNet18) architecture and a second model with the same transformer backbone as MonoViT (Robust-Depth*). From Robust-Depth*, we see maintained *sunny* depth abilities and impressive *bad weather* results, further demonstrating that our method is able to handle augmentations and thus changes in domains.

4.4. Further Testing

The data used to do the robustness tests up to this point are sampled from the same distribution that our model was trained on. To truly demonstrate our model's capabilities on all of these weather changes and nighttime scenes, we test

Domain	Method	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Foggy	Monodepth2 [17]	0.143	1.952	9.817	0.218	0.812	0.937	0.974
	HR-Depth [30]	0.154	2.112	10.116	0.225	0.786	0.933	0.977
	CADepth [50]	0.141	1.778	9.448	0.208	0.809	0.945	0.981
	DIFFNet [56]	0.125	1.560	8.724	0.188	0.840	0.956	0.985
	MonoViT [54]	0.109	1.206	7.758	0.167	0.870	0.967	0.990
	Robust-Depth	0.140	1.907	9.098	0.203	0.827	0.949	0.980
	Robust-Depth†	0.138	1.655	9.064	0.203	0.817	0.950	0.983
	Robust-Depth*	**0.105**	**1.135**	**7.276**	**0.158**	**0.882**	**0.974**	**0.992**
Cloudy	Monodepth2 [17]	0.170	2.211	8.453	0.232	0.781	0.932	0.973
	HR-Depth [30]	0.173	2.424	8.592	0.237	0.783	0.927	0.972
	CADepth [50]	0.161	2.086	8.167	0.222	0.804	0.936	0.976
	DIFFNet [56]	0.154	1.839	7.679	0.212	0.809	0.941	0.978
	MonoViT [54]	**0.141**	**1.626**	7.550	**0.201**	**0.831**	**0.948**	**0.981**
	Robust-Depth	0.173	2.281	8.269	0.231	0.782	0.933	0.973
	Robust-Depth†	0.168	1.969	8.363	0.231	0.775	0.932	0.975
	Robust-Depth*	0.148	1.781	**7.472**	0.204	0.825	0.947	**0.981**
Rainy	Monodepth2 [17]	0.245	3.641	12.282	0.310	0.600	0.852	0.945
	HR-Depth [30]	0.267	4.270	12.750	0.331	0.593	0.833	0.932
	CADepth [50]	0.226	3.338	11.828	0.288	0.633	0.874	0.956
	DIFFNet [56]	0.197	2.669	10.682	0.256	0.678	0.904	0.971
	MonoViT [54]	0.175	2.138	9.616	0.232	0.730	0.931	0.979
	Robust-Depth	0.199	2.670	10.595	0.260	0.677	0.902	0.972
	Robust-Depth†	0.182	2.250	10.241	0.247	0.711	0.910	0.973
	Robust-Depth*	**0.167**	**2.019**	**9.157**	**0.221**	**0.755**	**0.938**	**0.982**
Sunny	Monodepth2 [17]	0.177	2.103	8.209	0.240	0.782	0.925	0.968
	HR-Depth [30]	0.173	1.910	7.924	0.233	0.768	0.932	0.975
	CADepth [50]	0.164	1.838	7.890	0.225	0.794	0.936	0.976
	DIFFNet [56]	0.162	1.755	7.489	0.221	0.801	0.936	0.974
	MonoViT [54]	**0.150**	1.615	7.657	0.211	**0.815**	0.943	**0.979**
	Robust-Depth	0.185	2.174	8.084	0.246	0.765	0.919	0.965
	Robust-Depth†	0.181	1.922	8.190	0.245	0.757	0.921	0.968
	Robust-Depth*	0.152	**1.574**	**7.293**	**0.210**	0.812	**0.944**	**0.979**

Table 3. **Results on DrivingStereo Dataset [51].** Models were trained on the KITTI dataset and tested on four different domains. Robust-Depth* represents Robust-Depth trained with MonoViT's backbone and Robust-Depth† is Robust-Depth trained with weather and time-specific augmentations.

on out-of-distribution data. We first compare our method using the DrivingStereo dataset to display depth estimation performance in different domains. Specifically, we are testing on foggy, cloudy, rainy and sunny data. We show, in Table 3, impressive performance improvements on both the foggy and rainy domains but also difficulties with cloudy and sunny domains. This can be explained by our depth network being to some degree biased towards *bad weather*. Furthermore, both the foggy and rainy domains demonstrate light weather conditions, and we would expect our results to be more pronounced for more pronounced weather conditions. Furthering this point, we show our results on the Foggy CityScape scenario in Table 4, which show significant improvements. As this fog is denser, our models perform better than others. This foggy setting, although synthetic, is more realistic as it represents a more dense foggy layer, and we expect our model to perform better for more extreme weather conditions.

Method	Foggy CityScapes						
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [17]	0.208	3.093	12.447	0.337	0.656	0.842	0.917
HR-Depth [30]	0.212	3.012	12.263	0.336	0.642	0.841	0.920
CADepth [50]	0.207	2.738	11.542	0.318	0.650	0.856	0.933
DIFFNet† [56]	0.187	2.583	11.337	0.302	0.698	0.867	0.937
MonoViT [54]	0.155	1.873	9.585	0.244	0.771	0.910	0.967
Robust-Depth	0.160	1.569	7.912	0.224	0.757	0.937	0.982
Robust-Depth†	0.165	1.561	8.073	0.227	0.750	0.938	0.983
Robust-Depth*	**0.127**	**1.038**	**6.604**	**0.180**	**0.847**	**0.967**	**0.991**

Table 4. **Results on Foggy CityScape Dataset [39].** All models are trained on the KITTI dataset.

For both the Drivingstereo and Foggy CityScape datasets, Robust-Depth trained with MonoViTs backbone returns a similar pattern. When using Mono-ViT as our backbone, we achieve remarkable performance in all domains. Moreover, when Robust-Depth is trained with only weather and time-specific augmentations, we see a respectable performance improvement. For a final test of our method, we use the NuScene-Night dataset defined by [46] to test the capabilities of the model at night. From Table 5, we find a significant improvement in nighttime understanding from Robust-Depth, and when using MonoViTs backbone the improvements become even more pronounced. It is worth noting that our method outperforms ADDS-DepthNet, a model specifically trained to handle night data.

Method	NuScenes-Night						
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [17]	0.397	6.206	14.569	0.568	0.378	0.650	0.794
HR-Depth [30]	0.461	6.633	15.028	0.622	0.301	0.571	0.749
CADepth [50]	0.421	5.949	14.509	0.593	0.331	0.613	0.776
DIFFNet [56]	0.344	4.851	13.152	0.491	0.440	0.710	0.838
ADDS-DepthNet [29]	0.415	5.693	14.905	0.590	0.343	0.594	0.766
MonoViT [54]	0.313	4.143	12.252	0.455	0.485	0.736	0.858
Robust-Depth	0.355	6.344	12.510	0.457	0.523	0.762	0.862
Robust-Depth†	0.344	4.873	12.290	0.458	0.461	0.752	0.862
Robust-Depth*	**0.276**	**4.075**	**10.470**	**0.380**	**0.607**	**0.819**	**0.912**

Table 5. **Results on NuScenes-Night data [7].** All models were trained at a resolution of 640×192 except ADDS-DepthNet at 256×512. ADDS-DepthNet is trained with RobotCar dataset [31].

5. Conclusion

In this paper, we show how self-supervised MDE can overcome adverse weather effects, a major obstacle to real-world applications (e.g. autonomous vehicles). The solution we put forward is a carefully constructed, architecture-agnostic data-augmentation scheme. Central to our scheme is bi-directional pseudo-supervision loss, a novel loss function which uses unaugmented depth estimations to self-supervise the augmented depth estimations and vice versa. The paper also introduces several other algorithmic steps, each of which improves depth estimation even under extremely harsh image degradations. This is done without significant increases in memory requirements or computing. The set of data augmentations we use enables the depth network to extract more varied and reliable cues for depth, leading to significantly better depth estimation than SotA in adverse weather, and equally good or better than SotA even in fine weather. The same proposed techniques can be used in other pixel-wise estimation tasks in the future to improve their robustness to similar augmentations.

6. Acknowledgement

This research was funded and supported by the EPSRC's DTP, Grant EP/W524566/1. Most experiments were run on Aston EPS Machine Learning Server, funded by the EPSRC Core Equipment Fund, Grant EP/V036106/1.

References

[1] Ashutosh Agarwal and Chetan Arora. Attention attention everywhere: Monocular depth prediction with skip attention. *arXiv preprint arXiv:2210.09071*, 2022. 2

[2] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2800–2810, 2018. 3, 5

[3] Jinwoo Bae, Kyumin Hwang, and Sunghoon Im. A study on the generality of neural network structures for monocular depth estimation. *arXiv preprint arXiv:2301.03169*, 2023. 1, 3

[4] Jongbeom Baek, Gyeongnyeon Kim, Seonghoon Park, Honggyu An, Matteo Poggi, and Seungryong Kim. Semi-supervised learning of monocular depth estimation via consistency regularization with k-way disjoint masking, 2022. 3

[5] 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. 2

[6] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Zhichao Li, Le Zhang, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth learning from video. *International Journal of Computer Vision*, 129(9):2548–2564, 2021. 2

[7] 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. 6, 8

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 6

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5, 7

[10] Tom van Dijk and Guido de Croon. How do neural networks see depth in single images? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2183–2191, 2019. 1, 3

[11] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 6, 7

[12] 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

[13] 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

[14] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016. 2

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 1, 6

[16] 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. 2

[17] 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

[18] 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. 2

[19] Akhil Gurram, Ahmet Faruk Tuna, Fengyi Shen, Onay Urfalioglu, and Antonio M López. Monocular depth estimation through virtual-world supervision and real-world sfm self-supervision. *IEEE Transactions on Intelligent Transportation Systems*, 2021. 3, 5

[20] Mu He, Le Hui, Yikai Bian, Jian Ren, Jin Xie, and Jian Yang. Ra-depth: Resolution adaptive self-supervised monocular depth estimation. In *European Conference on Computer Vision*, pages 565–581. Springer, 2022. 2, 3, 4, 5

[21] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. 5

[22] Hanjiang Hu, Baoquan Yang, Zhijian Qiao, Ding Zhao, and Hesheng Wang. Seasondepth: Cross-season monocular depth prediction dataset and benchmark under multiple environments. *arXiv preprint arXiv:2011.04408*, 2020. 1

[23] Yasunori Ishii and Takayoshi Yamashita. Cutdepth: Edge-aware data augmentation in depth estimation. *arXiv preprint arXiv:2107.07684*, 2021. 3

[24] Doyeon Kim, Woonghyun Ga, Pyungwhan Ahn, Donggyu Joo, Sehwan Chun, and Junmo Kim. Global-local path networks for monocular depth estimation with vertical cutdepth. *arXiv preprint arXiv:2201.07436*, 2022. 3

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[26] 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

[27] Seokju Lee, Sunghoon Im, Stephen Lin, and In So Kweon. Learning monocular depth in dynamic scenes via instance-aware projection consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1863–1872, 2021. 4

[28] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022. 2

[29] 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. 1, 3, 4, 8

[30] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: High resolution self-supervised monocular depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2294–2301, 2021. 2, 7, 8

[31] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 8

[32] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019. 5

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[34] Rui Peng, Ronggang Wang, Yawen Lai, Luyang Tang, and Yangang Cai. Excavating the potential capacity of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15560–15569, 2021. 1, 3, 5

[35] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017. 1

[36] Fabio Pizzati, Pietro Cerri, and Raoul de Charette. Comogan: continuous model-guided image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14288–14298, 2021. 5

[37] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7

[38] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12240–12249, 2019. 5, 7

[39] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, Sep 2018. 6, 8

[40] Ujjwal Saxena and Rohan Giriraj. Automold. `https://github.com/UjjwalSaxena/Automold--Road-Augmentation-Library`. Accessed: 2022-12-20. 5

[41] Minsoo Song, Seokjae Lim, and Wonjun Kim. Monocular depth estimation using laplacian pyramid-based depth residuals. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(11):4381–4393, 2021. 2

[42] 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. 3, 5

[43] Maxime Tremblay, Shirsendu Sukanta Halder, Raoul De Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *International Journal of Computer Vision*, 129(2):341–360, 2021. 5

[44] 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. *arXiv preprint arXiv:2206.13850*, 2022. 1, 3, 5

[45] Arnav Varma, Hemang Chawla, Bahram Zonooz, and Elahe Arani. Transformers in self-supervised monocular depth estimation with unknown camera intrinsics. *arXiv preprint arXiv:2202.03131*, 2022. 3, 5

[46] 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. 1, 2, 3, 5, 6, 8

[47] 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. 2

[48] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1164–1174, 2021. 2

[49] Felix Wimbauer, Nan Yang, Lukas Von Stumberg, Niclas Zeller, and Daniel Cremers. Monorec: Semi-supervised dense reconstruction in dynamic environments from a single moving camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6112–6122, 2021. 2

[50] Jiaxing Yan, Hong Zhao, Penghui Bu, and YuSheng Jin. Channel-wise attention-based network for self-supervised

monocular depth estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 464–473. IEEE, 2021. 7, 8

[51] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 899–908, 2019. 6, 8

[52] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation, 2022. 2

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

[54] 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. *arXiv preprint arXiv:2208.03543*, 2022. 3, 7, 8

[55] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020. 1

[56] Hang Zhou, David Greenwood, and Sarah Taylor. Self-supervised monocular depth estimation with internal feature fusion. *arXiv preprint arXiv:2110.09482*, 2021. 2, 7, 8

[57] 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. 1, 2, 3