SeasonDepth: Cross-Season Monocular Depth Prediction Dataset and Benchmark under Multiple Environments

Hanjiang Hu [1] **Jiacheng Zhu** [1] **Zuxin Liu** [1] **Wenhao Ding** [1] **Shuai Wang** [1] **Jairun Wei** [1] **Baoquan Yang** [2]
Zhijian Qiao [2] **Hesheng Wang** [2] **Ding Zhao** [1]

Abstract

Different environments pose a great challenge to the outdoor robust visual perception for long-term autonomous driving and the generalization of learning-based algorithms. Although monocular depth prediction has been well studied, there is little work focusing on the robust depth prediction across different environments, *e.g.* changing illumination and seasons, owing to the lack of such a multi-environment real-world dataset and benchmark. To this end, we introduce the first cross-season monocular depth prediction dataset and benchmark *SeasonDepth* to benchmark the depth prediction performance under different environments. Using representative and recent state-of-the-art open-source supervised, self-supervised and domain adaptation methods from *KITTI* leaderboard with several newly-formulated metrics, the influence of multiple environments on performance and robustness is analyzed qualitatively and quantitatively, validating the challenging problem and giving promising avenues to enhance the robustness to changing environments.

1. Introduction

perception and localization for autonomous driving and mobile robotics has made significant progress due to the boost of deep convolutional neural networks (Eigen et al., 2014; Liu et al., 2015; Laina et al., 2016; Xu et al., 2017) in recent years. However, since the outdoor environmental conditions are changing because of different seasons, weather and daytime (Maddern et al., 2017; Sattler et al., 2018; Liu et al., 2019), the pixel-level appearance is drastically affected, which casts a big challenge for the robust long-term visual perception and localization. Monocular depth prediction plays a critical role in the long-term visual perception and localization (Zhou et al., 2021; Larsson et al., 2019b; Jenicek & Chum, 2019; Hu et al., 2020; Piasco et al., 2021) and is also significant to safe applications such as self-driving cars under different environmental conditions. Although some depth prediction datasets (Cordts et al., 2016; Ranftl et al., 2020; Antequera et al., 2020) include some different environments for diversity, it is still not clear what kind of algorithm is more robust to adverse conditions and how they influence depth prediction performance. Besides, the generalization of learning-based depth prediction methods on different weather and illumination effects is still an open problem. Therefore, building a new dataset and benchmark under multiple environments is needed to study this problem systematically. To the best of our knowledge, we are the first to study the generalization of learning-based depth prediction under changing environments, which is essential and significant to both robust learning algorithms and practical applications like autonomous driving.

The outdoor high-quality dense depth maps are not easy to obtain using LiDAR or laser scanner projection (Geiger et al., 2012; Saxena et al., 2008; Antequera et al., 2020), or stereo matching (Cordts et al., 2016; Xian et al., 2018; 2020), let alone collections under multiple environments. We adopt Structure from Motion (SfM) and Multi-View Stereo (MVS) pipeline with RANSAC followed by careful manual post-processing to build a scaleless dense depth prediction dataset *SeasonDepth* with multi-environment traverses based on the urban part of CMU Visual Localization dataset (Sattler et al., 2018; Badino et al., 2011). Some examples in the dataset are shown in Fig. 1.

For the benchmark on the proposed dataset, several statistical metrics are proposed for the experimental evaluation of the representative and state-of-the-art open-source methods from *KITTI* benchmark (Geiger et al., 2012; Uhrig et al., 2017). The typical baselines we choose include supervised (Eigen et al., 2014; Lee et al., 2019; Yin et al., 2019; Li & Snavely, 2018), stereo training based self-supervised (Godard et al., 2017; Wong & Soatto, 2019; Tosi et al., 2019), monocular video based self-supervised (Zhou et al., 2017; Guizilini et al., 2020; Godard et al., 2019; Ranjan et al.,

[1]Carnegie Mellon University [2]Shanghai Jiao Tong University. Correspondence to: Hanjiang Hu <hanjianghu@cmu.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Figure 1. SeasonDepth samples with depth maps under *Cloudy + Foliage*, *Low Sun + Foliage*, *Cloudy + Mixed Foliage*, *Overcast + Mixed Foliage* and *Low Sun + Mixed Foliage*.

2019; Klingner et al., 2020) and domain adaptation (Atapour et al., 2018; Zheng et al., 2018; Zhao et al., 2019) algorithms. Through thoroughly analyzing benchmark results, we find that no method can present satisfactory performance in terms of $Average$, $Variance$ and $RelativeRange$ metrics simultaneously even if some methods give impressive results on *KITTI* Eigen split (Eigen et al., 2014) and are well fine-tuned on our training set. We further give hints of promising avenues to addressing this problem through self-supervised learning or stereo geometry constraint for model training. Furthermore, the performance under each environment is investigated both qualitatively and quantitatively for adverse environments.

For the open problem of generalizability of learning-based depth prediction methods on different environmental conditions, our dataset is the first one that contains real-world RGB images with multiple environments under the same routes so that fair cross-environment evaluation can be conducted, giving hints to the future research on robust perception in changing environments. In summary, our contributions in this work are listed as follows.

- A new monocular depth prediction dataset *SeasonDepth* with the same multi-traverse routes under changing environments is introduced through SfM and MVS pipeline and is publicly available.
- We benchmark representative best open-sourced supervised, self-supervised, and domain adaptation depth prediction methods from *KITTI* leaderboard on *SeasonDepth* using several statistical metrics.
- From the extensive cross-environment evaluation, we point out that which kind of methods are robust to different environments and how changing environments affect the depth prediction to give future research directions.

2. Related Work

2.1. Monocular Depth Prediction Datasets

Depth prediction plays an important role in the perception and localization of autonomous driving and other computer vision applications. Many indoor datasets are built through calibrated RGBD camera (Silberman et al., 2012; Kim et al., 2018; Koch et al., 2018), expensive laser scanner (Saxena et al., 2008; Vasiljevic et al., 2019) and web stereo photos (Wang et al., 2019; Xian et al., 2018; 2020; Ranftl et al., 2020). However, outdoor depth maps as ground truth are more complex to get, *e.g.* projecting 3D point cloud data onto the image plane (Geiger et al., 2012; Saxena et al., 2008; Antequera et al., 2020) for sparse map and using stereo matching to calculate inaccurate and limited-scope depth (Cordts et al., 2016; Ranftl et al., 2020; Xian et al., 2018). Another way to get the depth map is through SfM (Chen et al., 2016; Li & Snavely, 2018; Chen et al., 2020; Antequera et al., 2020) from monocular sequences. Although this method is time-consuming, it generates pretty accurate relatively-scaled dense depth maps , which is more general for depth prediction under different scenarios. For the long-term robust perception under changing environments, though some real-world datasets (Cordts et al., 2016; Antequera et al., 2020; Ranftl et al., 2020) include some environmental changes, there are still no multi-environment traverses with the same routes, which is essential and necessary for fair evaluation of robustness across different environments. Since graphical rendering is becoming more and more realistic, some virtual synthetic datasets (Gaidon et al., 2016; Ros et al., 2016; Wang et al., 2020; Miralles, 2017) contain multi-environment traverses though the rendered RGB images are still different from real-world ones due to the domain gap and cannot be used to benchmark real-world cross-environment performance. The details of comparison between datasets are shown in Tab. 1 and Sec. 3.2.

Table 1. Comparison between *SeasonDepth* and Other Datasets

Name	Scene	Real or Virtual	Depth Value	Sparse or Dense	Multiple Traverses	Different Environments
NYUV2 (Silberman et al., 2012)	Indoor	Real	Absolute	Dense	×	×
DIML (Kim et al., 2018)	Indoor	Real	Absolute	Dense	×	×
iBims-1 (Koch et al., 2018)	Indoor	Real	Absolute	Dense	×	×
Make3D (Saxena et al., 2008)	Outdoor & Indoor	Real	Absolute	Sparse	×	×
ReDWeb (Xian et al., 2018)	Outdoor & Indoor	Real	Relative	Dense	×	×
WSVD (Wang et al., 2019)	Outdoor & Indoor	Real	Relative	Dense	×	×
HR-WSI (Xian et al., 2020)	Outdoor & Indoor	Real	Absolute	Dense	×	×
DIODE (Vasiljevic et al., 2019)	Outdoor & Indoor	Real	Absolute	Dense	×	×
OASIS (Chen et al., 2020)	Outdoor & Indoor	Real	Relative	Dense	×	×
3D Movies (Ranftl et al., 2020)	Outdoor & Indoor	Real	Relative	Dense	×	✓
KITTI (Geiger et al., 2012)	Outdoor	Real	Absolute	Sparse	×	×
Cityscapes (Cordts et al., 2016)	Outdoor	Real	Absolute	Dense	×	✓
DIW (Chen et al., 2016)	Outdoor	Real	Relative	Sparse	×	×
MegaDepth (Li & Snavely, 2018)	Outdoor	Real	Relative	Dense	×	×
DDAD (Guizilini et al., 2020)	Outdoor	Real	Absolute	Dense	×	×
MPSD (Antequera et al., 2020)	Outdoor	Real	Absolute	Dense	×	✓
V-KITTI (Gaidon et al., 2016)	Outdoor	Virtual	Absolute	Dense	✓	✓
SYNTHIA (Ros et al., 2016)	Outdoor	Virtual	Absolute	Dense	×	×
TartanAir (Wang et al., 2020)	Outdoor & Indoor	Virtual	Absolute	Dense	✓	✓
DeepGTAV (Miralles, 2017)	Outdoor	Virtual	Absolute	Dense	✓	✓
SeasonDepth	**Outdoor**	**Real**	**Relative**	**Dense**	✓	✓

2.2. Outdoor Monocular Depth Prediction Algorithms

The monocular depth prediction task aims to predict the dense depth map in an active way given one single RGB image. Early studies including MRF and other graph models (Saxena et al., 2006; 2008; Liu et al., 2010) largely depend on man-made descriptors, constraining the performance of depth prediction. Afterwards, studies based on CNNs (Eigen et al., 2014; Eigen & Fergus, 2015; Laina et al., 2016) have shown promising results for monocular depth estimation. Eigen *et al.* (Eigen et al., 2014) first predict depth map using CNN model, while (Laina et al., 2016) introduces fully convolutional neural networks to regress the depth value. After that, supervised methods for monocular depth prediction have been well studied through normal estimation (Yin et al., 2019; Kusupati et al., 2020), the supervision of depth map and stereo disparity ground truth (Li & Snavely, 2018; Fu et al., 2018; Lee et al., 2019; Xian et al., 2020; Qiao et al., 2021). However, since outdoor depth map ground truth is expensive and time-consuming to obtain, self-supervised depth estimation methods have appeared using stereo geometric left-right consistency (Garg et al., 2016; Godard et al., 2017; Luo et al., 2018; Wong & Soatto, 2019; Tosi et al., 2019; GonzalezBello & Kim, 2020), egomotion-pose constraint through monocular video (Zhou et al., 2017; Mahjourian et al., 2018; Casser et al., 2019; Guizilini et al., 2020; Godard et al., 2019) and multi-task learning with optical flow, motion and semantics segmentation (Yin & Shi, 2018; Zou et al., 2018; Ranjan et al., 2019; Klingner et al., 2020) inside monocular video training pipeline as secondary supervisory signals. Besides, to avoid using expensive real-world depth map ground truth, other algorithms are trained on synthetic virtual datasets (Gaidon et al., 2016; Ros et al., 2016; Wang et al., 2020; Miralles, 2017) to leverage high-quality depth map ground truth with zero cost. Such methods (Zheng et al., 2018; Atapour et al., 2018; Chen et al., 2019; Zhao et al., 2019; Bozorgtabar et al., 2019) confront with the domain adaptation from synthetic to real-world domain only with supervision on virtual datasets for model training.

3. SeasonDepth Dataset

Our proposed dataset *SeasonDepth* is derived from CMU Visual Localization dataset (Badino et al., 2011) through SfM algorithm. The original CMU Visual Localization dataset covers over one year in Pittsburgh, USA, including 12 different environmental conditions. Images were collected from two identical cameras on the left and right of the vehicle along a route of 8.5 kilometers. And this dataset is also derived for long-term visual localization (Sattler et al., 2018) by calculating the 6-DoF camera pose of images with more appropriate categories about the weather, vegetation and area. To be consistent with the content of driving scenes in other datasets like *KITTI*, we adopt images from Urban area categorized in (Sattler et al., 2018) to build our dataset. More details about the dataset can be found in Appendix Sec. A.1.

3.1. Depth Dense Reconstruction and Post-processing

We reconstruct the dense model for each traversal under every environmental condition through SfM and MVS pipeline (Schönberger et al., 2016), which is commonly used for depth reconstruction (Guizilini et al., 2020; Li & Snavely,

2018) and most suitable for multi-environment dense reconstruction for 3D mapping (Larsson et al., 2019a; Sattler et al., 2018) and show advantage on the aspects of high dense quality despite of huge computational efforts compared to active sensing from LiDAR. Specifically, similar to *MegaDepth* (Li & Snavely, 2018), COLMAP (Schonberger & Frahm, 2016; Schönberger et al., 2016) with SIFT descriptor (Lowe, 2004) is used to obtain the depth maps through photometric and geometric consistency from sequential images.

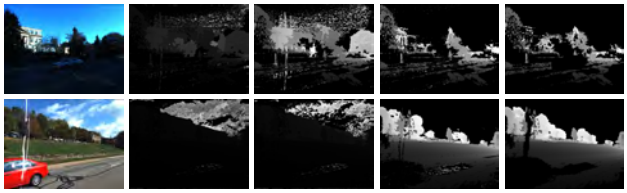


Figure 2. The illustration of depth map processing.

Furthermore, we adopt RANSAC algorithm in the SfM to remove the inaccurate values of dynamic objects in the images through effective modification in SIFT matching triangulation based on original COLMAP, where dynamic objects with additional motion besides relative camera motion do not obey the multi-view geometry constraint and should be removed as noise via RANSAC in bundle adjustment optimization. Besides, from our justification experiments in Sec. B.3, it is validated that using relative depth values and removing dynamic noise will not significantly influence the training and the performance of depth prediction models. Because the MVS algorithm generates the depth maps with error pixel values which are out of range or too close, like the cloud in the sky or noisy points on the very near road, we filter those outside the normal range of the depth map.

After the reconstruction, based on the observation of noise distribution in the HSV color space, *e.g.* blue pixels always appear in the sky and dark pixels always appear in the shade of low sun, which tend to be noise in most cases, we remove the noisy values in the HSV color space given some specific thresholds. Though outliers are set to be empty in RANSAC, instance segmentation is adopted through MaskRCNN (He et al., 2017) to fully remove the noise of dynamic objects. However, since it is difficult to generate accurate segmentation maps only for dynamic objects under drastically changing environments, we leverage human annotation as the last step to finally check the depth map. The data processing is shown in Fig.2 with normalization after each step. Since we are rigorous and serious to the quality of valid depth pixels which are used for benchmark, we set most noise to be invalid (which causes some "holes" on the boundary from appearance) to avoid any possible pollution to the following benchmark, ensuring the reliable evaluation and benchmark results. More details can be found in Appendix Sec. A.1.

3.2. Comparison with Other Datasets

The current datasets are introduced in Sec. 2.1. The comparison between *SeasonDepth* and current datasets is shown in Tab. 1. The distinctive feature of the proposed dataset is that *SeasonDepth* contains comprehensive outdoor real-world multi-environment sequences with repeated scenes, just like virtual synthetic datasets (Gaidon et al., 2016; Miralles, 2017; Wang et al., 2020) but they are rendered from computer graphics and suffer from the huge domain gap. Though real-world datasets (Antequera et al., 2020; Ranftl et al., 2020; Cordts et al., 2016) include different environments, they lack the same-route traverses under different conditions, so they are unable to fairly evaluate the performance across changing environments. Similar to outdoor datasets (Chen et al., 2016; Li & Snavely, 2018; Chen et al., 2020), the depth maps of ours are scaleless with relative depth values, where the metrics should be designed for evaluation, as the following section shows. The depth map ground truth from SfM is dense compared to LiDAR-based sparse depth maps. Besides, the comparison of depth value distribution is shown in Fig. 3. Note that the values of our dataset are scaleless and relative, so the x-axes of other datasets are also omitted for a fair comparison. We normalize the depth values for all the environments to mitigate the influence of the aggregation from relative depth distributions under different environments to get the final distribution map. The details of implementation can be found in Appendix Sec. A.2. From Fig. 3, it can be seen that our dataset also follows the long-tail distribution (Jiao et al., 2018) which is the same as other datasets, with a difference of missing large-depth part due to range truncation during the building process in Sec. 3.1.

4. Benchmark Setup

4.1. Evaluation Metrics

The challenge for the design of evaluation metrics lies in two folds. One is to cope with scaleless and partially valid dense depth map ground truth, and the other is to fully measure the depth prediction average performance and the stability or robustness across different environments. Due to scaleless ground truth of relative depth value, some common metrics (Uhrig et al., 2017) cannot be used for evaluation directly. Since focal lengths of two cameras are close enough to generate similarly distributed depth values, unlike (Zhou et al., 2017; Li & Snavely, 2018; Chen et al., 2020), we align the distribution of depth prediction to depth ground truth via mean value and variance for a fair evaluation. The other key point for multi-environment evaluation lies in the reflection of robustness to changing environments for same-route sequences, which has not been studied in the

Figure 3. Comparison of relative depth distributions of several datasets.

previous work to the best of our knowledge. We formulate our metrics below.

Figure 4. The examples of depth adjustment (from the first to second row) for prediction results.

First, for each pair of predicted and ground truth depth maps, the valid pixels $D^{i,j}_{valid_{predicted}}$ of the predicted depth map $D_{valid_{predicted}}$ are determined by non-empty valid pixels $D^{i,j}_{valid_{GT}}$ of the depth map ground truth. And then the valid mean and variance of both $D_{valid_{GT}}$ and $D_{valid_{predicted}}$ are calculated as Avg_{GT},Avg_{pre} and Var_{GT},Var_{pre}. Then we adjust the predicted depth map D_{adj} to get the same distribution with $D_{valid_{GT}}$,

$$D_{adj} = (D_{pre} - Avg_{pre}) \times \sqrt{Var_{GT}/Var_{pre}} + Avg_{GT}$$

The examples of adjusted depth prediction are shown in Fig. 4. After this operation, we can eliminate scale difference for depth prediction across datasets, which makes this zero-shot evaluation on *SeasonDepth* reliable and applicable to all the models even though they predict absolute depth values, showing generalization ability on new datasets and robustness across different environments. Denote the adjusted valid depth prediction D_{adj} as D_P in the following formulation. To measure the depth prediction performance, we choose the most distinguishable metrics under multiple environments from commonly-used metrics in (Uhrig et al., 2017), *AbsRel* and $\delta < 1.25$ (a_1). For environment k, we have,

$$AbsRel^k = \frac{1}{n}\sum_{i,j}^{n} \left| {D_P}^k{}_{i,j} - {D_{GT}}^k{}_{i,j} \right| / {D_{GT}}^k{}_{i,j}$$

$$a_1^k = \frac{1}{n}\sum_{i,j}^{n} \mathbb{1}(max\{\frac{{D_P}^k{}_{i,j}}{{D_{GT}}^k{}_{i,j}}, \frac{{D_{GT}}^k{}_{i,j}}{{D_P}^k{}_{i,j}}\} < 1.25)$$

For the evaluation under different environments, six secondary metrics are derived based on original metrics and statistics,

$$AbsRel^{avg} = \frac{1}{m}\sum_k AbsRel^k$$

$$AbsRel^{var} = \frac{1}{m}\sum_k \left| AbsRel^k - \frac{1}{m}\sum_k AbsRel^k \right|^2$$

$$a_1^{avg} = \frac{1}{m}\sum_k a_1^k$$

$$a_1^{var} = \frac{1}{m}\sum_k \left| a_1^k - \frac{1}{m}\sum_k a_1^k \right|^2$$

where avg terms $AbsRel^{avg}$, a_1^{avg} and var terms $AbsRel^{var}$, a_1^{var} come from *Mean* and *Variance* in statistics, indicating the average performance and the fluctuation around the mean value across multiple environments.

Considering the depth prediction applications, it should be more rigorous to prevent better results fluctuation than worse results under changing conditions. Therefore, we use the *Relative Range* terms $AbsRel^{relRng}$, a_1^{relRng} to calculate the relative difference of maximum and minimum for all the environments.

$$AbsRel^{relRng} = \frac{\max\{AbsRel^k\} - \min\{AbsRel^k\}}{\frac{1}{m}\sum_k AbsRel^k}$$

$$a_1^{relRng} = \frac{\max\{1 - a_1^k\} - \min\{1 - a_1^k\}}{\frac{1}{m}\sum_k (1 - a_1^k)}$$

Relative Range terms for *AbsRel* and $1 - a_1$ are more strict than the *Variance* terms $AbsRel^{var}$, a_1^{var} and note that $1 - a_1$ instead of a_1 is used to calculate a_1^{relRng} to make relative range fluctuation more distinguishable for better methods.

4.2. Evaluated Algorithms

Following the category introduced in Sec. 2.2, we have chosen the representative baseline methods together with recent open-source state-of-the-art models on *KITTI* leaderboard (Uhrig et al., 2017) to evaluate the performance on

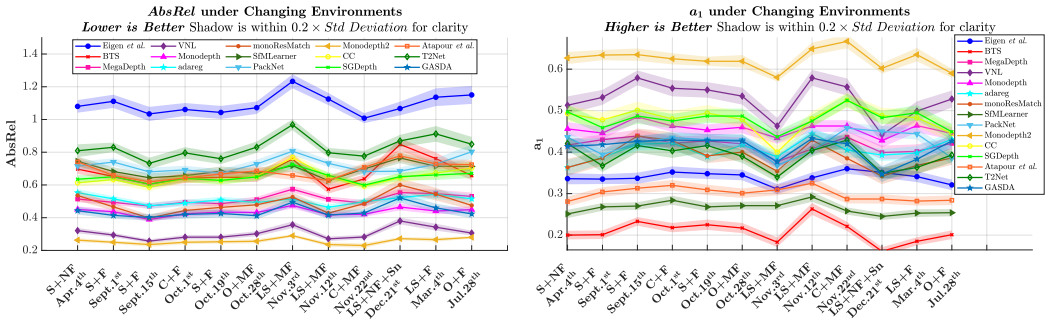


Figure 5. Results on *SeasonDepth* dataset under 12 different environments with dates. The shadows indicate error bars around mean values with $0.2 \times Standard\ Deviation$ for more clarity.

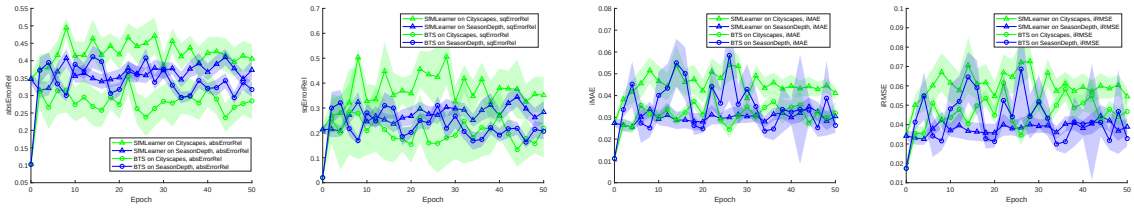


Figure 6. Cross-dataset quantitative performance evolution on KITTI validation set (Uhrig et al., 2017) with models fine-tuned on *SeasonDepth* and *Cityscapes* (Cordts et al., 2016).

the *SeasonDepth* dataset. The evaluated methods include supervised and self-supervised models trained on real-world images, and domain adaptation models trained on virtual synthetic images. For the **supervised models**, we choose Eigen *et al.* (Eigen et al., 2014), *BTS* (Lee et al., 2019), *MegaDepth* (Li & Snavely, 2018) and *VNL* (Yin et al., 2019). Eigen *et al.* propose the first method using CNNs to predict depth map with scale-invariant loss. *BTS* proposes novel multi-scale local planar guidance layers in decoders for full spatial resolution to get impressive ranked-4th performance. *MegaDepth* introduces an end-to-end hourglass network for depth prediction using semantic and geometric information as supervision. *VNL* proposes the virtual normal estimation, which utilizes a stable geometric constraint for long-range relations in a global view to predict depth.

We further choose **self-supervised models** of stereo training, monocular video training and multi-task learning as secondary signals with video training. Previous work *Monodepth* (Godard et al., 2017) and two recent work *adareg* (Wong & Soatto, 2019), *monoResMatch* (Tosi et al., 2019) are evaluated to present the performance of models trained with stereo geometric constraint. For joint pose regression and depth prediction using video sequences, we test the first method *SfMLearner* (Zhou et al., 2017) and two recent methods *Monodepth2* (Godard et al., 2019), *PackNet* (Guizilini et al., 2020), where *Monodepth2* model also involves stereo geometric information in model training. Besides, we evaluate *CC* (Ranjan et al., 2019) with optical flow estimation and motion segmentation, and *SGDepth* (Klingner et al., 2020) with supervised semantic segmentation inside the monocular video based self-supervised framework. For **domain adaptation models** trained on the virtual dataset with multiple environments, we evaluate several recent competitive algorithms Atapour *et al.* (Atapour et al., 2018), *T2Net* (Zheng et al., 2018) and *GASDA* (Zhao et al., 2019). Atapour *et al.* (Atapour et al., 2018) use CycleGAN (Zhu et al., 2017) to train depth predictor with translated synthetic images using virtual ground truth from DeepGTAV (Miralles, 2017). *T2Net* is a fully supervised method both on *KITTI* and *V-KITTI* dataset, and it enables synthetic-to-real translation and depth prediction simultaneously. But *GASDA* is self-supervised for real-world images by incorporating geometry-aware loss through wrapping stereo images together with image translation from synthetic to the real-world domain. More details about the benchmark models, including fine-tuning details, can be found in Appendix Sec. B.1.

5. Experimental Evaluation Results

5.1. Evaluation Comparison from Overall Metrics

In this section, we analyze and discuss what kinds of algorithms are more robust to changing environments by giving several main findings and their impacts on performance.

Table 2. SeasonDepth Benchmark Results (↓: Lower Better, ↑: Higher Better, **Best**, Second Best)

	Method	***KITTI* Eigen Split**		***SeasonDepth*: Average**		**Variance**(10^{-2})		**Relative Range**	
		AbsRel ↓	a_1 ↑	*AbsRel* ↓	a_1 ↑	*AbsRel* ↓	a_1 ↓	*AbsRel* ↓	$1-a_1$ ↓
Supervised	Eigen *et al.* (Eigen et al., 2014)	0.203	0.702	1.093	0.340	0.346	**0.0170**	0.206	0.0746
	BTS (Lee et al., 2019)	**0.060**	**0.955**	0.676	0.209	0.545	0.0650	0.405	0.129
	BTS (fine-tuned)	—	—	0.339	0.479	0.0425	0.0389	0.203	0.117
	MegaDepth (Li & Snavely, 2018)	0.220	0.632	0.515	0.417	0.0874	0.0285	0.200	0.107
	VNL (Yin et al., 2019)	0.072	0.938	0.306	0.527	0.126	0.166	0.400	0.290
Self-supervised Stereo Training	Monodepth (Godard et al., 2017)	0.148	0.803	0.436	0.455	0.0475	0.0213	0.198	0.104
	adareg (Wong & Soatto, 2019)	0.126	0.840	0.507	0.405	0.0630	0.0474	0.178	**0.0137**
	monoResMatch (Tosi et al., 2019)	0.096	0.890	0.487	0.389	0.286	0.0871	0.414	0.160
Self-supervised Monocular Video Training	SfMLearner (Zhou et al., 2017)	0.181	0.733	0.360	0.495	0.0801	0.0628	0.269	0.182
	SfMLearner (fine-tuned)	—	—	0.413	0.440	0.0502	0.0290	**0.177**	0.100
	PackNet (Guizilini et al., 2020)	0.116	0.865	0.722	0.421	0.187	0.0705	0.186	0.155
	Monodepth2 (Godard et al., 2019)	0.106	0.874	**0.256**	**0.624**	**0.0311**	0.0532	0.235	0.229
	CC (Ranjan et al., 2019)	0.140	0.826	0.648	0.479	0.223	0.0881	0.280	0.241
	SGDepth (Klingner et al., 2020)	0.113	0.879	0.648	0.480	0.0987	0.0498	0.197	0.169
Syn-to-real Domain Adaptation	Atapour *et al.* (Atapour et al., 2018)	0.110	0.923	0.687	0.300	0.224	0.0220	0.231	0.0622
	T2Net (Zheng et al., 2018)	0.169	0.769	0.827	0.391	0.399	0.0799	0.286	0.146
	GASDA (Zhao et al., 2019)	0.143	0.836	0.438	0.411	0.121	0.0665	0.271	0.145

The quantitative results of open-source best depth prediction baselines can be found in Tab. 2. To alleviate the impact of dataset bias between *KITTI* and *SeasonDepth*, we adopt the held-out training set to fine-tune one supervised (Lee et al., 2019) and one self-supervised model (Zhou et al., 2017), which initially perform poor zero-shot results. Since our dataset does not contain stereo images, segmentation ground truth, and KITTI-like scenarios, just like in V-KITTI, the stereo training based, semantic segmentation involved multi-task training and domain adaptation models are omitted for the sake of fairness.

Analysis of fine-tuning To make sure the findings and claims are predominantly owing to the different conditions instead of domain shift, the fine-tuning analysis is first presented before other critical findings for this problem. The best fine-tuned results of $Average$ are chosen in Tab. 2 together with the corresponding $Variance$ and $RelativeRange$ results. Consequently, fine-tuning gives limited help to the robustness to changing environments though overall performance is better because of reducing the domain gap, indicating that solely increasing the variability of training data cannot address the challenge of environmental changes. To make the evaluation and comparison fair, we draw our conclusion regardless they are fine-tuned or not. $Variance$ and $RelativeRange$ metrics are convincing to reflect robustness across different environments.

Supervised *vs* self-supervised methods Basically, self-supervised methods show more robustness to different environments than supervised ones. Supervised methods suffer from large values of $Variance$ and $RelativeRange$ across multiple environments compared to self-supervised methods, showing that supervised methods are more sensitive to changing environments and even fine-tuning cannot critically improve the cross-environment performance. Besides, although the first proposed several depth prediction methods (Eigen et al., 2014; Godard et al., 2017; Zhou et al., 2017; Atapour et al., 2018) perform worse than recent methods regarding overall $Average$, they show impressive stability to different environments through low $Variance$ and $RelativeRange$.

Effectiveness of stereo training Inside the self-supervised methods, stereo training based methods (Godard et al., 2017; Wong & Soatto, 2019; Tosi et al., 2019) are more robust to different environments than monocular video training based (Zhou et al., 2017; Guizilini et al., 2020) and multi-task learning (Ranjan et al., 2019; Klingner et al., 2020) methods from $Variance$ and $RelativeRange$. Coming broadly to monocular video training and syn-to-real models, training with stereo geometry constraint (Godard et al., 2019; Zhao et al., 2019) is clearly beneficial to improve the robustness to the changing environments compared to those without it, as shown quantitatively with light blue shades in Tab. 2 and qualitatively with underlines in Fig. 7. Interestingly, the methods with good $Variance$ performance are not consistent with those with good $Average$ performance, which indicates that algorithms tend to work well in specific environments instead of being effective and robust to all conditions, validating the significance of the cross-environment study with *SeasonDepth* dataset.

5.2. Performance under Different Environment Conditions

In this section, we further study how different environments influence the depth prediction results. Different from how different methods perform under multiple environments, this section investigates which environment is more difficult for the current depth prediction models The line chart with shadow error bar in Fig. 5 shows performance in changing environments intuitively. The abbreviations of environments are *S* for *Sunny*, *C* for *Cloudy*, *O* for *Overcast*, *LS* for *Low Sun*, *Sn* for *Snow*, *F* for *Foliage*, *NF* for *No Foliage*, and *MF* for *Mixed Foliage*.

Figure 7. Comparison among supervised, self-supervised stereo based (S-Sup-S), self-supervised monocular video based (S-Sup-M) and domain adaptation (Syn-to-Real) methods. Underlines are denoted for training with stereo geometry constraint.

Figure 8. Qualitative comparison results on KITTI validation set (Uhrig et al., 2017) with depth prediction models fine-tuned on *SeasonDepth* and *Cityscapes* (Cordts et al., 2016)

Influence of different environments From Fig. 5, we can see that although different methods perform differently on *AbsRel* and a_1, the influence of some environments is similar for all the methods. Most methods perform well under *S+F, Sept.* 15^{th} and *LS+MF, Nov.* 12^{th} while dusk scenes in *LS+MF, Nov.* 3^{rd} and snowy scenes in *LS+NF+Sn, Dec.* 21^{st} pose great challenge for most algorithms, which points out directions for future research and safe applications. Besides, the error bar in Fig. 5 shows adverse environments always result in large deviations for all algorithms, indicating adverse environments influence the results of all the methods.

Promising methods for adverse environments Under these adverse environmental conditions, the promising algorithms can also be found. For the dusk or snowy scenes, although training with virtual synthetic images with multiple environments through domain adaptation (Atapour et al., 2018; Zheng et al., 2018; Zhao et al., 2019) helps little for the overall metrics, some of the domain adaptation methods (Atapour et al., 2018; Zheng et al., 2018) present impressive robustness under adverse scenes due to the various appearances of synthetic images. For the snowy scenes, self-supervised stereo-based (Wong & Soatto, 2019; Godard et al., 2017; 2019) and monocular video training models (Ranjan et al., 2019; Klingner et al., 2020; Guizilini et al., 2020) are less influenced compared to supervised methods.

Qualitative analysis Qualitative experimental results in Fig. 12 show how extreme illumination or vegetation changes affect the depth prediction. We visualize the adjusted results of three overall good methods with robustness to changing environments according to Sec. 5.1 and Tab. 2. From the top two rows, it can be seen that illumination change of low sun makes the depth prediction of tree trunks less clear under the same vegetation condition as green and red blocks show. Also, no foliage tends to make telephone pole and tree trunk less distinguishable by comparing red and green blocks from the last two rows, while the depth prediction of heavy vegetation is difficult as red blocks show on the fourth row given the same illumination and weather condition. More qualitative and detailed results with mean values and standard deviations can be found in Appendix Sec. B.2.

6. Conclusion

In this paper, a new dataset *SeasonDepth* is built for monocular depth prediction under different environments, and supervised, self-supervised, and domain adaptation open-source algorithms from *KITTI* leaderboard are evaluated. From the experimental results, we find that there is still a long way

to go to achieve robustness for long-term depth prediction, and several promising avenues are given. Self-supervised methods present better robustness than supervised methods to changing environments, and stereo geometry involved training also helps under cross-environment cases. Through studying how adverse environments influence, our findings via this dataset and benchmark will impact the research on long-term robust perception and related applications.

References

Antequera, M. L., Gargallo, P., Hofinger, M., Bulò, S. R., Kuang, Y., and Kontschieder, P. Mapillary planet-scale depth dataset. In *European Conference on Computer Vision*, pp. 589–604. Springer, 2020.

Atapour, A. A., Breckon, T. P., and et al. 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*, pp. 2800–2810, 2018.

Badino, H., Huber, D., and Kanade, T. Visual topometric localization. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 794–799. IEEE, 2011.

Benbihi, A., Geist, M., and Pradalier, C. Image-based place recognition on bucolic environment across seasons from semantic edge description. 2020.

Bozorgtabar, B., Rad, M. S., Mahapatra, D., and Thiran, J.-P. Syndemo: Synergistic deep feature alignment for joint learning of depth and ego-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4210–4219, 2019.

Casser, V., Pirk, S., Mahjourian, R., and Angelova, A. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8001–8008, 2019.

Chen, W., Fu, Z., Yang, D., and Deng, J. Single-image depth perception in the wild. In *Advances in neural information processing systems*, pp. 730–738, 2016.

Chen, W., Qian, S., Fan, D., Kojima, N., Hamilton, M., and Deng, J. Oasis: A large-scale dataset for single image 3d in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 679–688, 2020.

Chen, Y., Li, W., Chen, X., and Gool, L. V. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1841–1850, 2019.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Eigen, D. and Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pp. 2650–2658, 2015.

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

Fu, H., Gong, M., Wang, C., Batmanghelich, K., and Tao, D. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2002–2011, 2018.

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

Garg, R., BG, V. K., Carneiro, G., and Reid, I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pp. 740–756. Springer, 2016.

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

Godard, C., Mac Aodha, O., and Brostow, G. J. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270–279, 2017.

Godard, C., Aodha, O. M., Firman, M., and Brostow, G. J. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3828–3838, 2019.

GonzalezBello, J. L. and Kim, M. Forget about the lidar: Self-supervised depth estimators with med probability volumes. *Advances in Neural Information Processing Systems*, 33, 2020.

Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., and Gaidon, A. 3d packing for self-supervised monocular

depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2485–2494, 2020.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Hu, H., Qiao, Z., Cheng, M., Liu, Z., and Wang, H. Dasgil: Domain adaptation for semantic and geometric-aware image-based localization. *IEEE Transactions on Image Processing*, 30:1342–1353, 2020.

Jenicek, T. and Chum, O. No fear of the dark: Image retrieval under varying illumination conditions. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9696–9704, 2019.

Jiao, J., Cao, Y., Song, Y., and Lau, R. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 53–69, 2018.

Kim, H. J., Dunn, E., and Frahm, J.-M. Learned contextual feature reweighting for image geo-localization. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3251–3260. IEEE, 2017.

Kim, Y., Jung, H., Min, D., and Sohn, K. Deep monocular depth estimation via integration of global and local predictions. *IEEE transactions on Image Processing*, 27(8): 4131–4144, 2018.

Klingner, M., Termöhlen, J.-A., Mikolajczyk, J., and Fingscheidt, T. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *European Conference on Computer Vision*, pp. 582–600. Springer, 2020.

Koch, T., Liebel, L., Fraundorfer, F., and Körner, M. Evaluation of cnn-based single-image depth estimation methods. In Leal-Taixé, L. and Roth, S. (eds.), *European Conference on Computer Vision Workshop (ECCV-WS)*, pp. 331–348. Springer International Publishing, 2018.

Kusupati, U., Cheng, S., Chen, R., and Su, H. Normal assisted stereo depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., and Navab, N. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pp. 239–248. IEEE, 2016.

Larsson, M., Stenborg, E., Hammarstrand, L., Pollefeys, M., Sattler, T., and Kahl, F. A cross-season correspondence dataset for robust semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9532–9542, 2019a.

Larsson, M., Stenborg, E., Toft, C., Hammarstrand, L., Sattler, T., and Kahl, F. Fine-grained segmentation networks: Self-supervised segmentation for improved long-term visual localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 31–41, 2019b.

Lee, J. H., Han, M.-K., Ko, D. W., and Suh, I. H. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019.

Li, Z. and Snavely, N. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2041–2050, 2018.

Liu, B., Gould, S., and Koller, D. Single image depth estimation from predicted semantic labels. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1253–1260. IEEE, 2010.

Liu, F., Shen, C., Lin, G., and Reid, I. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015.

Liu, Z., Zhou, S., Suo, C., Yin, P., Chen, W., Wang, H., Li, H., and Liu, Y.-H. Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2831–2840, 2019.

Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60 (2):91–110, 2004.

Luo, Y., Ren, J., Lin, M., Pang, J., Sun, W., Li, H., and Lin, L. Single view stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 155–163, 2018.

Maddern, W., Pascoe, G., Linegar, C., and Newman, P. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

Mahjourian, R., Wicke, M., and Angelova, A. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5667–5675, 2018.

Miralles, A. R. An open-source development environment for self-driving vehicles. http://openaccess.uoc.edu/webapps/o2/bitstream/10609/63765/6/aruanomTFM0617memory.pdf, 2017.

Piasco, N., Sidibé, D., Gouet-Brunet, V., and Demonceaux, C. Learning scene geometry for visual localization in challenging conditions. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9094–9100. IEEE, 2019.

Piasco, N., Sidibé, D., Gouet-Brunet, V., and Demonceaux, C. Improving image description with auxiliary modality for visual localization in challenging conditions. *International Journal of Computer Vision*, pp. 1–18, 2020.

Piasco, N., Sidibe, D., Gouet-Brunet, V., and Demonceaux, C. Improving image description with auxiliary modality for visual localization in challenging conditions. *International Journal of Computer Vision*, 129(1):185–202, 2021.

Qiao, S., Zhu, Y., Adam, H., Yuille, A., and Chen, L.-C. Vip-deeplab: Learning visual perception with depth-aware video panoptic segmentation. 2021.

Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., and Koltun, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., and Black, M. J. 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*, pp. 12240–12249, 2019.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3234–3243, 2016.

Sattler, T., Maddern, W., Toft, C., Torii, A., Hammarstrand, L., Stenborg, E., Safari, D., Okutomi, M., Pollefeys, M., Sivic, J., et al. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8601–8610, 2018. https://www.visuallocalization.net/.

Saxena, A., Chung, S. H., and Ng, A. Y. Learning depth from single monocular images. In *Advances in neural information processing systems*, pp. 1161–1168, 2006.

Saxena, A., Sun, M., and Ng, A. Y. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5): 824–840, 2008.

Schonberger, J. L. and Frahm, J.-M. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.

Schönberger, J. L., Zheng, E., Frahm, J.-M., and Pollefeys, M. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pp. 501–518. Springer, 2016.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pp. 746–760. Springer, 2012.

Tang, L., Wang, Y., Luo, Q., Ding, X., and Xiong, R. Adversarial feature disentanglement for place recognition across changing appearance. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1301–1307. IEEE, 2020.

Tosi, F., Aleotti, F., Poggi, M., and Mattoccia, S. Learning monocular depth estimation infusing traditional stereo knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9799–9809, 2019.

Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., and Geiger, A. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017.

Vasiljevic, I., Kolkin, N., Zhang, S., Luo, R., Wang, H., Dai, F. Z., Daniele, A. F., Mostajabi, M., Basart, S., Walter, M. R., et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019.

Wang, C., Lucey, S., Perazzi, F., and Wang, O. Web stereo video supervision for depth prediction from dynamic scenes. In *2019 International Conference on 3D Vision (3DV)*, pp. 348–357. IEEE, 2019.

Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., and Scherer, S. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916, 2020.

Wong, A. and Soatto, S. Bilateral cyclic constraint and adaptive regularization for unsupervised monocular depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5644–5653, 2019.

Xian, K., Shen, C., Cao, Z., Lu, H., Xiao, Y., Li, R., and Luo, Z. Monocular relative depth perception with web stereo data supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Xian, K., Zhang, J., Wang, O., Mai, L., Lin, Z., and Cao, Z. Structure-guided ranking loss for single image depth prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 611–620, 2020.

Xin, Z., Cai, Y., Lu, T., Xing, X., Cai, S., Zhang, J., Yang, Y., and Wang, Y. Localizing discriminative visual landmarks for place recognition. In *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

Xu, D., Ricci, E., Ouyang, W., Wang, X., and Sebe, N. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5354–5362, 2017.

Xu, J., Wang, C., Qi, C., Shi, C., and Xiao, B. Unsupervised semantic-based aggregation of deep convolutional features. *IEEE Transactions on Image Processing*, 28(2): 601–611, 2018.

Yin, W., Liu, Y., Shen, C., and Yan, Y. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5684–5693, 2019.

Yin, Z. and Shi, J. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1983–1992, 2018.

Zhao, S., Fu, H., Gong, M., and Tao, D. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9788–9798, 2019.

Zheng, C., Cham, T.-J., and Cai, J. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 767–783, 2018.

Zheng, Z., Wu, Y., Han, X., and Shi, J. Forkgan: Seeing into the rainy night. In *The IEEE European Conference on Computer Vision (ECCV)*, August 2020.

Zhou, H., Ma, J., Tan, C. C., Zhang, Y., and Ling, H. Cross-weather image alignment via latent generative model with intensity consistency. *IEEE Transactions on Image Processing*, 29:5216–5228, 2020.

Zhou, Q., Sattler, T., and Leal-Taixe, L. Patch2pix: Epipolar-guided pixel-level correspondences. 2021.

Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1851–1858, 2017.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

Zou, Y., Luo, Z., and Huang, J.-B. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 36–53, 2018.

A. Building SeasonDepth Dataset

In this section, we present more details about the process of building *SeasonDepth* dataset and statistical analysis of depth maps in each environment.

A.1. Details in Building Dataset

We adopt the categorized slices of the Urban part according to (Sattler et al., 2018) as original images after rectification through camera intrinsic file. Specifically, we use `slice2, slice3, slice7, slice8` as the split test slices for evaluation and benchmark, and the other slices `slice4, slice5, slice6` are intended to treat as training sets. Note that since not all images from the original dataset are appropriate for depth prediction due to huge noise, *e.g.*, a moving truck covering almost all the pixels, we remove such images in the final version. The numbers of images under all the environments for all slices in training set and test set are shown in Tab. 3. The abbreviations of environments are `S` for `Sunny`, `C` for `Cloudy`, `O` for `Overcast`, `LS` for `Low Sun`, `Sn` for `Snow`, `F` for `Foliage`, `NF` for `No Foliage`, and `MF` for `Mixed Foliage`. It could be seen that the total number of test set is larger than that of training set with more different slices, which helps to make the benchmark results more accurate and reliable. Also, the training set can be used to fine-tune pre-trained models, which do not need too many images. Images from left and right cameras are merged together in the same slice for calculation.

We adopt COLMAP's MVS pipeline (Schonberger & Frahm, 2016; Schönberger et al., 2016) to find the 3D structure and depth map. We follow the instruction on `https://colmap.github.io/` with sequential SIFT matching with RANSAC, sparse reconstruction, and dense reconstruction. Some important detailed hyperparameters can be found in Tab. 4, while others are with the default configuration. To make full use of the image sequences, we adjust the sequential matching overlap to be 15 instead of the whole sequence, improving the local optimization with less noise. During each iteration of RANSAC algorithm in triangulation, the minimum inlier ratio for SIFT matching is set to be 0.65 for the consideration that most pixels of a single image are static in most cases. The maximum SIFT matching distance is 0.55 to adapt the distance of dynamic objects and improve efficiency. The image samples after SfM can be found in Fig. 9-(b)

The valid pixels of the original depth map are between the lower threshold and upper threshold to filter most noise pixels. For one thing, since the fields, forests, and cloud in the far distance away from the camera matter little to the depth prediction applications for autonomous driving, we truncate the depth values over 92% (80% in some cases) of the whole image to focus more on the near roads, vehicles, buildings, vegetation, *etc.* For another, due to the camera placement on both sides of the car, the very near descriptors of the road cannot be correctly matched during SfM and reconstructed for dense depth map, which should be removed by filtering the pixel values less than 5% of the whole depth map. Besides, in the special cases that all the near-road noises appear on the bottom of the images, we directly filter the pixels with depth values greater than a threshold in that rectangular bottom area of the images. The samples after depth range truncation can be seen in Fig. 9-(c).

Although depth range truncation removes some pixels with too large depth values, there are still misrecontructed pixels of sky, cloud or shadow with normal depth values. We use `PowerToys` from `https://github.com/microsoft/PowerToys` to pick up typical HSV values for further refinement and denoising. As Tab. 5 shows, the minimal and maximal HSV values are given for some typical noises, including sky, cloud, reflections and shadows. For the clear or cloudy sky, Value tends to be high around 200 and Hue is usually blue or white. However, for those areas in the shadow of low sun, Saturation and Value are extremely low to be about 10% so that the depth map pixels are too hard to be correctly reconstructed, which need to be filtered. The samples after HSV refinement are shown in Fig. 9-(d).

Though RANSAC algorithm inside the SfM and MVS pipeline largely removes pixels of the dynamic objects to ensure the accuracy of overall depth values, the dynamic pixels cannot been fully eliminated and the contours of objects are not clear as well. Therefore, we employ MaskRCNN (He et al., 2017) with pre-trained models from Detectron2 on `https://github.com/facebookresearch/detectron2`. We adopt the pre-trained model with configuration file of *COCO-InstanceSegmentation/mask_rcnn_R50_FPN_3x.yaml* and modify the `MODEL.ROI_HEADS.SCORE_THRESH_TEST` to be `0.5` to find the instance segmentation with the class of `car`, `person` and `bus`. To process the image directly, we modify the visualization part in the official colab notebook, omitting boxes, keypoints and labels and letting $\alpha = 1$ in `draw_polygon` function to set the pixels of the target objects to be black. But semantic or instance segmentation cannot distinguish dynamic objects that need to be removed, we use human annotation to check whether segmented vehicles or pedestrians are moving or not, relabeling the missing dynamic objects and correcting the mislabeled objects. The depth map samples after all the post-processing can be found in Fig. 9-(e). Note that since there are often more mis-reconstructed depth pixels around thin objects like branches and poles, we manually filter some of them in the processing for accuracy and reliable evaluation.

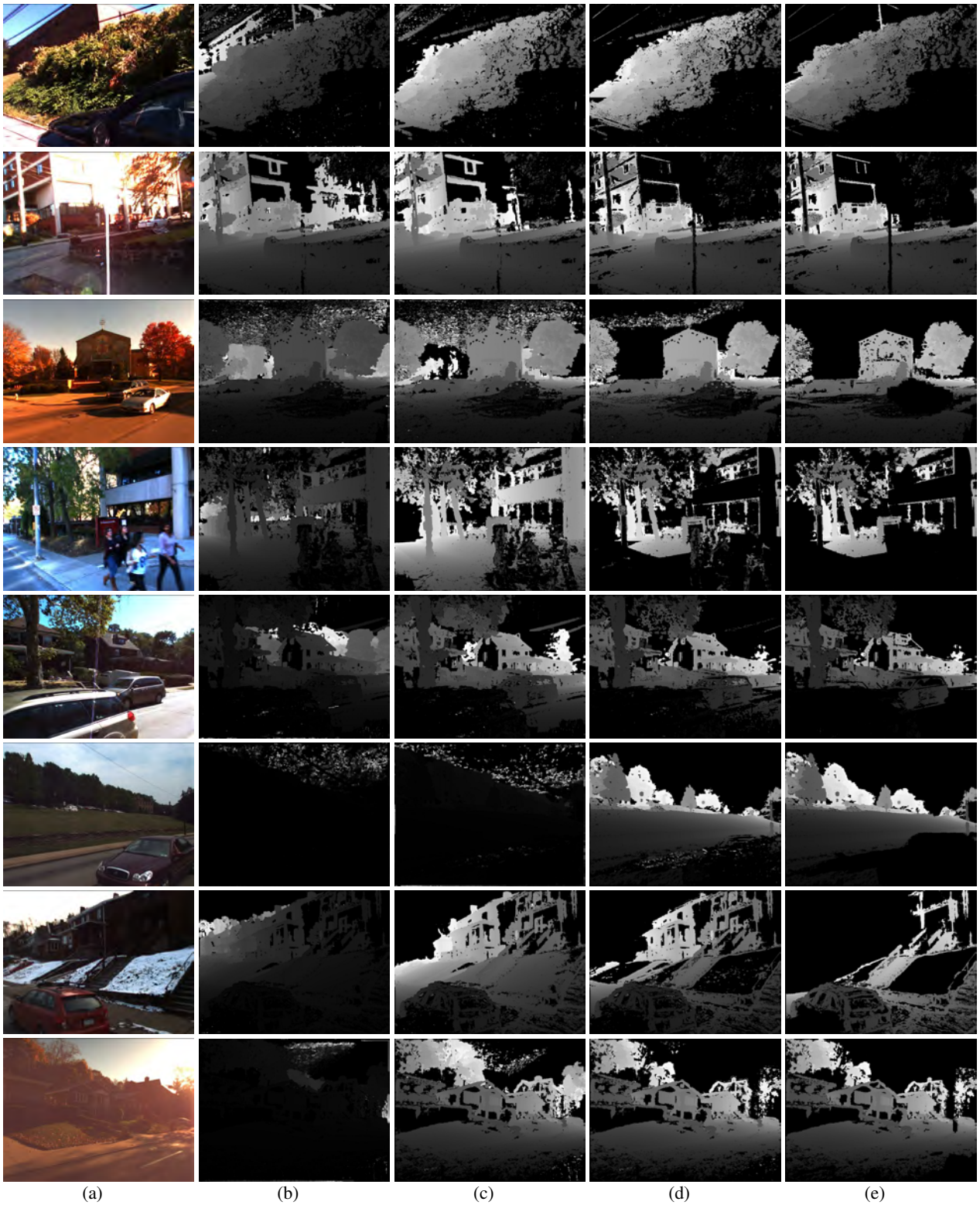


Figure 9. The processing samples given RGB image followed by normalized depth maps for clear visualization of (a) dense reconstruction, (b) range filtering, (c) HSV-based refinement and (d) manual post-processing.

Table 3. Numbers of Images under All the Environments for All Slices

Environments	Training Set				Test Set				
	slice4	slice5	slice6	All Slices	slice2	slice3	slice7	slice8	All Slices
S+NF Apr. 4th	221	129	543	893	382	450	190	449	1471
S+F Sept. 1st	116	230	190	536	385	464	249	490	1588
S+F Sept. 15th	202	213	526	941	335	329	462	457	1583
C+F Oct. 1st	406	205	626	1237	347	438	350	244	1379
S+F Oct. 19th	288	192	558	1038	301	439	412	230	1382
O+MF Oct. 28th	394	194	536	1124	333	418	362	442	1555
LS+MF Nov. 3rd	445	198	399	1042	335	447	203	416	1401
LS+MF Nov. 12th	0	221	552	762	352	500	357	501	1710
C+MF Nov. 22nd	323	163	578	1064	298	436	380	423	1537
LS+NF+Sn Dec. 21st	241	14	592	847	284	512	56	147	999
LS+F Mar. 4th	175	19	498	692	354	222	0	512	1088
O+F Jul. 28th	458	212	560	1230	256	425	384	467	1532
All Environments	3269	1980	6158	11407	3962	5080	3405	4778	17225

A.2. Statistics and Analysis of Depth Map for Each Environment

Here we give the statistical analysis of the proposed *SeasonDepth* dataset for each environment. Since all the depth values are scale-free and not absolute for distance, it is not applicable to directly find the pixel value distribution for the dataset as (Guizilini et al., 2020; Vasiljevic et al., 2019) do. However, the depth values of sequential frames in similar urban scenes under the same environment are similarly distributed, *i.e.* the depth of images along the similar streets and blocks are consistent. Then the key point is to align the distribution of each environment to the mean of all environments, obtaining the normalized whole distribution map and dismissing the scale discrepancy.

Therefore, we first find the original depth value distribution $p_{D_i}(x)$ for all the slices under each environment i. Then lower quartile Q_1 (25%), median Q_2 (50%) and upper quartile Q_3 (75%) are calculated for the original distribution of every environment and the mean value of quartiles can be found as reference quartiles$Q_{1_{ref}}, Q_{2_{ref}}, Q_{3_{ref}}$ for all n environments,

$$Q_{1_{ref}} = \frac{1}{n}\sum_{i=1}^{n} Q_{1_i}, Q_{2_{ref}} = \frac{1}{n}\sum_{i=1}^{n} Q_{2_i}, Q_{3_{ref}} = \frac{1}{n}\sum_{i=1}^{n} Q_{3_i}$$

To find the scale normalization ratio r_i, we use arithmetic mean to measure the ratio of reference quartiles

Table 4. Some Important Hyperparameters for COLMAP

Process	Hyperparameter	Value
Sequential SIFT Matching	min_inlier_ratio	0.65
	max_distance	0.55
	min_num_inliers	50
	overlap_num	15
RANSAC	dyn_num_trials_multiplier	3.0
	confidence	0.99
	min_inlier_ratio	0.1
Sparse Reconstucion	abs_pose_min_inlier_ratio	0.25
	filter_max_reproj_error	4.0
	filter_min_tri_angle	1.5
Dense Reconstucion	geom_consistency_max_cost	3.0
	geom_consistency_regularizer	0.3

Table 5. Some Typical Noises and HSV Thresholds

Noise Source and Type	minimal threshold (H, S, V)	maximal threshold (H, S, V)
Blue Sky	(172, 5%, 40%)	(240, 90%, 100%)
White Cloud and Bright Reflections from Windows	(0, 0%, 100%)	(360, 100%, 100%)
Dark and Black Shadows	(0,0%,0%)	(0,0%,0)%
Dusk Cloud and Refections from Roads and Cars	(0,0%,70%)	(90,20%,100%)
Dusk Sky	(140, 11%, 40%)	(160, 50%, 100%)

$Q_{1_{ref}}, Q_{2_{ref}}, Q_{3_{ref}}$ and other quartiles $Q_{1_i}, Q_{2_i}, Q_{3_i}$,

$$r_i = \frac{1}{3}\left(\frac{Q_{1_{ref}}}{Q_{1_i}} + \frac{Q_{2_{ref}}}{Q_{2_i}} + \frac{Q_{3_{ref}}}{Q_{3_i}}\right) \tag{1}$$

Then the distribution $p_{D_i}(x)$ can be normalized to mean reference environment to obtain $p_{D\_norm_i}(x)$,

$$p_{D\_norm_i}(x) = r_i p_{D_i}(x) \tag{2}$$

After that, the normalized distribution of all the environments can be added directly to get the whole distribution. The distribution map of each environment can be found in Fig. 10. It can be seen that all the pixels follow a similar long-tail distribution, and the average y-axis numbers of per-image pixels overcome the bias caused by unbalanced image quantities across different environments. The normalization makes each distribution aligned on the x-axis, which can be directly added to obtain the total distribution map, as Fig.3 in the main body paper shows.

B. SeasonDepth Benchmark

B.1. Details about Evaluated Models

For the fairness to evaluate the performance of off-the-shelf depth prediction algorithms under changing environments, we investigate a large amount of depth prediction methods and choose to benchmark the representative and recent state-of-the-art supervised, self-supervised, and domain adaptation models from well-known *KITTI* leaderboard (Uhrig et al., 2017), which are with open-source codes and pre-trained models for a fair comparison. Here give important details for all the evaluated baselines. Our experiments are conducted on two NVIDIA 2080Ti cards with 64G RAM on Ubuntu 18.04 system. The evaluation metrics are modified based on `development kit` (Uhrig et al., 2017) on `http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction`.

For the supervised methods, we evaluate four representative methods, Eigen *et al.* (Eigen et al., 2014), *BTS* (Lee et al., 2019), *MegaDepth* (Li & Snavely, 2018) *VNL* (Yin et al., 2019). Eigen *et al.* propose the first CNNs-based

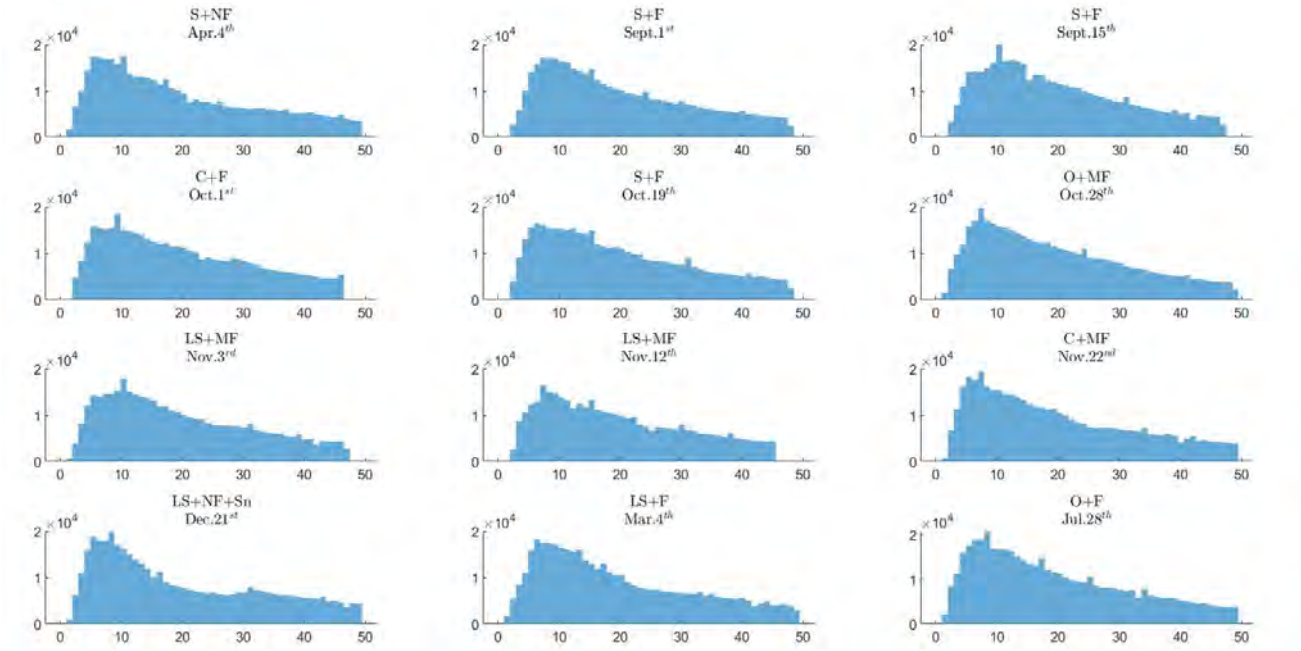


Figure 10. The normalized depth map distribution under all environments. The values of y-axes are number of pixels with the value of abscissa on each image on average.

depth prediction method and introduce the famous Eigen split of *KITTI* dataset for depth prediction benchmark. We hence evaluate this representative method through https://github.com/DhruvJawalkar/Depth-Map-Prediction-from-a-Single-Image-using-a-Multi-Scale-Deep-Network with the improved image gradient component in the newer loss to see the performance across multiple environments. Recent supervised work *BTS* ranks $4th$ on *KITTI* benchmark and we test it on https://github.com/cogaplex-bts/bts using the pre-trained model `DenseNet161` on Eigen split. We further fine-tune this pre-trained model of *BTS* on our training set for 20 epochs with a batch size of 16. The best performance of $Average$ metric is obtained from epoch 20. Due to the scaleless and partially validated ground truth, we only calculate the non-zero pixels and conduct alignment using the mean value for loss when fine-tuning. Note that `focal value` does not influence the experimental results due to the relative scale of the depth metrics. We test *MegaDepth* method according to https://www.cs.cornell.edu/projects/megadepth/ with the `MegaDepth` pre-trained models as described in the paper and all the hyperparameters are set as default. *VNL* are evaluated using https://github.com/YvanYin/VNL_Monocular_Depth_Prediction with the pre-trained model of `ResNext101_32x4d` backbone and trained on *KITTI* dataset.

For self-supervised methods, we further categorize them and choose baselines respectively, *i.e.* *Monodepth* (Godard et al., 2017), *adareg* (Wong & Soatto, 2019) and *monoResMatch* (Tosi et al., 2019) for stereo geometry based methods, *SfMLearner* (Zhou et al., 2017), *Monodepth2* (Godard et al., 2019) and *PackNet* (Guizilini et al., 2020) for monocular video SfM based methods, and *CC* (Ranjan et al., 2019) and *SGDepth* (Klingner et al., 2020) for multi-task learning with monocular SfM unsupervised pipeline. For stereo geometry based unsupervised methods, *Monodepth* method is evaluated using https://github.com/OniroAI/MonoDepth-PyTorch, which is able to reproduce similar results to those in the paper on Eigen split. We test the model of *adareg* from https://github.com/alexklwong/adareg-monodispnet pre-trained with Eigen split. *monoResMatch* is tested through https://github.com/fabiotosi92/monoResMatch-Tensorflow with *KITTI* pretrined model with default hyperparameters.

For sequence SfM based unsupervised methods, we adopt https://github.com/ClementPinard/SfmLearner-Pytorch to benchmark *SfMLearner* for better performance than original repo with slight modification. We further fine-tune the pre-trained models of `dispnet_model_best` and `exp_pose_model_best` on our training set using default configuration file with sequence length of 5

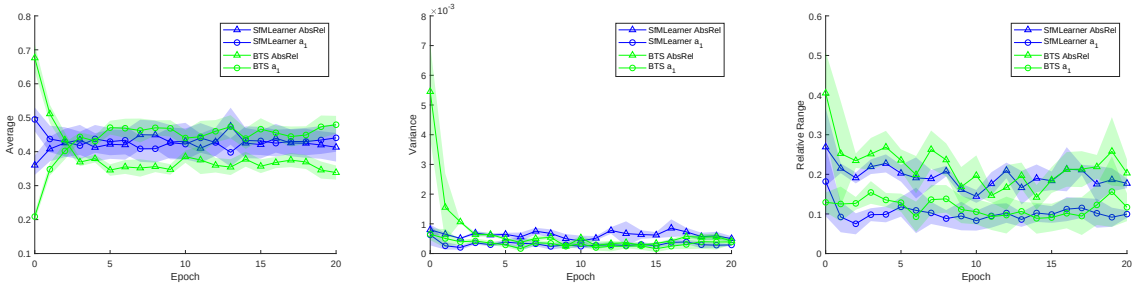


Figure 11. Performance evolution after fine-tuning on *SeasonDepth* training set.

Table 6. ***AbsRel*** Results (**Lower Better**) under Each Environment: Mean(Standard Deviation)

Method	S+NF Apr. 4th	S+F Sept. 1st	S+F Sept. 15th	C+F Oct. 1st	S+F Oct. 19th	O+MF Oct. 28th	LS+MF Nov. 3rd	LS+MF Nov. 12th	C+MF Nov. 22nd	LS+NF+Sn Dec. 21st	LS+F Mar. 4th	O+F Jul. 28th
Eigen *et al.* (Eigen et al., 2014)	1.080(0.39)	1.111(0.40)	1.034(0.43)	1.061(0.40)	1.043(0.40)	1.072(0.38)	1.233(0.43)	1.125(0.37)	1.008(0.32)	1.067(0.42)	1.136(0.54)	1.150(0.55)
BTS (Lee et al., 2019)	0.697(0.29)	0.652(0.24)	0.605(0.24)	0.641(0.29)	0.647(0.27)	0.646(0.28)	0.758(0.35)	0.574(0.27)	0.637(0.27)	0.848(0.36)	0.761(0.38)	0.657(0.28)
MegaDepth (Li & Snavely, 2018)	0.514(0.20)	0.494(0.16)	0.471(0.17)	0.494(0.18)	0.486(0.18)	0.510(0.18)	0.574(0.21)	0.512(0.18)	0.489(0.19)	0.553(0.26)	0.547(0.25)	0.530(0.24)
VNL (Yin et al., 2019)	0.321(0.16)	0.294(0.13)	0.257(0.11)	0.281(0.14)	0.281(0.13)	0.302(0.16)	0.357(0.20)	0.271(0.14)	0.282(0.14)	0.380(0.21)	0.342(0.21)	0.306(0.15)
Monodepth (Godard et al., 2017)	0.450(0.19)	0.437(0.16)	0.389(0.14)	0.424(0.18)	0.434(0.18)	0.432(0.16)	0.475(0.20)	0.418(0.17)	0.421(0.16)	0.465(0.21)	0.441(0.20)	0.449(0.20)
adareg (Wong & Soatto, 2019)	0.553(0.22)	0.515(0.16)	0.473(0.18)	0.489(0.20)	0.509(0.19)	0.493(0.19)	0.515(0.17)	0.463(0.18)	0.498(0.20)	0.523(0.20)	0.543(0.29)	0.515(0.25)
monoResMatch (Tosi et al., 2019)	0.536(0.31)	0.466(0.24)	0.398(0.19)	0.444(0.27)	0.463(0.25)	0.479(0.31)	0.526(0.28)	0.428(0.25)	0.486(0.28)	0.600(0.40)	0.544(0.39)	0.475(0.26)
SfMLearner (Zhou et al., 2017)	0.745(0.29)	0.682(0.26)	0.644(0.27)	0.657(0.28)	0.684(0.29)	0.671(0.28)	0.718(0.35)	0.627(0.27)	0.698(0.27)	0.765(0.32)	0.714(0.29)	0.713(0.31)
PackNet (Guizilini et al., 2020)	0.715(0.27)	0.740(0.23)	0.680(0.26)	0.692(0.26)	0.672(0.24)	0.728(0.27)	0.806(0.27)	0.732(0.22)	0.682(0.25)	0.684(0.22)	0.727(0.36)	0.803(0.43)
Monodepth2 (Godard et al., 2019)	0.263 (0.13)	0.250 (0.10)	0.236 (0.13)	0.250 (0.12)	0.253 (0.11)	0.256 (0.13)	0.290 (0.08)	0.236 (0.13)	0.230 (0.14)	0.272 (0.10)	0.266 (0.12)	0.280 (0.07)
CC (Ranjan et al., 2019)	0.613(0.23)	0.633(0.23)	0.587(0.25)	0.640(0.24)	0.627(0.27)	0.652(0.24)	0.768(0.25)	0.649(0.23)	0.593(0.24)	0.644(0.28)	0.673(0.34)	0.703(0.39)
SGDepth (Klingner et al., 2020)	0.635(0.24)	0.650(0.21)	0.605(0.23)	0.640(0.23)	0.628(0.23)	0.649(0.24)	0.726(0.26)	0.659(0.20)	0.599(0.19)	0.651(0.23)	0.661(0.31)	0.671(0.29)
Atapour *et al.* (Atapour et al., 2018)	0.741(0.27)	0.658(0.22)	0.619(0.24)	0.643(0.27)	0.667(0.27)	0.686(0.29)	0.658(0.28)	0.627(0.29)	0.708(0.27)	0.778(0.32)	0.728(0.29)	0.724(0.30)
T2Net (Zheng et al., 2018)	0.809(0.39)	0.830(0.29)	0.732(0.34)	0.796(0.35)	0.760(0.33)	0.831(0.35)	0.968(0.33)	0.797(0.29)	0.776(0.33)	0.869(0.37)	0.912(0.48)	0.849(0.45)
GASDA (Zhao et al., 2019)	0.443(0.24)	0.414(0.20)	0.402(0.21)	0.420(0.26)	0.426(0.24)	0.412(0.22)	0.495(0.26)	0.416(0.24)	0.429(0.24)	0.521(0.29)	0.460(0.26)	0.423(0.26)

Table 7. $\boldsymbol{a_1}$ Results (**Higher Better**) under Each Environment: Mean(Standard Deviation)

Method	S+NF Apr. 4th	S+F Sept. 1st	S+F Sept. 15th	C+F Oct. 1st	S+F Oct. 19th	O+MF Oct. 28th	LS+MF Nov. 3rd	LS+MF Nov. 12th	C+MF Nov. 22nd	LS+NF+Sn Dec. 21st	LS+F Mar. 4th	O+F Jul. 28th
Eigen *et al.* (Eigen et al., 2014)	0.336(0.14)	0.335(0.12)	0.337(0.14)	0.352(0.14)	0.348(0.13)	0.345(0.13)	0.311(0.12)	0.338(0.13)	0.360(0.12)	0.351(0.13)	0.341(0.13)	0.321(0.13)
BTS (Lee et al., 2019)	0.200(0.11)	0.201(0.10)	0.233(0.10)	0.218(0.11)	0.225(0.12)	0.217(0.12)	0.183(0.12)	0.263(0.15)	0.221(0.11)	0.161(0.10)	0.185(0.10)	0.201(0.11)
MegaDepth (Li & Snavely, 2018)	0.417(0.14)	0.430(0.13)	0.439(0.15)	0.422(0.16)	0.427(0.13)	0.420(0.15)	0.377(0.13)	0.408(0.15)	0.436(0.15)	0.399(0.17)	0.402(0.17)	0.421(0.15)
VNL (Yin et al., 2019)	0.513(0.21)	0.532(0.18)	0.579(0.18)	0.554(0.20)	0.550(0.19)	0.535(0.20)	0.463(0.20)	0.579(0.19)	0.557(0.21)	0.442(0.19)	0.499(0.23)	0.528(0.21)
Monodepth (Godard et al., 2017)	0.456(0.17)	0.446(0.15)	0.485(0.13)	0.463(0.15)	0.453(0.14)	0.460(0.15)	0.434(0.14)	0.463(0.14)	0.463(0.14)	0.428(0.17)	0.464(0.16)	0.445(0.15)
adareg (Wong & Soatto, 2019)	0.363(0.18)	0.387(0.14)	0.419(0.15)	0.422(0.17)	0.389(0.14)	0.417(0.15)	0.389(0.15)	0.444(0.16)	0.405(0.17)	0.393(0.15)	0.398(0.16)	0.431(0.18)
monoResMatch (Tosi et al., 2019)	0.363(0.21)	0.386(0.18)	0.439(0.18)	0.428(0.20)	0.391(0.17)	0.400(0.19)	0.354(0.18)	0.429(0.20)	0.385(0.19)	0.342(0.19)	0.368(0.20)	0.386(0.17)
SfMLearner (Zhou et al., 2017)	0.251(0.10)	0.268(0.09)	0.270(0.09)	0.284(0.11)	0.268(0.11)	0.271(0.10)	0.271(0.11)	0.292(0.12)	0.258(0.09)	0.245(0.09)	0.253(0.09)	0.254(0.09)
PackNet (Guizilini et al., 2020)	0.436(0.13)	0.394(0.13)	0.422(0.15)	0.435(0.15)	0.430(0.14)	0.429(0.14)	0.368(0.13)	0.403(0.12)	0.458(0.13)	0.450(0.13)	0.444(0.14)	0.386(0.17)
Monodepth2 (Godard et al., 2019)	0.627 (0.15)	0.634 (0.12)	0.635 (0.15)	0.625 (0.13)	0.619 (0.11)	0.619 (0.15)	0.580 (0.10)	0.649 (0.15)	0.667 (0.17)	0.602 (0.13)	0.635 (0.16)	0.590 (0.11)
CC (Ranjan et al., 2019)	0.493(0.19)	0.478(0.18)	0.501(0.21)	0.480(0.20)	0.494(0.19)	0.479(0.19)	0.400(0.15)	0.480(0.18)	0.525(0.18)	0.488(0.19)	0.483(0.20)	0.445(0.21)
SGDepth (Klingner et al., 2020)	0.497(0.17)	0.459(0.16)	0.487(0.19)	0.475(0.18)	0.487(0.17)	0.487(0.18)	0.437(0.14)	0.475(0.15)	0.525(0.15)	0.483(0.16)	0.495(0.18)	0.449(0.19)
Atapour *et al.* (Atapour et al., 2018)	0.281(0.12)	0.304(0.12)	0.313(0.12)	0.320(0.13)	0.309(0.13)	0.301(0.11)	0.309(0.13)	0.325(0.15)	0.287(0.11)	0.287(0.11)	0.282(0.11)	0.284(0.12)
T2Net (Zheng et al., 2018)	0.421(0.17)	0.367(0.15)	0.416(0.17)	0.403(0.17)	0.416(0.16)	0.390(0.16)	0.340(0.13)	0.404(0.15)	0.429(0.17)	0.349(0.14)	0.363(0.16)	0.393(0.17)
GASDA (Zhao et al., 2019)	0.414(0.18)	0.418(0.16)	0.426(0.14)	0.429(0.17)	0.428(0.16)	0.427(0.15)	0.377(0.16)	0.433(0.18)	0.420(0.17)	0.347(0.19)	0.383(0.19)	0.427(0.16)

for 20 epochs to get the best performance on *Average* metric at epoch 20. We use the model of `ResNet18` pre-trained on *ImageNet* and fine-tuned on *KITTI* with the resolution of 640 × 192 to test *PackNet* on `https://github.com/TRI-ML/packnet-sfm`. Similarly, in order to incorporate stereo geometric constraint into the monocular SfM framework, we use the model of `mono+stereo` pre-trained on *ImageNet* and *KITTI* with the resolution of 640 × 192 to evaluate the performance of *Monodepth2* on `https://github.com/nianticlabs/monodepth2`. For the multi-task SfM unsupervised learning methods, *CC* is evaluated with `DispNet, PoseNet, MaskNet and FlowNet` pre-trained model on *KITTI* through `https://github.com/anuragranj/cc`. We also test another recent work *SGDepth* on `https://github.com/ifnspaml/SGDepth` with the `full` model of semantic segmentation and depth prediction with the resolution of 640 × 192.

Since synthetic datasets like *V-KITTI* include multiple environments in spite of existing domain gap, we additionally evaluate the performance of three domain adaptation methods from *KITTI* benchmark, Atapour *et al.* (Atapour et al., 2018), *T2Net* (Zheng et al., 2018) and *GASDA* (Zhao et al., 2019). We follow the instruction on `https://github.com/atapour/monocularDepth-Inference` to evaluate the method proposed by Atapour *et al.* with the model pre-trained on `KITTI` and `DeepGTAV` (Miralles, 2017). *T2Net* is tested on `https://github.com/lyndonzheng/Synthetic2Realistic` with the weakly-supervised pre-

trained model for outdoor scenes of *KITTI* and *V-KITTI*. We then evaluate the performance of *GASDA* on https://github.com/sshan-zhao/GASDA with the model pre-trained on *V-KITTI* and *KITTI* using self-supervised stereo geometric information.

B.2. Detailed Evaluation Results across Environments

In this section, the detailed results with mean values and standard deviations are shown in Tab. 6 and Tab. 7, it can be seen that models with larger mean values tend to have more significant deviation for each environment. However, though some large standard deviations in Tab. 6 and Tab. 7 weaken the credibility and reliability for the performance of methods, the quality of depth map ground truths is assured. So we attribute it to the poor generalization ability of those algorithms since not all the methods present such poor results with too large variances, which cannot be correctly analyzed.

Moreover, all the evaluated baselines are visualized after adjustment under typical challenging environments, including dark illumination, snowy scene, and complex vegetation. See Fig. 13 for more details. From the results of supervised methods, it can be seen that the patterns of predicted depth maps are similar, especially for *BTS* (Lee et al., 2019) and *VNL* (Yin et al., 2019), where the top and bottom areas are dark while the middle areas are bright due to overfitting, see buildings as examples. But *VNL* (Yin et al., 2019) shows advantage on depth details (*e.g.* telephone poles and vegetation) in the middle areas which accounts for the best average performance.

Stereo training involved self-supervised methods (including *Monodepth2* (Godard et al., 2019) and *GASDA* (Zhao et al., 2019)) perform best continuous depth results for the same entity under all environments, *e.g.* depth values of buildings. Monocular video-based self-supervised methods do better in distinguishing relative depth from far and near areas, *e.g.* depth values for objects along different directions of roads, especially for multi-task learning ones *CC* (Ranjan et al., 2019) and *SGDepth* (Klingner et al., 2020). Besides, domain adaptation methods still suffer from domain gaps, which shows that synthetic multi-environment images help little to improve performance under real-world changing environments.

Qualitative analysis Qualitative results for different types of baselines are shown in Fig. 7. It can be seen that supervised methods *BTS* (Lee et al., 2019) and *VNL* (Yin et al., 2019) suffer from overfitting through the predicted pattern where the top and bottom areas are dark while the central areas are light, even for buildings. Stereo training involved methods with underlines (Godard et al., 2019; Zhao et al., 2019) perform continuous depth results for the same entity under all environments, *e.g.* the depth prediction of buildings compared to other self-supervised monocular (S-Sup-M) video based methods (Guizilini et al., 2020; Klingner et al., 2020) and syn-to-real (Syn-to-Real) domain adaptation method (Zheng et al., 2018), validating the improvement of robustness using stereo geometry constraint like quantitative results in Tab. 2.

B.3. Justification of SeasonDepth Used for Model Training

Based on Fig. 11 with shadows of cross-slice standard deviation after zooming 0.5, 0.2, and 0.5 times, it can be seen that after the fine-tuning, overall performance is improved while some $Variance$ and $RelativeRange$ results still perform badly, especially for SfMLearner (Zhou et al., 2017). Since the lack of dynamic objects in our dataset, we further conduct more experiment to justify that the depth accuracy and the ground truth are good enough for the dataset usage of autonomous driving for model training. Specifically, inspired by cross-dataset transfer degradation evaluation in (Ranftl et al., 2020), we compare our dataset with the stereo depth dataset *Cityscapes* (Cordts et al., 2016) in terms of the degraded performance on KITTI dataset after cross-dataset fine-tuning. Using the same pre-trained models on KITTI as introduced in the last section, we fine-tune BTS (Lee et al., 2019) and SfMLearner (Zhou et al., 2017) models on *SeasonDepth* and *Cityscapes* dataset with the same amount of images, and evaluate the per-epoch depth prediction on KITTI validation set via the metrics of *AbsRel*, *SqRel*, *iMAE* and *iRMSE* from (Uhrig et al., 2017). The shadows show the range of metrics after centering and zooming 0.02, 0.01, 0.035 and 0.04 times for clarity.

From the quantitative results in Fig. 6, we can see that although the performance will be degraded compared to the KITTI pre-trained models due to domain shift when fine-tuning, the performance fine-tuned on *SeasonDepth* is better than models fine-tuned on *Cityscapes*, especially for *SfMLearner* method and *iMAE* and *iRMSE* metrics. Besides, the fluctuation of models fine-tuned on *SeasonDepth* is much less than those fine-tuned on *Cityscapes* in terms of *AbsRel* and *SqRel* metrics. Based on the qualitative performance in Fig. 8, we can find that models fine-tuned on *SeasonDepth* perform better than those fine-tuned on *Cityscapes* on the unseen KITTI dataset. Consequently, although the depth maps of *SeasonDepth* are reconstructed from structure from motion and do not contain dynamic objects, the ground truth accuracy is eligible to be used for model training compared to the stereo depth dataset *Cityscapes*, justifying our ground truth accuracy is adequate though it is not perfect.

B.4. Details about Cross-dataset Justification on KITTI

In this section, we present more details about the cross-dataset experiment to justify our depth quality

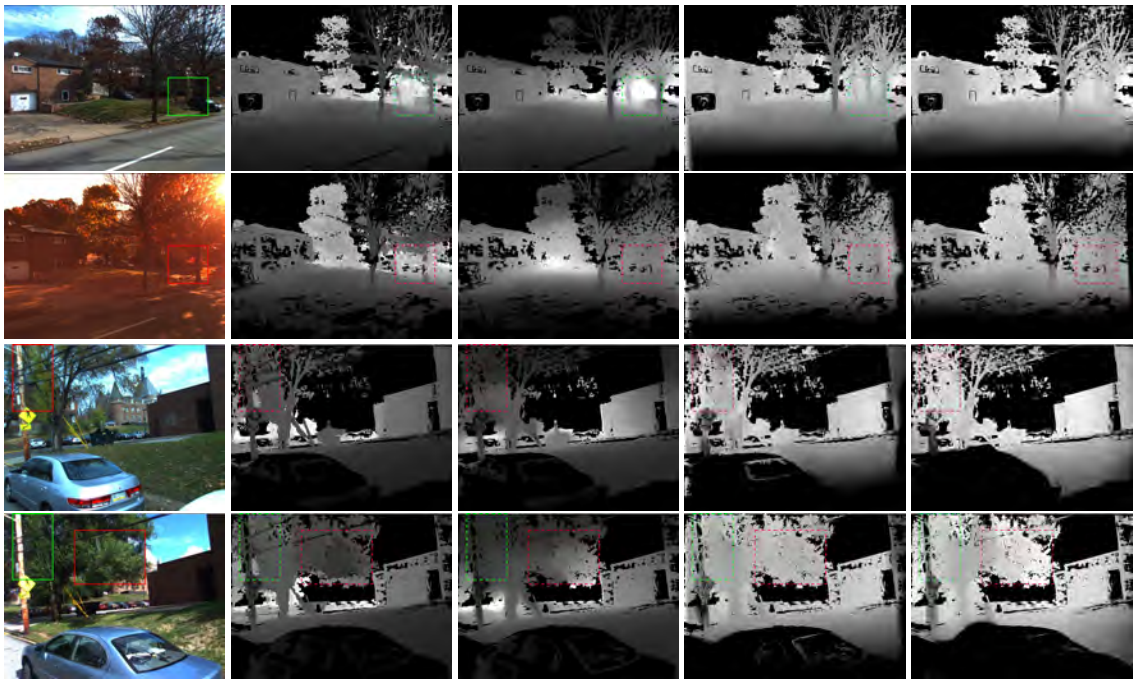


Figure 12. Qualitative comparison results with illumination or vegetation changes. The conditions from top to down are *C+MF, Nov.* 22^{nd}, *LS+MF, Nov.* 3^{rd}, *C+MF, Nov.* 22^{nd} and *C+F, Oct.* 1^{st}. Green blocks indicate good performance while red blocks are for bad results.

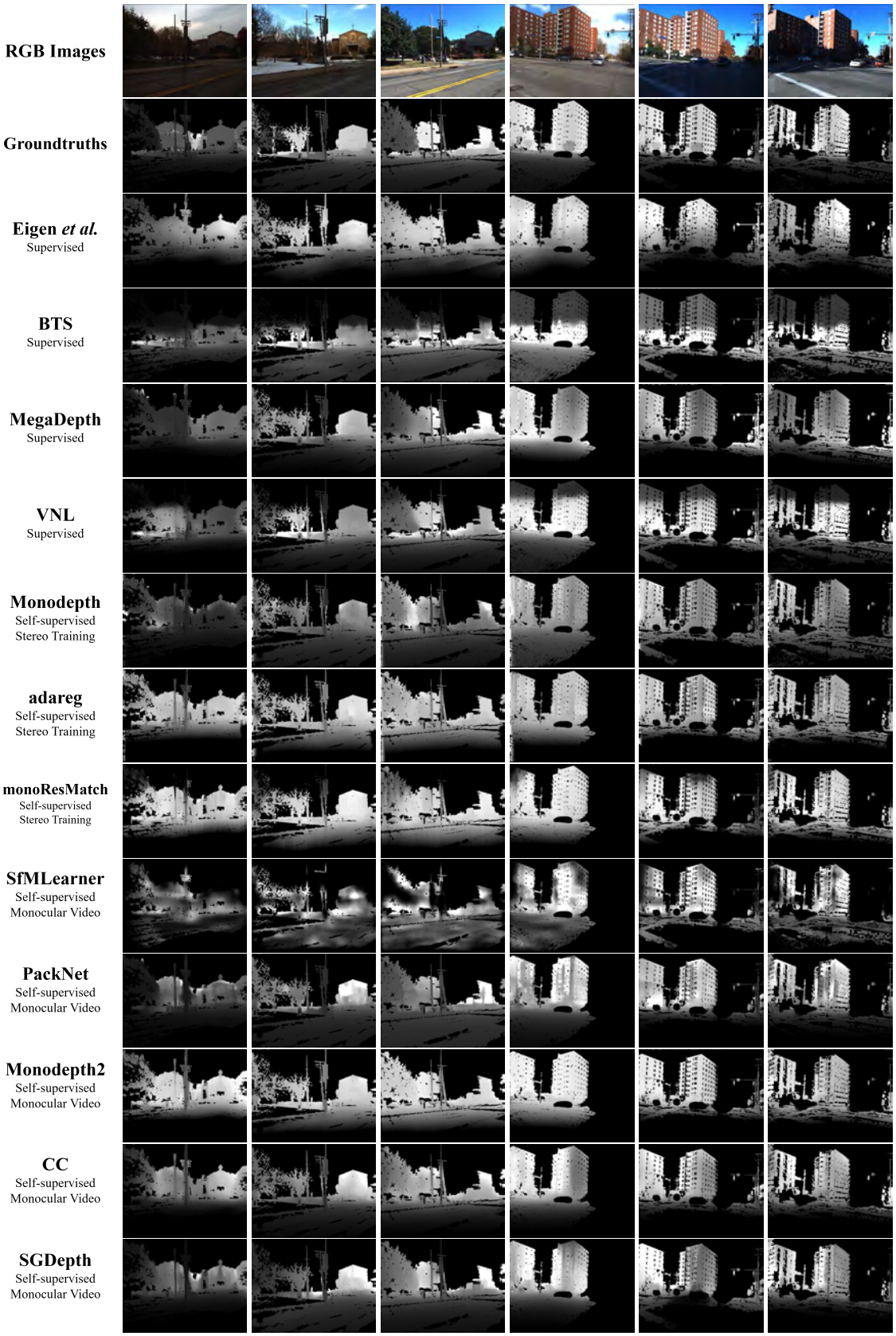
for model training. As it is introduced in Sec. B.1, we choose the KITTI pre-trained models for *BTS* and *SfMLearner* methods, and fine-tune them on our training set and *Cityscapes* (Cordts et al., 2016) for 50 epochs, respectively. Finally, we evaluate the cross-dataset transfer performance on KITTI validation set (Uhrig et al., 2017) using `development kit` from `http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction`. We choose 11407 images from `train_extra` in *Cityscapes* (Cordts et al., 2016) to fine-tune the models, which is exactly the same amount of images in our training set to make the comparison fair.

To fine-tune the self-supervised model of *SfMLearner*, we set `batch_size` to be 4, `epoch_size` to be 1000 and `sequence_length` to be 1000. Along with the instructions to train with own data `https://github.com/ClementPinard/SfmLearner-Pytorch/issues/108`, we crop a quarter of the bottom in the image and resize it to be 416×128 to remove the car logo in *Cityscapes* dataset. We change the intrinsic parameters accordingly to make them consistent with cropped images. For a fair comparison, we also conduct such cropping for the images from *SeasonDepth* dataset. When testing the KITTI validation set, we resize the images to be 416×128 before feeding them into the networks.

When fine-tuning supervised *BTS* model, we set `batch_size` to be 16, `input_size` to be 256×192 for *SeasonDepth* images and 256×128 for *Cityscapes* images. For depth ground truth, we directly adopt the depth maps in *Cityscapes* as supervision signals while for *SeasonDepth* dataset, we only consider the non-zero pixels and conduct alignment using mean value to the ground truth to construct loss when fine-tuning. The experimental results show that such alignment to construct supervised loss is effective using our dataset for supervised model training.

B.5. Further Discussion

In this section, we will discuss how to improve the robustness across multiple environments methodologically to boost more research on long-term robust visual perception. Research about long-term performance under changing environments stems from visual place recognition and localization. Most of deep learning based methods leverage environmentally-insensitive perceptual auxiliary information like semantic (Xu et al., 2018; Benbihi et al., 2020), geometric (Piasco et al., 2019; 2020), context-aware (Kim et al., 2017; Xin et al., 2019) information, or learn the domain-invariant representation (Zhou et al., 2020; Tang et al., 2020) or image translation (Jenicek & Chum, 2019; Zheng et al.,



To be continued

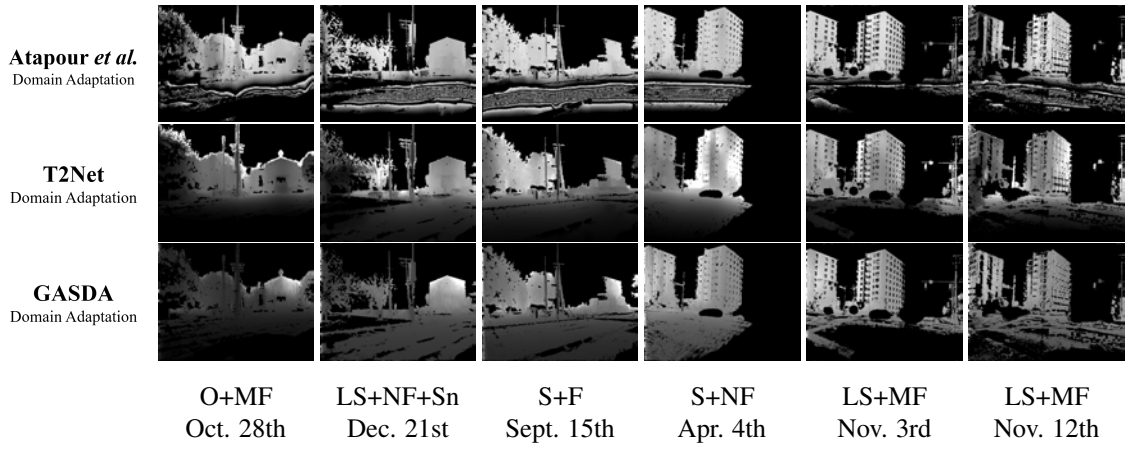


Figure 13. Qualitative results for all the baselines with multiple illumination, vegetation and weather conditions.

2020) in multi-domain setting.

Our work targets the robustness of these auxiliary perceptual tasks, evaluating the best monocular depth prediction methods under different environments. The findings from our benchmark are intuitively environmentally-insensitive, *e.g.*, stereo geometric constraint, multi-task learning with semantic segmentation. Following the methodology and taxonomy of long-term visual place recognition methods, many potential models can be developed using our dataset and benchmark, like domain-invariant feature-based methods, attention mechanism involved methods *etc.*

C. Limitation and Discussion

In this section, we discuss the limitation of our work. As mentioned before, our *SeasonDepth* dataset is built based on CMU Visual Localization dataset, which was initially collected for visual localization and contained multiple scenes but without challenging night scenes. Although it is different from the dataset for autonomous driving like *KITTI*, which causes concern about the evaluation due to the domain gap. However, based on the experimental evidence, it is acceptable that fine-tuned models only provide limited help in terms of $Variance$ and $RelativeRange$. Although dynamic objects are not included in the dataset to ensure accuracy and reliability, it does not affect the evaluation for real driving applications because it cannot be distinguished whether the objects are dynamic or static given a single monocular image when testing. And the cross-dataset justification experiment also shows that missing dynamic objects do not influence the model training too much. Consequently, the evaluation of the depth prediction of static objects can reveal the performance of dynamic objects, although they are not involved in the ground truth.

Besides, though normalizing the scale of evaluation metrics through alignment of mean and variance can also be done through quantile alignment shown in Sec A.2, it is more sensitive to noise to adopt quantile-based alignment of every single image for evaluation. Although we try our best to survey and test the open-source representative models as much as possible, it is impossible to involve all the monocular depth prediction methods in our benchmark. So we will release the test set and benchmark toolkit to make up for it. Another limitation is that it is not straightforward to train models on the dataset because of the ground truths of scaleless relative values, but it can be trained after the mean value alignment to the ground truth just as the fine-tuned BTS does. It can also reflect how environmental changes affect depth prediction models and give hints of what kind of method is more promising to this problem.