Transformers in Self-Supervised Monocular Depth Estimation with Unknown Camera Intrinsics

Arnav Varma[a], Hemang Chawla[a], Bahram Zonooz and Elahe Arani
Advanced Research Lab, NavInfo Europe, The Netherlands
{*arnav.varma, hemang.chawla, elahe.arani*}*@navinfo.eu, bahram.zonooz@gmail.com*

Keywords: Transformers, Convolutional Neural Networks, Monocular Depth Estimation, Camera Self-Calibration, Self-Supervised Learning

Abstract: The advent of autonomous driving and advanced driver assistance systems necessitates continuous developments in computer vision for 3D scene understanding. Self-supervised monocular depth estimation, a method for pixel-wise distance estimation of objects from a single camera without the use of ground truth labels, is an important task in 3D scene understanding. However, existing methods for this task are limited to convolutional neural network (CNN) architectures. In contrast with CNNs that use localized linear operations and lose feature resolution across the layers, vision transformers process at constant resolution with a global receptive field at every stage. While recent works have compared transformers against their CNN counterparts for tasks such as image classification, no study exists that investigates the impact of using transformers for self-supervised monocular depth estimation. Here, we first demonstrate how to adapt vision transformers for self-supervised monocular depth estimation. Thereafter, we compare the transformer and CNN-based architectures for their performance on KITTI depth prediction benchmarks, as well as their robustness to natural corruptions and adversarial attacks, including when the camera intrinsics are unknown. Our study demonstrates how transformer-based architecture, though lower in run-time efficiency, achieves comparable performance while being more robust and generalizable.

1 INTRODUCTION

There have been rapid improvements in scene understanding for robotics and advanced driver assistance systems (ADAS) over the past years. This success is attributed to the use of Convolutional Neural Networks (CNNs) within a mostly encoder-decoder paradigm. Convolutions provide spatial locality and translation invariance which has proved useful for image analysis tasks. The encoder, often a convolutional Residual Network (ResNet) (He et al., 2016), learns feature representations from the input and is followed by a decoder which aggregates these features and converts them into final predictions. However, the choice of architecture has a major impact on the performance and generalizability of the task.

While CNNs have been the preferred architecture in computer vision, transformers have also recently gained traction (Dosovitskiy et al., 2021) motivated by their success in natural language processing (Vaswani et al., 2017). Notably, they have also outperformed CNNs for object detection (Carion et al., 2020) and semantic segmentation (Zheng et al., 2021). This is also reflected in methods for monocular dense depth estimation, a pertinent task for autonomous planning and navigation, where supervised transformer-based methods (Li et al., 2020; Ranftl et al., 2021) have been proposed as an alternative to supervised CNN-based methods (Lee et al., 2019; Aich et al., 2021). However, supervised methods require extensive RGB-D ground truth collected from costly LiDARs or multi-camera rigs. Instead, self-supervised methods have increasingly utilized concepts of Structure from Motion (SfM) with known camera intrinsics to train monocular depth and ego-motion estimation networks simultaneously (Guizilini et al., 2020; Lyu et al., 2020; Chawla et al., 2021). While transformer ingredients such as attention have been utilized for self-supervised depth estimation (Johnston and Carneiro, 2020), most methods are nevertheless limited to the use of CNNs that have localized linear operations and lose feature resolution during downsampling to increase their limited receptive field (Yang et al., 2021).

On the other hand, transformers with fewer in-

[a]Equal contribution

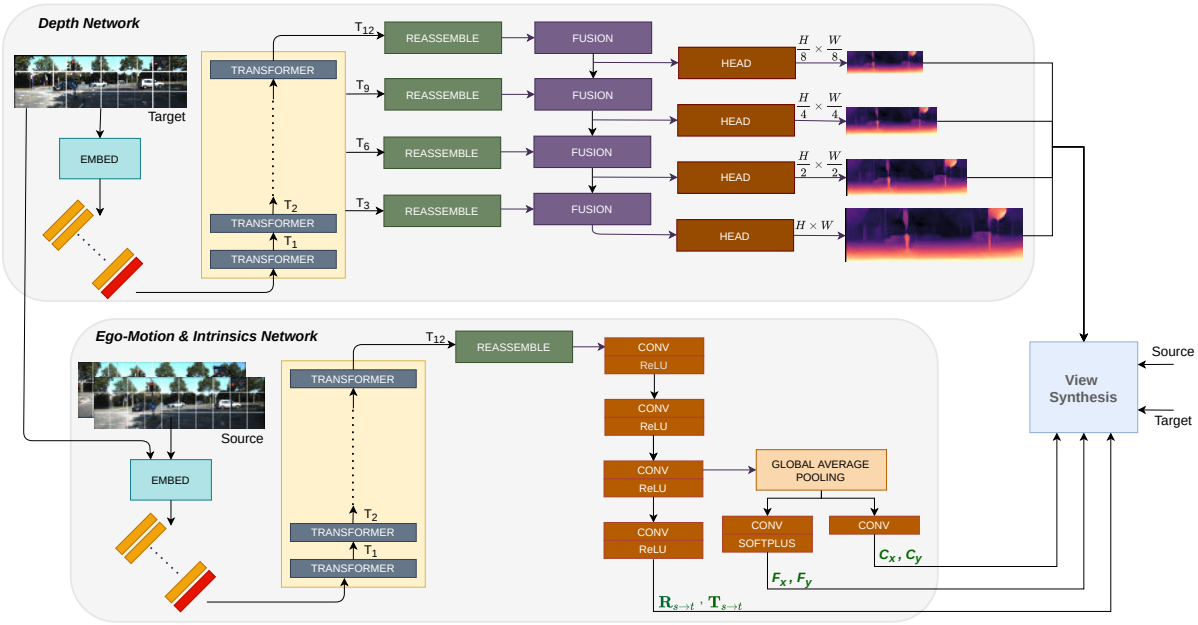

Figure 1: An overview of Monocular Transformer Structure from Motion Learner (MT-SfMLearner) with learned intrinsics. We readapt modules from Dense Prediction Transformer (DPT) and Monodepth2 to be trained with appearance-based losses for self-supervised monocular depth, ego-motion, and intrinsics estimation.

ductive biases allow for more globally coherent predictions with different layers attending to local and global features simultaneously (Touvron et al., 2021). However, transformers require more training data and can be more computationally demanding (Caron et al., 2021). While multiple studies have compared transformers against CNNs for tasks such as image classification (Raghu et al., 2021; Bhojanapalli et al., 2021), no study exists that evaluates the impact of transformers in self-supervised monocular depth estimation, including when the camera intrinsics may be unknown.

In this work, we conduct a comparative study between CNN- and transformer-based architectures for self-supervised monocular depth estimation. Our contributions are as follows:

- We demonstrate how to adapt vision transformers for self-supervised monocular depth estimation by implementing a method called Monocular-Transformer SfMLearner (MT-SfMLearner).
- We compare MT-SfMLearner and CNNs for their performance on the KITTI monocular depth Eigen Zhou split (Eigen et al., 2014) and the online depth prediction benchmark (Geiger et al., 2013).
- We investigate the impact of architecture choices for the individual depth and ego-motion networks on performance as well as robustness to natural corruptions and adversarial attacks.
- We also introduce a modular method that simultaneously predicts camera focal lengths and principal point from the images themselves and can easily be utilized within both CNN- and transformer-based architectures.
- We study the accuracy of intrinsics estimation as well as its impact on the performance and robustness of depth estimation.
- Finally, we also compare the run-time computational and energy efficiency of the architectures for depth and intrinsics estimation.

MT-SfMLearner provides real-time depth estimates and illustrates how transformer-based architecture, though lower in run-time efficiency, can achieve comparable performance as the CNN-based architectures while being more robust under natural corruptions and adversarial attacks, even when the camera intrinsics are unknown. Thus, our work presents a way to analyze the trade-off between the performance, robustness, and efficiency of transformer- and CNN-based architectures for depth estimation.

2 RELATED WORKS

Recently, transformer architectures such as Vision Transformer (ViT) (Ranftl et al., 2021) and Data-efficient image Transformer (DeiT) (Touvron et al.,

2021) have outperformed CNN architectures in image classification. Studies comparing ViT and CNN architectures like ResNet have further demonstrated that transformers are more robust to natural corruptions and adversarial examples in classification (Bhojanapalli et al., 2021; Paul and Chen, 2021). Motivated by their success, researchers have replaced CNN encoders with transformers in scene understanding tasks such as object detection (Carion et al., 2020; Liu et al., 2021), semantic segmentation (Zheng et al., 2021; Strudel et al., 2021), and supervised monocular depth estimation (Ranftl et al., 2020; Yang et al., 2021).

For *supervised* monocular depth estimation, Dense Prediction Transformer (DPT) (Ranftl et al., 2020) uses ViT as the encoder with a convolutional decoder and shows more coherent predictions than CNNs due to the global receptive field of transformers. TransDepth (Yang et al., 2021) additionally uses a ResNet projection layer and attention gates in the decoder to induce the spatial locality of CNNs for *supervised* monocular depth and surface-normal estimation. Lately, some works have inculcated elements of transformers such as self-attention (Vaswani et al., 2017) in *self-supervised* monocular depth estimation (Johnston and Carneiro, 2020; Xiang et al., 2021). However, there has been no investigation of transformers to replace the traditional CNN-based methods (Godard et al., 2019; Lyu et al., 2020) for *self-supervised* monocular depth estimation.

Moreover, self-supervised monocular depth estimation still requires prior knowledge of the camera intrinsics (focal length and principal point) during training, which may be different for each data source, may change over time, or be unknown a priori (Chawla et al., 2020). While multiple approaches to *supervised* camera intrinsics estimation have been proposed (Lopez et al., 2019; Zhuang et al., 2019), not many *self-supervised* approaches exist (Gordon et al., 2019).

Therefore, we investigate the impacts of transformer architectures on self-supervised monocular depth estimation for their performance, robustness, and run-time efficiency, even when intrinsics are unknown.

3 METHOD

Our objective is to study the effect of utilizing vision transformers for self-supervised monocular depth estimation in contrast with the contemporary methods that utilize Convolutional Neural Networks.

Given a set of n images from a video sequence, we simultaneously train depth and ego-motion prediction networks. The inputs to the networks are a sequence of temporally consecutive RGB image triplets $\{I_{-1}, I_0, I_1\} \in \mathbb{R}^{H\times W\times 3}$. The depth network learns the model $f_D : \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^{H\times W}$ to output dense depth (or disparity) for each pixel coordinate p of a single image. Simultaneously, the ego-motion network learns the model $f_E : \mathbb{R}^{2\times H\times W\times 3} \rightarrow \mathbb{R}^6$ to output relative translation (t_x, t_y, t_z) and rotation (r_x, r_y, r_z) forming the affine transformation $\begin{bmatrix} \hat{R} & \hat{t} \\ 0 & 1 \end{bmatrix} \in \mathrm{SE}(3)$ between a pair of overlapping images. The predicted depth $\hat{D}$ and ego-motion $\hat{T}$ are linked together via the perspective projection model,

$$p_s \sim K\hat{R}_{s\leftarrow t}\hat{D}_t(p_t)K^{-1}p_t + K\hat{T}_{s\leftarrow t}, \qquad (1)$$

that warps the source images $I_s \in \{I_{-1}, I_1\}$ to the target image $I_t \in \{I_0\}$, with the camera intrinsics described by K. This process is called view synthesis, as shown in Figure 1. We train the networks using the appearance-based *photometric* loss between the real and synthesized target images, as well as a *smoothness* loss on the depth predictions (Godard et al., 2019).

3.1 Architecture

Here we describe Monocular Transformer Structure from Motion Learner (MT-SfMLearner), our transformer-based method for self-supervised monocular depth estimation (Figure 1).

3.1.1 Depth Network

For the depth network, we readapt the Dense Prediction Transformer (DPT) (Ranftl et al., 2020) for *self-supervised learning*, with a DeiT-Base (Touvron et al., 2021) in the encoder. There are five components of the depth network:

- An ***Embed*** module, which is a part of the encoder, takes an image $I \in \mathbb{R}^{H\times W\times 3}$, and converts non-overlapping image patches of size $p \times p$ into $N_p = H \cdot W/p^2$ tokens $t_i \in \mathbb{R}^d \ \forall i \in [1, 2, ...N_p]$, where $d = 768$ for DeiT-Base. This is implemented as a large $p \times p$ convolution with stride $s = p$ where $p = 16$. The output from this module is concatenated with a *readout* token of the same size as the remaining tokens.
- The ***Transformer*** block, that is also a part of the encoder, consists of 12 transformer layers which process these tokens with multi-head self-attention (MHSA) (Vaswani et al., 2017) modules. MHSA processes inputs at constant resolution and can simultaneously attend to global and local features.

Table 1: Architecture details of the *Reassemble* modules. DN and EN refer to depth and ego-motion networks, respectively. The subscripts of *DN* refer to the transformer layer from which the respective *Reassemble* module takes its input (see Figure 1). Input image size is $H \times W$, p refers to the patch size, $N_p = H \cdot W / p^2$ refers to the number of patches from the image, and d refers to the feature dimension of the transformer features.

Operation	**Input size**	**Output size**	**Function**	**Parameters** (DN_3, DN_6, DN_9, DN_{12}, EN)
Read	$(N_p+1) \times d$	$N_p \times d$	Drop readout token	–
Concatenate	$N_p \times d$	$d \times H/p \times W/p$	Transpose and Unflatten	–
Pointwise Convolution	$d \times H/p \times W/p$	$N_c \times H/p \times W/p$	N_c channels	$N_c = [96, 768, 1536, 3072, 2048]$
Strided Convolution	$N_c \times H/p \times W/p$	$N_c \times H/2p \times W/2p$	$k \times k$ convolution, Stride= 2, N_c channels, padding= 1	$k = [-, -, -, 3, -]$
Transpose Convolution	$N_c \times H/p \times W/p$	$N_c \times H/s \times W/s$	$p/s \times p/s$ deconvolution, stride= p/s, N_c channels	$s = [4, 2, -, -, -]$

- Four ***Reassemble*** modules in the decoder, which are responsible for extracting image-like features from the 3rd, 6th, 9th, and 12th (final) transformer layers by dropping the readout token and concatenating the remaining tokens in 2D. This is followed by pointwise convolutions to change the number of channels, and transpose convolution in the first two reassemble modules to upsample the representations (corresponding to T_3 and T_6 in Figure 1). To make the transformer network comparable to its convolutional counterparts, we increase the number of channels in the pointwise convolutions of the last three *Reassemble* modules by a factor of 4 with respect to DPT. The exact architecture of the *Reassemble* modules can be found in Table 1.

- Four ***Fusion*** modules in the decoder, based on RefineNet (Lin et al., 2017). They progressively fuse information from the *Reassemble* modules with information passing through the decoder, and upsample the features by 2 at each stage. Unlike DPT, we enable batch normalization in the decoder as it was found to be helpful for self-supervised depth prediction. We also reduce the number of channels in the *Fusion* block to 96 from 256 in DPT.

- Four ***Head*** modules at the end of each *Fusion* module to predict depth at 4 scales following previous self-supervised methods (Godard et al., 2019). Unlike DPT, the *Head* modules use 2 convolutions instead of 3 as we found no difference in performance. For further details of the *Head* modules, refer to Table 2.

Table 2: Architecture details of *Head* modules in Figure 1.

Layers
32 3×3 *Convolutions*, stride=1, padding= 1
ReLU
Bilinear Interpolation to upsample by 2
32 *Pointwise Convolutions*
Sigmoid

3.1.2 Ego-Motion Network

For the ego-motion network, we adapt DeiT-Base (Touvron et al., 2021) in the encoder. Since the input to the transformer for the ego-motion network consists of two images concatenated along the channel dimension, we repeat the embedding layer accordingly. We use a *Reassemble* module to pass transformer tokens to the decoder. For details on the structure of this *Reassemble* module, refer to Table 1. We adopt the decoder for the ego-motion network from Monodepth2 (Godard et al., 2019).

When both depth and ego-motion networks use transformers as described above, we refer to the resulting architecture as Monocular Transformer Structure from Motion Learner (*MT-SfMLearner*).

3.2 Appearance-based Losses

Following contemporary self-supervised monocular depth estimation methods, we adopt the *appearance-based losses* and an *auto-masking* procedure from the CNN-based Monodepth2 (Godard et al., 2019) for the above described transformer-based architecture as well. We employ a photometric reprojection loss composed of the pixel-wise ℓ_1 distance and the Structural Similarity (SSIM) between the real and synthesized target images, along with a multi-scale edge-aware *smoothness* loss on the depth predictions. We also use auto-masking to disregard the temporally stationary pixels in the image triplets. Furthermore, we reduce texture-copy artifacts by calculating the total loss after upscaling the depths, predicted at 4 scales, from intermediate decoder layers to the input resolution.

3.3 Intrinsics

Accurate camera intrinsics given by

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

are essential to self-supervised depth estimation as can be seem from Equation 1. However, the intrinsics may vary within a dataset with videos collected from different camera setups, or over a long period of time. These parameters can also be unknown for crowdsourced datasets.

We address this by introducing an intrinsics estimation module. We modify the ego-motion network, which takes a pair of consecutive images as input, and learns to estimate the focal length and principal point along with the translation and rotation. Specifically, we add a convolutional path in the ego-motion decoder to learn the intrinsics. The decoder features before activation from the penultimate layer are passed through a global average pooling layer, followed by two branches of pointwise convolutions to reduce the number of channels from 256 to 2. One branch uses a softplus activation to estimate focal lengths along x and y axes as the focal length is always positive. The other branch doesn't use any activation to estimate the principal point as it has no such constraint. Note that the ego-motion decoder is the same for both the convolutional as well as transformer architectures. Consequently, the intrinsics estimation method can be modularly utilized with both architectures. Figure 1 demonstrates MT-SfMLearner with learned intrinsics.

4 RESULTS

In this section, we perform a comparative analysis between our transformer-based method, MT-SfMLearner, and the existing CNN-based methods for self-supervised monocular depth estimation. We also perform a contrastive study on the architectures for the depth and ego-motion networks to evaluate their impact on the prediction accuracy, robustness to natural corruptions and adversarial attacks, and the run-time computational and energy efficiency. Finally, we analyze the correctness and run-time efficiency of intrinsics predictions, and also study its impact on the accuracy and robustness of depth estimation.

4.1 Implementation Details

4.1.1 Dataset

We report all results on the Eigen Split (Eigen et al., 2014) of KITTI (Geiger et al., 2013) dataset after removing the static frames as per (Zhou et al., 2017), unless stated otherwise. This split contains $39,810$ training, 4424 validation, and 697 test images, respectively. All results are reported on the per-image scaled dense depth prediction without post-processing (Godard et al., 2019), unless stated otherwise.

4.1.2 Training Settings

The networks are implemented in PyTorch (Paszke et al., 2019) and trained on a TeslaV100 GPU for 20 epochs at a resolution of 640×192 with batch-size 12. MT-SfMLearner is further trained at 2 more resolutions - 416×128 and 1024×320, with batch-sizes of 12 and 2, respectively for experiments in Section 4.2. The depth and ego-motion encoders are initialized with ImageNet (Deng et al., 2009) pre-trained weights. We use the Adam (Kingma and Ba, 2014) optimizer for CNN-based networks (in Sections 4.3 and 4.4) and AdamW (Loshchilov and Hutter, 2017) optimizer for transformer-based networks with initial learning rates of $1e^{-4}$ and $1e^{-5}$, respectively. The learning rate is decayed after 15 epochs by a factor of 10. Both optimizers use $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

4.2 Depth Estimation Performance

First we compare MT-SfMLearner, where both depth and ego-motion networks are transformer-based, with the existing fully convolutional neural networks for their accuracy on self-supervised monocular depth estimation. Their performance is evaluated using metrics from (Eigen et al., 2014) up to a fixed range of 80 m as shown in Table 3. We compare the methods at different input image sizes clustered into categories of Low-Resolution (LR), Medium-Resolution (MR), and High-Resolution (HR). We do not compare against methods that use ground-truth semantic labels during training. All methods assume known ground-truth camera intrinsics.

We observe that MT-SfMLearner is able to achieve comparable performance at all resolutions under *error* as well as *accuracy* metrics. This includes methods that also utilize a heavy encoder such as ResNet-101 (Johnston and Carneiro, 2020) and PackNet (Guizilini et al., 2020).

Online Benchmark: We also measure the performance of MT-SfMLearner on the KITTI Online Benchmark for depth prediction[1] using the metrics from (Uhrig et al., 2017). We train on an image size of 1024×320, and add the G2S loss (Chawla et al., 2021) for obtaining predictions at metric scale. Results ordered by their rank are shown in Table 4. The performance of MT-SfMLearner is on par with state-of-the-art self-supervised methods, and outperforms

[1] http://www.cvlibs.net/datasets/kitti/eval_depth.php?benchmark=depth_prediction. See under MT-SfMLearner.

Table 3: Quantitative results comparing MT-SfMLearner with existing methods on KITTI Eigen split. For each category of image sizes, the best results are displayed in bold, and the second best results are underlined.

	Methods	Resolution	Error↓				Accuracy↑		
			Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
LR	SfMLearner(Zhou et al., 2017)	416×128	0.208	1.768	6.856	0.283	0.678	0.885	0.957
	GeoNet (Yin and Shi, 2018)	416×128	0.155	1.296	5.857	0.233	0.793	0.931	0.973
	Vid2Depth (Mahjourian et al., 2018)	416×128	0.163	1.240	6.220	0.250	0.762	0.916	0.968
	Struct2Depth (Casser et al., 2019)	416×128	0.141	1.026	5.291	0.215	0.816	0.945	0.979
	Roussel et al. (Roussel et al., 2019)	416×128	0.179	1.545	6.765	0.268	0754	0916	0.966
	VITW (Gordon et al., 2019)	416×128	0.129	0.982	5.230	0.213	0.840	0.945	0.976
	Monodepth2 (Godard et al., 2019)	416×128	0.128	1.087	5.171	0.204	**0.855**	**0.953**	0.978
	MT-SfMLearner (Ours)	416×128	**0.125**	**0.905**	**5.096**	**0.203**	0.851	0.952	**0.980**
MR	CC (Ranjan et al., 2019)	832×256	0.140	1.070	5.326	0.217	0.826	0.941	0.975
	SC-SfMLearner (Bian et al., 2019)	832×256	0.137	1.089	5.439	0.217	0.830	0.942	0.975
	Monodepth2 (Godard et al., 2019)	640×192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	SG Depth (Klingner et al., 2020)	640×192	0.117	0.907	4.844	0.194	0.875	0.958	0.980
	PackNet-SfM (Guizilini et al., 2020)	640×192	0.111	0.829	4.788	0.199	0.864	0.954	0.980
	Poggi et. al (Poggi et al., 2020)	640×192	0.111	0.863	4.756	0.188	0.881	0.961	0.982
	Johnston & Carneiro (Johnston and Carneiro, 2020)	640×192	**0.106**	0.861	4.699	**0.185**	**0.889**	**0.962**	0.982
	HR-Depth (Lyu et al., 2020)	640×192	0.109	**0.792**	**4.632**	**0.185**	0.884	**0.962**	**0.983**
	MT-SfMLearner (Ours)	640×192	0.112	0.838	4.771	0.188	0.879	0.960	0.982
HR	Packnet-SfM (Guizilini et al., 2020)	1280×384	0.107	0.803	4.566	0.197	0.876	0.957	0.979
	HR-Depth (Lyu et al., 2020)	1024×320	0.106	**0.755**	**4.472**	**0.181**	0.892	**0.966**	**0.984**
	G2S (Chawla et al., 2021)	1024×384	0.109	0.844	4.774	0.194	0.869	0.958	0.981
	MT-SfMLearner (Ours)	1024×320	**0.104**	0.799	4.547	**0.181**	**0.893**	0.963	0.982

several supervised methods. This further confirms that the transformer-based method can achieve comparable performance to the convolutional neural networks for self-supervised depth estimation.

4.3 Contrastive Study

We saw in the previous section that MT-SfMLearner performs competently on independent and identically distributed (i.i.d.) test set with respect to the state-of-the-art. However, networks that perform well on an i.i.d. test set may still learn shortcut features that are non-robust and generalize poorly to out-of-distribution (o.o.d.) datasets (Geirhos et al., 2020). Since self-supervised monocular depth estimation networks concurrently train an ego-motion network (see Equation 1), we investigate the impact of each network's architecture on depth estimation.

We consider both Convolutional (C) and Transformer (T) networks for depth and ego-motion estimation. The resulting four combinations of (Depth Network, Ego-Motion Network) architectures are (C, C), (C, T), (T, C), and (T, T), ordered on the basis of their increasing influence of transformers on depth estimation. To compare our transformer-based method fairly with convolutional networks, we utilize Monodepth2 (Godard et al., 2019) with a ResNet-101 (He et al., 2016) encoder. All four combinations are trained thrice using the settings described in Section 4.1 for an image size of 640×192. All combinations assume known ground-truth camera intrinsics.

Table 4: Quantitative comparison of *unscaled* dense depth prediction on the KITTI Depth Prediction Benchmark (online server). Supervised training with ground truth depths is denoted by D. Use of monocular sequences or stereo pairs is represented by M and S, respectively. Seg represents additional supervised semantic segmentation training. The use of GPS for multi-modal self-supervision is denoted by G.

Method	Train	SILog↓	SqErrRel↓
DORN (Fu et al., 2018)	D	11.77	2.23
SORD (Diaz and Marathe, 2019)	D	12.39	2.49
VNL (Yin et al., 2019)	D	12.65	2.46
DS-SIDENet (Ren et al., 2019)	D	12.86	2x.87
PAP (Zhang et al., 2019)	D	13.08	2.72
Guo et al. (Guo et al., 2018)	D+S	13.41	2.86
G2S (Chawla et al., 2021)	M+G	14.16	3.65
Ours	M+G	14.25	3.72
Monodepth2 (Godard et al., 2019)	M+S	14.41	3.67
DABC (Li et al., 2018b)	D	14.49	4.08
SDNet (Ochs et al., 2019)	D+Seg	14.68	3.90
APMoE (Kong and Fowlkes, 2019)	D	14.74	3.88
CSWS (Li et al., 2018a)	D	14.85	348
HBC (Jiang and Huang, 2019)	D	15.18	3.79
SGDepth (Klingner et al., 2020)	M+Seg	15.30	5.00
DHGRL (Zhang et al., 2018)	D	15.47	4.04
PackNet-SfM (Guizilini et al., 2020)	M+V	15.80	4.73
MultiDepth (Liebel and Körner, 2019)	D	16.05	3.89
LSIM (Goldman et al., 2019)	S	17.92	6.88
Monodepth (Godard et al., 2017)	S	22.02	20.58

4.3.1 Performance

For the four combinations, we report the best performance on i.i.d. in Table 5, and visualize the depth predictions for the same in Figure 2. The i.i.d. test set used for comparison is same as in Section 4.2.

We observe from Table 5 that the combination of transformer-based depth and ego-motion networks i.e MT-SfMLearner performs best under two of the *error* metrics as well as two of the *accuracy* metrics. The remaining combinations perform comparably on all the metrics.

From Figure 2, we observe more uniform estimates for larger objects like vehicles, vegetation, and

Table 5: Quantitative results on KITTI Eigen split for all four architecture combinations of depth and ego-motion networks. The best results are displayed in bold, the second best are underlined.

Architecture	Error↓				Accuracy↑		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
C, C	**0.111**	0.897	4.865	0.193	**0.881**	0.959	0.980
C, T	0.113	0.874	4.813	0.192	0.880	**0.960**	0.981
T, C	0.112	0.843	**4.766**	0.189	0.879	**0.960**	**0.982**
T, T	0.112	**0.838**	4.771	**0.188**	0.879	**0.960**	**0.982**

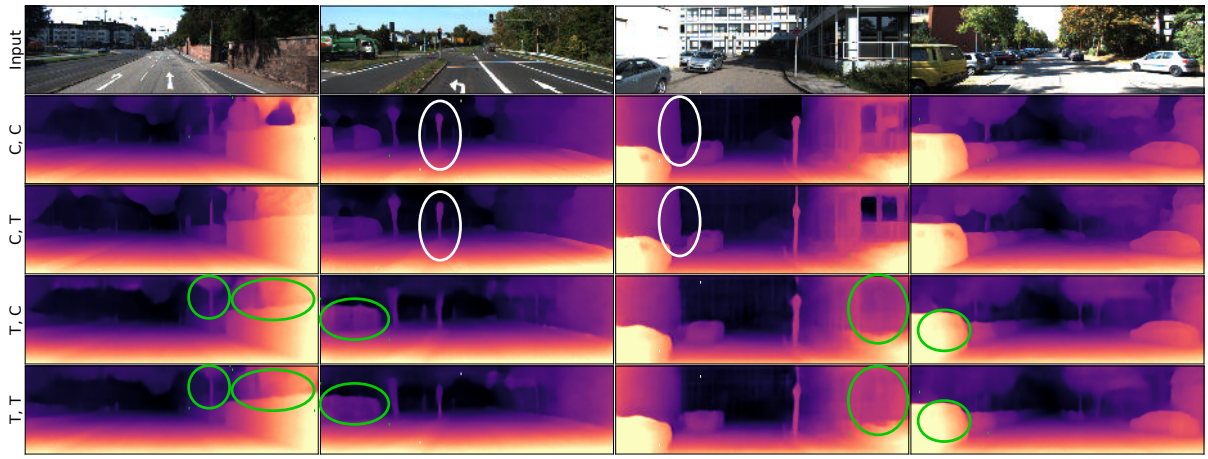

Figure 2: Disparity maps on KITTI for qualitative comparison of all four architecture combinations of depth and ego-motion networks. Example areas where the global receptive field of transformer is advantageous are highlighted in green. Example areas where local receptive field of CNNs is advantageous are highlighted in white.

buildings when depth is learned using transformers. Transformers are also less affected by reflection from windows of vehicles and buildings. This is likely because of the larger receptive fields of the self-attention layers, which lead to more globally coherent predictions. On the other hand, convolutional networks produce sharper boundaries, and perform better on thinner objects such as traffic signs and poles. This is likely because of the inherent inductive bias for spatial locality present in convolutional layers.

4.3.2 Robustness

While the different combinations perform comparably on the i.i.d. dataset, they may differ in robustness and generalizability. Therefore, we study the impact of natural corruptions and adversarial attacks on the depth performance using the following:

- **Natural corruptions**. Following (Hendrycks and Dietterich, 2019) and (Michaelis et al., 2019), we generate 15 corrupted versions of the KITTI i.i.d. test set at the highest severity(= 5). These natural corruptions fall under 4 categories - *noise* (Gaussian, shot, impulse), *blur* (defocus, glass, motion, zoom), *weather* (snow, frost, fog, brightness), and *digital* (contrast, elastic, pixelate, JPEG).
- **Projected Gradient Descent (PGD) adversarial examples**. Adversarial attacks make imperceptible (to humans) changes to input images to create adversarial examples that fool networks. We generate adversarial examples from the i.i.d. test set using PGD (Madry et al., 2018) at attack strength $\varepsilon \in \{0.25, 0.5, 1.0, 2.0, 4.0, 8.0, 16.0, 32.0\}$. The gradients are calculated with respect to the training loss. Following (Kurakin et al., 2016), the perturbation is accumulated over $min(\varepsilon + 4, \lceil 1.25 \cdot \varepsilon \rceil)$ iterations with a step-size of 1. When the test image is from the beginning or end of a KITTI sequence, the appearance-based loss is only calculated for the feasible pair of images.
- **Symmetrically flipped adversarial examples**. Inspired by (Wong et al., 2020), we generate these adversarial examples to fool the networks into predicting flipped estimates. For this, we use the gradients of the RMSE loss, where the targets are symmetrical horizontal and vertical flips of the i.i.d. predictions. This evaluation is conducted at attack strength $\varepsilon \in \{1.0, 2.0, 4.0\}$, similar to the PGD attack described above.

We report the mean RMSE across three training runs on natural corruptions, PGD adversarial examples, and symmetrically flipped adversarial examples in Figures 3, 4, and 5, respectively.

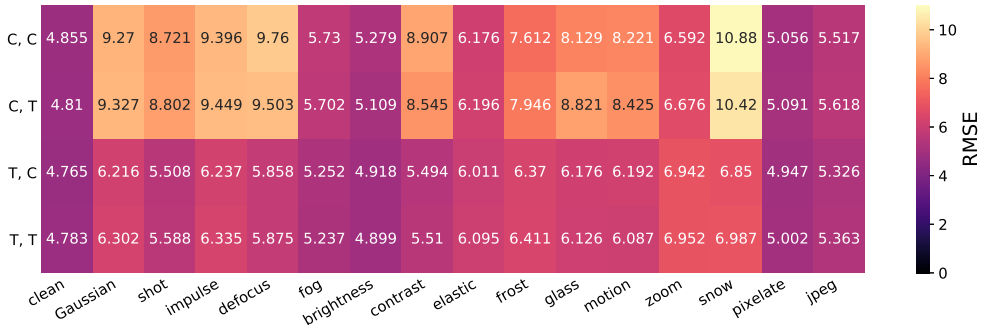

Figure 3: RMSE for natural corruptions of KITTI for all four combinations of depth and ego-motion networks. The i.i.d. evaluation is denoted by *clean*.

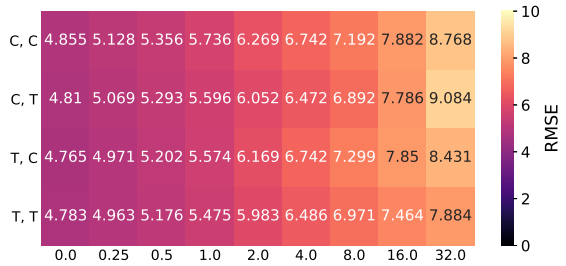

Figure 4: RMSE for adversarial corruptions of KITTI generated using PGD at all attack strengths (0.0 to 32.0) for all four combinations of depth and ego-motion networks. Attack strength 0.0 refers to i.i.d. evaluation.

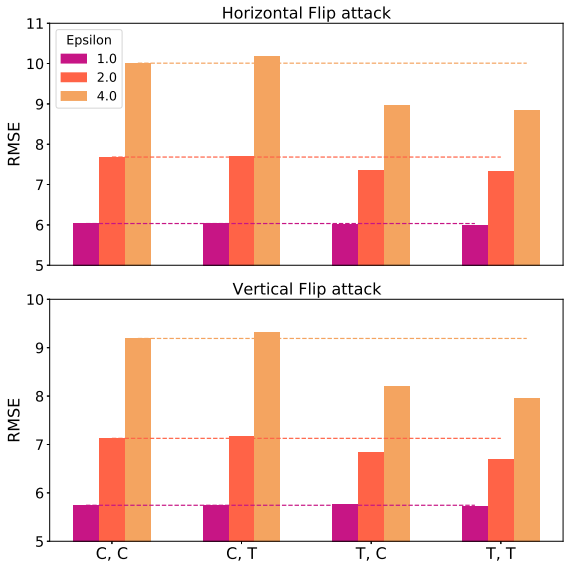

Figure 5: RMSE for adversarial corruptions of KITTI generated using horizontal and vertical flip attacks for all four combinations of depth and ego-motion networks.

Figure 3 demonstrates a significant improvement in the robustness to all the natural corruptions when learning depth with transformers instead of convolutional networks. Figure 4 further shows a general improvement in adversarial robustness when learning either depth or ego-motion with transformers instead of convolutional networks. Finally, Figure 5 shows an improvement in robustness to symmetrically flipped adversarial attacks when depth is learned using transformers instead of convolutional networks. Furthermore, depth estimation is most robust when both depth and ego-motion are learned using transformers.

Therefore, MT-SfMLearner, where both depth and ego-motion and learned with transformers, provides the highest robustness and generalizability, in line with the studies on image classification (Paul and Chen, 2021; Bhojanapalli et al., 2021). This can be attributed to their global receptive field, which allows for better adjustment to the localized deviations by accounting for the global context of the scene.

4.4 Intrinsics

Here, we evaluate the accuracy of our proposed method for self-supervised learning of camera intrinsics and its impact on the depth estimation performance. As shown in Table 6, the percentage error for intrinsics estimation is low for both convolutional and transformer-based methods trained on an image size of 640×192. Moreover, the depth *error* as well *accuracy* metrics are both similar to when the ground truth intrinsics are known a priori. This is also observed in Figure 6 where the learning of intrinsics causes no artifacts in depth estimation.

We also evaluate the models trained with learned intrinsics on all 15 natural corruptions as well as on PGD and symmetrically flipped adversarial examples. We report the mean RMSE (μRMSE) across all corruptions in Table 7. The RMSE for depth estimation on adversarial examples generated by PGD method for all strengths is shown in Figure 7. The mean RMSE (μRMSE) across all attack strengths for horizontally flipped and vertically flipped adversarial ex-

Table 6: Percentage error for intrinsics prediction and impact on depth estimation for KITTI Eigen split.

Network	Intrinsics	Intrinsics Error(%) ↓				Depth Error↓				Depth Accuracy↑		
		f_x	c_x	f_y	c_y	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
C,C	Given	-	-	-	-	0.111	0.897	4.865	0.193	0.881	0.959	0.980
	Learned	-1.889	-2.332	2.400	-9.372	0.113	0.881	4.829	0.193	0.879	0.960	0.981
T,T	Given	-	-	-	-	0.112	0.838	4.771	0.188	0.879	0.960	0.982
	Learned	-1.943	-0.444	3.613	-16.204	0.112	0.809	4.734	0.188	0.878	0.960	0.982

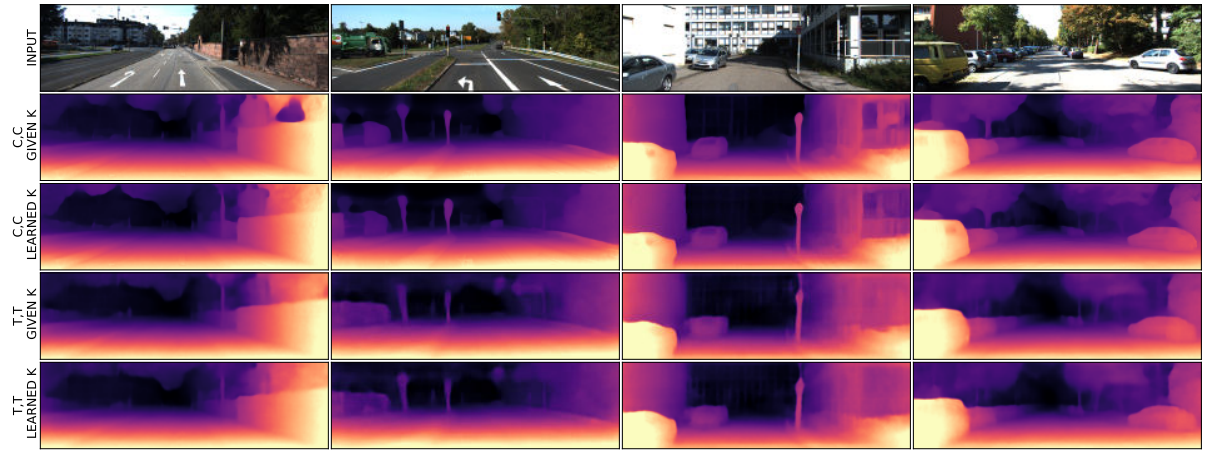

Figure 6: Disparity maps for qualitative comparison on KITTI, when trained with and without intrinsics (K). The second and fourth rows are same as the second and the fifth rows in Figure 2.

Table 7: Mean RMSE (μRMSE) for natural corruptions of KITTI, when trained with and without ground-truth intrinsics.

Architecture	Intrinsics	μRMSE↓
C, C	Given	7.683
	Learned	7.714
T, T	Given	5.918
	Learned	5.939

Table 8: Mean RMSE (μRMSE) for horizontal (H) and vertical (V) adversarial flips of KITTI, when trained with and without ground-truth intrinsics.

Architecture	Intrinsics	μRMSE↓ (H)	μRMSE↓ (V)
C, C	Given	7.909	7.354
	Learned	7.641	7.196
T, T	Given	7.386	6.795
	Learned	7.491	6.929

amples is shown in Table 8.

We observe that the robustness to natural corruptions and adversarial attacks is maintained by both architectures when the intrinsics are learned simultaneously. Furthermore, similar to the scenario with known ground truth intrinsics, MT-SfMLearner with learned intrinsics has higher robustness than its convolutional counterpart.

4.5 Efficiency

We further compare the networks on their computational and energy efficiency to examine their suitability for real-time applications.

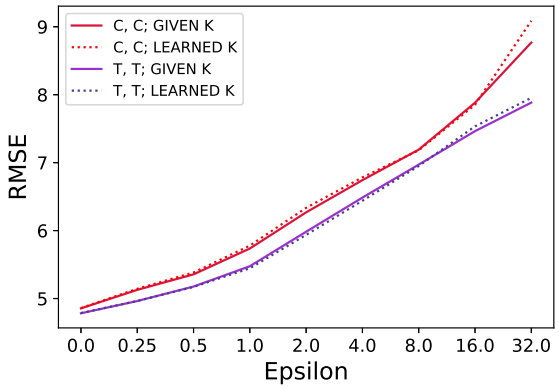

Figure 7: Mean RMSE for adversarial corruptions of KITTI generated using PGD, when trained with and without ground-truth intrinsics (K).

In Table 9, we report the mean inference speed in frames per second (fps) and the mean inference energy consumption in Joules per frame for depth and intrinsics estimation for both architectures. These metrics are computed over $10,000$ forward passes at a resolution of 640×192 on an NVidia GeForce RTX 2080 Ti.

Both architectures run depth and intrinsics estimation in real-time with an inference speed > 30 fps. However, the transformer-based method consumes higher energy and is computationally more de-

Table 9: Inference Speed (frames per second) and Energy Consumption (Joules per frame) for depth and intrinsics estimation using CNN- and transformer-based architectures.

Architecture	Estimate	Speed↑	Energy↓
C,C	Depth	84.132	3.206
	Intrinsics	97.498	2.908
T,T	Depth	40.215	5.999
	Intrinsics	60.190	4.021

manding than its convolutional counterpart.

5 CONCLUSION

This work is the first to investigate the impact of transformer architecture on the SfM inspired self-supervised monocular depth estimation. Our transformer-based method MT-SfMLearner performs comparably against contemporary convolutional methods on the KITTI depth prediction benchmark. Our contrastive study additionally demonstrates that while CNNs provide local spatial bias, especially for thinner objects and around boundaries, transformers predict uniform and coherent depths, especially for larger objects due to their global receptive field. We observe that transformers in the depth network result in higher robustness to natural corruptions, and transformers in both depth, as well as the ego-motion network, result in higher robustness to adversarial attacks. With our proposed approach to self-supervised camera intrinsics estimation, we also demonstrate how the above conclusions hold even when the focal length and principal point are learned along with depth and ego-motion. However, transformers are computationally more demanding and have lower energy efficiency than their convolutional counterparts. Thus, we contend that this work assists in evaluating the trade-off between performance, robustness, and efficiency of self-supervised monocular depth estimation for selecting the suitable architecture.

REFERENCES

Aich, S., Vianney, J. M. U., Islam, M. A., Kaur, M., and Liu, B. (2021). Bidirectional attention network for monocular depth estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., and Veit, A. (2021). Understanding robustness of transformers for image classification. *arXiv preprint arXiv:2103.14586*.

Bian, J., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M.-M., and Reid, I. (2019). Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *Advances in Neural Information Processing Systems*, pages 35–45.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

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

Chawla, H., Jukola, M., Brouns, T., Arani, E., and Zonooz, B. (2020). Crowdsourced 3d mapping: A combined multi-view geometry and self-supervised learning approach. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4750–4757. IEEE.

Chawla, H., Varma, A., Arani, E., and Zonooz, B. (2021). Multimodal scale consistency and awareness for monocular self-supervised depth estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Diaz, R. and Marathe, A. (2019). Soft labels for ordinal regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4738–4747.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.

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

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

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., and Wichmann, F. A. (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

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

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

Goldman, M., Hassner, T., and Avidan, S. (2019). Learn stereo, infer mono: Siamese networks for self-supervised, monocular, depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0.

Gordon, A., Li, H., Jonschkowski, R., and Angelova, A. (2019). Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras.

Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., and Gaidon, A. (2020). 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494.

Guo, X., Li, H., Yi, S., Ren, J., and Wang, X. (2018). Learning monocular depth by distilling cross-domain stereo networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–500.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hendrycks, D. and Dietterich, T. (2019). Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*.

Jiang, H. and Huang, R. (2019). Hierarchical binary classification for monocular depth estimation. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1975–1980. IEEE.

Johnston, A. and Carneiro, G. (2020). Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 4756–4765.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klingner, M., Termöhlen, J.-A., Mikolajczyk, J., and Fingscheidt, T. (2020). Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance. In *ECCV*.

Kong, S. and Fowlkes, C. (2019). Pixel-wise attentional gating for scene parsing. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1024–1033. IEEE.

Kurakin, A., Goodfellow, I., Bengio, S., et al. (2016). Adversarial examples in the physical world.

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

Li, B., Dai, Y., and He, M. (2018a). Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *Pattern Recognition*, 83:328–339.

Li, R., Xian, K., Shen, C., Cao, Z., Lu, H., and Hang, L. (2018b). Deep attention-based classification network for robust depth prediction. In *Asian Conference on Computer Vision (ACCV)*.

Li, Z., Liu, X., Drenkow, N., Ding, A., Creighton, F. X., Taylor, R. H., and Unberath, M. (2020). Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2011.02910*.

Liebel, L. and Körner, M. (2019). Multidepth: Single-image depth estimation via multi-task regression and classification. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1440–1447. IEEE.

Lin, G., Milan, A., Shen, C., and Reid, I. (2017). Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*.

Lopez, M., Mari, R., Gargallo, P., Kuang, Y., Gonzalez-Jimenez, J., and Haro, G. (2019). Deep single image camera calibration with radial distortion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11817–11825.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lyu, X., Liu, L., Wang, M., Kong, X., Liu, L., Liu, Y., Chen, X., and Yuan, Y. (2020). Hr-depth: high resolution self-supervised monocular depth estimation. *CoRR abs/2012.07356*.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Mahjourian, R., Wicke, M., and Angelova, A. (2018). 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*, pages 5667–5675.

Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A. S., Bethge, M., and Brendel, W. (2019). Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*.

Ochs, M., Kretz, A., and Mester, R. (2019). Sdnet: Semantically guided depth estimation network. In *German Conference on Pattern Recognition*, pages 288–302. Springer.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

Paul, S. and Chen, P.-Y. (2021). Vision transformers are robust learners. *arXiv preprint arXiv:2105.07581*.

Poggi, M., Aleotti, F., Tosi, F., and Mattoccia, S. (2020). On the uncertainty of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3227–3237.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A. (2021). Do vision transformers see like convolutional neural networks? *arXiv preprint arXiv:2108.08810*.

Ranftl, R., Bochkovskiy, A., and Koltun, V. (2021). Vision transformers for dense prediction. *ArXiv preprint*.

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

Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., and Black, M. J. (2019). Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 12240–12249.

Ren, H., El-Khamy, M., and Lee, J. (2019). Deep robust single image depth estimation neural network using scene understanding. In *CVPR Workshops*, pages 37–45.

Roussel, T., Van Eycken, L., and Tuytelaars, T. (2019). Monocular depth estimation in new environments with absolute scale. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1735–1741. IEEE.

Strudel, R., Garcia, R., Laptev, I., and Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. *arXiv preprint arXiv:2105.05633*.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR.

Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., and Geiger, A. (2017). Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*, pages 11–20. IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wong, A., Cicek, S., and Soatto, S. (2020). Targeted adversarial perturbations for monocular depth prediction. In *Advances in neural information processing systems*.

Xiang, X., Kong, X., Qiu, Y., Zhang, K., and Lv, N. (2021). Self-supervised monocular trained depth estimation using triplet attention and funnel activation. *Neural Processing Letters*, pages 1–18.

Yang, G., Tang, H., Ding, M., Sebe, N., and Ricci, E. (2021). Transformer-based attention networks for continuous pixel-wise prediction. In *ICCV*.

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

Yin, Z. and Shi, J. (2018). Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992.

Zhang, Z., Cui, Z., Xu, C., Yan, Y., Sebe, N., and Yang, J. (2019). Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4106–4115.

Zhang, Z., Xu, C., Yang, J., Tai, Y., and Chen, L. (2018). Deep hierarchical guidance and regularization learning for end-to-end depth estimation. *Pattern Recognition*, 83:430–442.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., et al. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890.

Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video.

Zhuang, B., Tran, Q.-H., Lee, G. H., Cheong, L. F., and Chandraker, M. (2019). Degeneracy in self-calibration revisited and a deep learning solution for uncalibrated slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3766–3773. IEEE.