Transformers in Unsupervised Structure-from-Motion

Hemang Chawla$^{*1,2[0000-0002-5999-6901]}$, Arnav Varma$^{*1[0000-0002-5919-0449]}$,
Elahe Arani$^{1,2[0000-0002-0952-7007]}$, and Bahram Zonooz$^{1,2[0000-0003-4124-3394]}$

[1] Advanced Research Lab, NavInfo Europe, Eindhoven, The Netherlands
[2] Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands
{h.chawla, e.arani, b.zonooz}@tue.nl
https://www.navinfo.eu/expertise/artificial-intelligence/

Abstract. Transformers have revolutionized deep learning based computer vision with improved performance as well as robustness to natural corruptions and adversarial attacks. Transformers are used predominantly for 2D vision tasks, including image classification, semantic segmentation, and object detection. However, robots and advanced driver assistance systems also require 3D scene understanding for decision making by extracting structure-from-motion (SfM). We propose a robust transformer-based monocular SfM method that learns to predict monocular pixel-wise depth, ego vehicle's translation and rotation, as well as camera's focal length and principal point, simultaneously. With experiments on KITTI and DDAD datasets, we demonstrate how to adapt different vision transformers and compare them against contemporary CNN-based methods. Our study shows that transformer-based architecture, though lower in run-time efficiency, achieves comparable performance while being more robust against natural corruptions, as well as untargeted and targeted attacks.[§]

Keywords: structure-from-motion, monocular depth estimation, monocular pose estimation, camera calibration, natural corruptions, adversarial attacks

1 Introduction

Scene understanding tasks have benefited immensely from advances in deep learning over the years [58]. Several existing methods in computer vision for robotics [33], augmented reality [25], and autonomous driving [43] have been using convolutional neural networks (CNNs) with its properties of spatial locality and translation invariance [31] resulting in excellent performance. With the advent of vision transformers [11, 54], models with the ability to learn from the global context have even outperformed CNNs for some tasks such as object detection [5] and semantic segmentation [59]. Nevertheless, despite performance improvements in curated test sets, safe deployment of models requires further consideration of robustness and generalizability.

Different neural network architectures have been shown to have different effects on model performance, robustness, and generalizability on different tasks [9, 21]. CNNs

[*] Equal contribution.

[§] Code: https://github.com/NeurAI-Lab/MT-SfMLearner have localized linear operations and lose feature resolution during downsampling to increase their limited receptive field [57]. Transformers, with their different layers that simultaneously attend more to global features with no inductive bias in favor of locality lead to more globally coherent predictions [52]. Among these architectures, transformers have been found to be more robust for tasks such as classification [2, 45], object detection, and semantic segmentation [22] despite requiring more training data and being more computationally expensive [6]. Although several studies have compared their performance on 2D vision tasks [22], studies that evaluate their performance on 3D scene understanding tasks such as monocular Structure-from-Motion (SfM) are lacking.

SfM is a prominent problem in 3D computer vision in which the 3D structure is reconstructed by simultaneously estimating scene depth, camera poses, and intrinsic parameters. Traditional methods for SfM rely on correspondence search followed by incremental reconstruction of the environment, and are able to handle a diverse set of scenes [50]. Instead, most deep learning based methods for SfM primarily focus on depth estimation with pose and intrinsics estimation as auxiliary tasks [18, 37]. Deep learning methods have increasingly been employed, either in replacement or together with traditional methods, to deal with issues of low-texture, thin structures, misregistration, failed image registration, etc. [60]. While most of the existing works for depth estimation were based on CNNs [16, 18, 37, 40], transformer-based approaches to supervised depth estimation have also been proposed [47]. However, unsupervised methods that do not require ground truth collected from costly LiDARs or RGB-D setups are often favored, as they can potentially be applied to innumerable data. Methods that utilize transformer ingredients [24] such as attention have been proposed to improve depth estimation, but transformer encoders have scarcely been adopted for depth and pose estimation.

In this work, we perform a comparative analysis between CNN- and transformer-based architectures for unsupervised depth estimation. We show how vision transformers can be adapted for *unsupervised* depth estimation with our method Monocular Transformer SfMLearner (MT-SfMLearner). We evaluate how the architecture choices for individual depth and pose estimation networks impact the depth estimation performance, as well as robustness to natural corruptions and adversarial attacks. Since SfM depends upon the knowledge of camera intrinsics, we also introduce a modular approach to predict the camera focal lengths and principal point from the input images, which can be utilized within both CNN- and transformer-based architectures. We also study the accuracy of intrinsics and pose estimation, including the impact of learning camera intrinsics on depth and pose estimation. Finally, we compare the computational and energy efficiency of the architectures for depth, pose, and intrinsics estimation.

This work is an extended version of our study comparing transformers and CNNs for unsupervised depth estimation [53]. While the previous work demonstrated how vision transformers that were built for classification-like tasks can be also used for unsupervised depth estimation, here we demonstrate that our method also extends to other transformer architectures. We also perform an additional evaluation on a more challenging dataset to further substantiate the generalizability of our study. Additionally, we also compare the impact of architectures on the performance of the auxiliary pose prediction task, including when the camera instrincs are learned simultaneously. With a more general purpose method, additional experiments, quantitative results, and visualizations, this work presents a way to compare the trade-off between the performance, robustness, and efficiency of transformer- and CNN-based architectures for monocular unsupervised depth estimation.

2 Related Works

The simultaneous estimation of Structure-from-Motion (SfM) is a well-studied problem with an established toolchain of techniques [50]. Although the traditional toolchain is effective and efficient in many cases, its reliance on accurate image correspondence can cause problems in areas of low texture, complex geometry/photometry, thin structures, and occlusions [60]. To address these issues, several of the pipeline stages have recently been tackled using deep learning, e.g., feature matching [23], pose estimation [3], and stereo and monocular depth estimation [16]. Of these, unsupervised monocular depth estimation in particular has been extensively explored with CNN-based depth and pose networks, with pose estimation as an auxiliary task [1, 16, 18, 37, 60]. These learning-based techniques are attractive because they can utilize external supervision during training and may circumvent the aforementioned issues when applied to test data. However, learned systems might not be robust to shifts in distribution during test time.

Recently, transformer-based architectures [11, 52], which outperform CNN-based architectures in image classification, have been proven to be more robust in image classification [2, 42], as well as dense prediction tasks such as object detection and semantic segmentation [22]. Motivated by their success, researchers have replaced CNN encoders with transformers in scene understanding tasks such as object detection [5, 35], semantic segmentation [51, 59], and supervised monocular depth estimation [47, 57]. Our previous work [53] and MonoFormer [1] further extend transformer-based architectures to unsupervised monocular unsupervised depth estimation. However, ours is the only work that comprehensively demonstrates the robustness of transformer-based architectures for unsupervised SfM. We now provide analyses to establish generalizability of our approach across multiple datasets, transformer-based architectures, and auxiliary SfM tasks such as pose estimation and intrinsics estimation.

3 Method

We study the impact of using vision transformer based architectures for unsupervised monocular Structure-from-Motion (SfM) in contrast to contemporary methods that utilize CNN-based architectures.

3.1 Monocular Unsupervised SfM

For training unsupervised SfM networks, we utilize videos captured from monocular cameras. Given a video sequence with n images, both the depth and pose estimation networks are trained simultaneously. This is unlike supervised networks, where depth

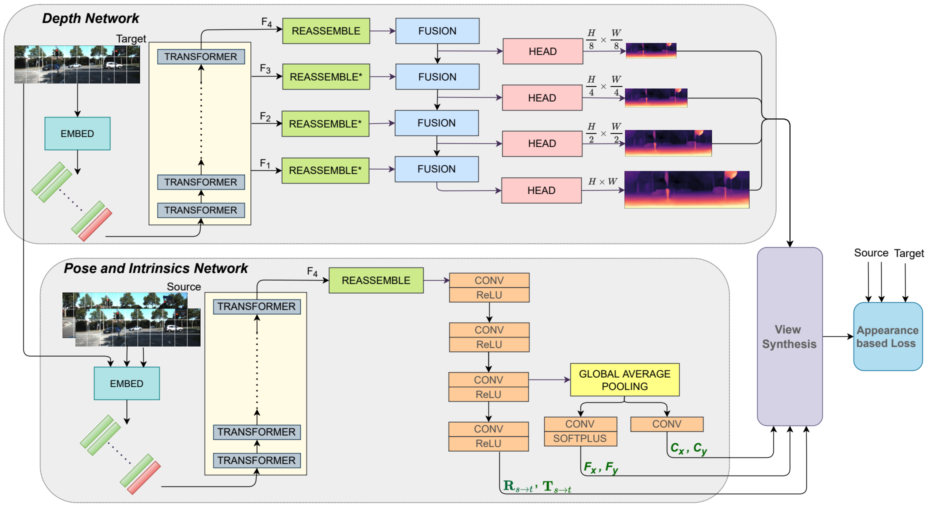


Fig. 1. An overview of Monocular Transformer Structure from Motion Learner (MT-SfMLearner) with learned intrinsics. We readapt modules from Dense Prediction Transformer (DPT) and Monodepth2 to be trained with appearance-based losses for unsupervised monocular depth, pose, and intrinsics estimation. * refers to optional modules that may not be present in all transformer-based architectures. Figure modified from [53].

or pose estimation may be trained independently. The input to the depth estimation network $f_D : \mathbb{R}^{H\times W\times 3}$ is a single image, for which it outputs pixel-wise depth in $\mathbb{R}^{H\times W}$. The input to the pose estimation network $f_E : \mathbb{R}^{H\times W\times 6}$ is a pair of images, for which it outputs the relative translation (t_x, t_y, t_z) and rotation $(r_x, r_y, r_z) \in \mathbb{R}^6$, which is used to form the affine transformation $\left[\begin{smallmatrix}\hat{R} & \hat{T}\\ 0 & 1\end{smallmatrix}\right] \in \mathrm{SE}(3)$. To train both networks simultaneously, a batch consists of triplets of temporally consecutive RGB images $\{I_{-1}, I_0, I_1\} \in \mathbb{R}^{H\times W\times 3}$. While I_0 is input into the depth estimation network, $\{I_{-1}, I_0\}$ and $\{I_0, I_1\}$ are input into the pose estimation network to predict the next and previous relative pose. The perspective projection model links together the predicted depth $\hat{D}$ and pose $\hat{T}$ such that,

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

This is used to warp the source images $I_s \in \{I_{-1}, I_1\}$ to the target image $I_t \in \{I_0\}$ as part of the view synthesis (see Figure 1), where K represents the camera intrinsics. For each triplet, two target images $\hat{I}_0$ are synthesized, which are compared with the real target image, to compute the appearance-based *photometric* loss. Additionally, we utilize a smoothness loss [18] on the predicted depth for regularization.

3.2 Architecture

Depth Network For the depth network, we use a transformer-based architecture in the encoder, and readapt the decoder from the DPT [46]. There are five components of the depth network:

Table 1. Architecture details of the *Reassemble* modules. *DN* and *PN* refer to depth and pose networks, respectively. The subscripts of *DN* refer to the transformer stage from which the respective *Reassemble* module takes its input (see Figure 1). The 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 in the image, s refers to the stride of the *Embed* module and d refers to the feature dimension of the transformer features. Table modified from [53].

Encoder	**Operation**	**Input size**	**Output size**	**Function**	**Parameters** (DN_1, DN_2, DN_3, DN_4, PN_4)
DeiT	Read	$(N_p + 1) \times d$	$N_p \times d$	Drop readout token	$-$
	Concatenate	$N_p \times d$	$d \times \frac{H}{p} \times \frac{W}{p}$	Transpose & Unflatten	$-$
	Pointwise Convolution	$d \times \frac{H}{p} \times \frac{W}{p}$	$N_c \times \frac{H}{p} \times \frac{W}{p}$	Change to N_c channels	$N_c = [96, 768, 1536, 3072, 2048]$
	Strided Convolution	$N_c \times \frac{H}{p} \times \frac{W}{p}$	$N_c \times \frac{H}{2p} \times \frac{W}{2p}$	$k \times k$ convolution, stride$=2$, N_c channels, padding$=1$	$k = [-, -, -, 3, -]$
	Transpose Convolution	$N_c \times \frac{H}{p} \times \frac{W}{p}$	$N_c \times \frac{H}{\alpha} \times \frac{W}{\alpha}$	$\frac{p}{\alpha} \times \frac{p}{\alpha}$ deconvolution, stride$= \frac{p}{\alpha}$, N_c channels	$\alpha = [4, 8, -, -, -]$
PVT	Reshape	$\frac{HW}{64s^2} \times N_b$	$N_b \times \frac{H}{8s} \times \frac{W}{8s}$	Reshape token to image-like $2D$ representations.	$N_b = [-, -, -, 512, 512]$

- ***Embed*** module, which is part of the encoder, takes an image $I \in \mathbb{R}^{H \times W \times 3}$ and converts image patches of size $p \times p$ to $N_p = H \cdot W / p^2$ tokens $t_i \in \mathbb{R}^d$ $\forall i \in [1, 2, ... N_p]$. This is implemented as a $p \times p$ convolution with stride $s \leq p$. The output of this module may be concatenated with a *readout* token $\in \mathbb{R}^d$, depending on the transformer based architecture.
- ***Transformer*** block, which is also part of the encoder, consists of multiple transformer stages that process these tokens with self-attention modules [54]. Self-attention processes inputs at constant resolution and can simultaneously attend to global and local features.
- ***Reassemble*** modules in the decoder are responsible for extracting image-like $2D$ representations from the features of the transformer block. At least one reassemble module is used, and additional modules may be used depending on the transformer-based architecture. The exact details of the *Reassemble* modules can be found in Table 1.
- ***Fusion*** modules in the decoder, based on RefineNet [34], are responsible for progressively fusing the features of the encoder or the *Reassemble* modules with the features of the decoder. The module is also responsible for upsampling the features

Table 2. Architecture details of *Head* modules in Figure 1. Source: [53].

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

by 2 at each stage. Unlike DPT, we enable batch normalization in the decoder, as it was found to be helpful for unsupervised depth prediction. We also reduce the number of channels in the *Fusion* block to 96 from 256 in DPT.

- ***Head*** modules after each *Fusion* module predict depth on four scales, according to previous unsupervised methods [17]. Unlike DPT, *Head* modules use 2 convolutions instead of 3 as we did not find any difference in performance. For the exact architecture of the *Head* modules, see Table 2.

Pose Network. For the pose network, we adopt an architecture similar to that of the depth network, with a transformer-based architecture in the encoder, but the decoder from Monodepth2 [17]. Since the input to the transformer for the pose network consists of two images concatenated along the channel dimension, we repeat the *Embed* module accordingly. Unlike the depth network, we only use a single *Reassemble* module to pass transformer tokens to the decoder, independently of the transformer-based architecture used. For details of the structure of this Reassemble module, refer to Table 1.

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

3.3 Intrinsics

As seen in Equation 1, unsupervised monocular SfM requires knowledge of the ground truth camera intrinsics. Intrinsics are given by

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

where f_x and f_y refer to the focal lengths of the camera along the x-axis and y-axis, respectively. c_x and c_y refer to x and y coordinates of the principal point in the pinhole camera model. Most unsupervised SfM methods are trained with intrinsics known a priori. 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.

Therefore, we introduce an intrinsics estimation module. We modify the pose network to additionally estimate the focal lengths and principal point along with the translation and rotation. Concretely, we add a convolutional path in the pose decoder to learn the intrinsics. The features before activation from the penultimate decoder layer are passed through a global average pooling layer. This is followed by two branches of pointwise convolutions that reduce the number of channels from 256 to 2. One branch uses softplus activation to estimate focal lengths along the x and y axes, as the focal length is always positive. The other branch estimates the principal point without employing any activation, as the principal point does not have such a constraint. Note that the pose decoder is the same for both CNN- and transformer-based architectures. Consequently, the intrinsics estimation method can be modularly utilized with both architectures. Figure 1 demonstrates MT-SfMLearner with learned intrinsics.

3.4 Appearance-based Losses

Following contemporary unsupervised monocular SfM methods, we adopt the *appearance-based losses* and an *auto-masking* procedure from CNN-based Monodepth2 [17] for the transformer-based architecture described above. We employ a photometric per-pixel minimum reprojection loss composed of the pixel-wise ℓ_1 distance as well as the Structural Similarity (SSIM) between the real and synthesized target images, along with a multiscale edge-aware *smoothness* loss on the depth predictions. We also use auto-masking to disregard the temporally stationary pixels in the image triplets. Finally, to reduce texture-copy artifacts, we calculate the total loss after upsampling the depth maps, predicted at 4 scales by the decoder, to the input resolution.

4 Experiments

We compare the CNN and transformer architectures for their impact on unsupervised monocular depth and pose estimation, including when the camera intrinsics are unknown and when they are estimated simultaneously.

4.1 Datasets

KITTI. For depth estimation, we report results on the Eigen Split [12] of KITTI [13] dataset after removing the static frames as per [60], unless stated otherwise. This split contains $39,810$ training images, 4424 validation images, and 697 test images, respectively. This dataset captures scenes from rural, city and highway areas around Karlsruhe, Germany. All results are reported on the per-image scaled dense depth prediction without post-processing [17] for an image size of 640×192, unless otherwise stated.

For pose estimation in Section 4.6, we report results on the Odom Split [60] of the KITTI dataset for an image size of 640×192, unless stated otherwise. This split contains 8 training sequences (sequences $00-02$, $04-08$) and two test sequences $(09, 10)$.

Dense Depth for Autonomous Driving. For depth estimation, we also report results on the Dense Depth for Autonomous Driving (DDAD) dataset [18], for an image size of $640\times$ 384 unless otherwise noted. It contains 12650 training samples from 150 sequences and 3950 test samples from 50 sequences, respectively. This dataset contains samples with long range depth (up to 250 m) from a diverse set of urban scenarios in multiple cities of the United States (Ann Arbor, Bay Area, Cambridge, Detroit, San Francisco) and Japan (Odaiba, Tokyo).

4.2 Architecture

For the transformer-based architecture in our depth and pose encoders, we use DeiT-base [52] except in Section 4.6, where we use PVT-b4 [55] to demonstrate that our approach generalizes to other transformer-based architectures. The *Embed* module of DeiT-base has a patch size $p = 16$ and stride $s = 16$, while that of PVT-b4 has a patch size $p = 7$ and a stride $s = 4$. DeiT-base employs 12 *Transformer* stages with features F_1, F_2, F_3, and F_4 (see Figure 1) taken from the 3^{rd}, 6^{th}, 9^{th}, and final stages to be sent to the decoder. PVT-b4, meanwhile, employs 4 transformer stages, each of which contributes to the features sent to the decoder. Finally, DeiT-base uses 4 *Reassemble* modules in the depth encoder, while PVT-b4 uses only one *Reassemble module* in the depth encoder. The exact architecture of the Reassemble modules can be found in Table 1.

4.3 Implementation Details

The networks are implemented in PyTorch [41] and trained on a TeslaV100 GPU for 20 epochs at a resolution of 640×192 with batch sizes 12 for DeiT-base encoder and 8 for PVT-b4 encoder, unless otherwise mentioned. The depth and pose encoders are initialized with ImageNet [10] pre-trained weights. We use the Adam [26] optimizer for CNN-based networks and AdamW [36] 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 $\beta_2 = 0.999$.

4.4 Evaluation Metrics

Depth Estimation. We measure the error and accuracy of depth estimation using various metrics. For error, we use the absolute relative error (Abs Rel) [49], squared relative error [28] (Sq Rel), linear root mean squared error (RMSE) [32], log scale invariant RMSE [12] (RMSE log). For accuracy, we measure under three thresholds, reported as ratios [30] ($\delta < 1.25$, $\delta < 1.25^2$, $\delta < 1.25^3$).

Pose Estimation. We measure translation and rotational errors for all possible subsequences of length $(100, 200, \ldots, 800)m$, and report the average of the values. The translation error is reported as a percentage and the rotation error as degrees per $100m$ [14].

Intrinsics Estimation. We measure the percentage error from its ground truth value for each camera intrinsic parameter.

Efficiency. We measure computational and energy efficiency using frames per second (fps) and Joules per frame, respectively.

Table 3. Quantitative results on KITTI Eigen split for all four architecture combinations of depth and pose networks. The best results are displayed in bold, and the second best are underlined. Source: [53].

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**

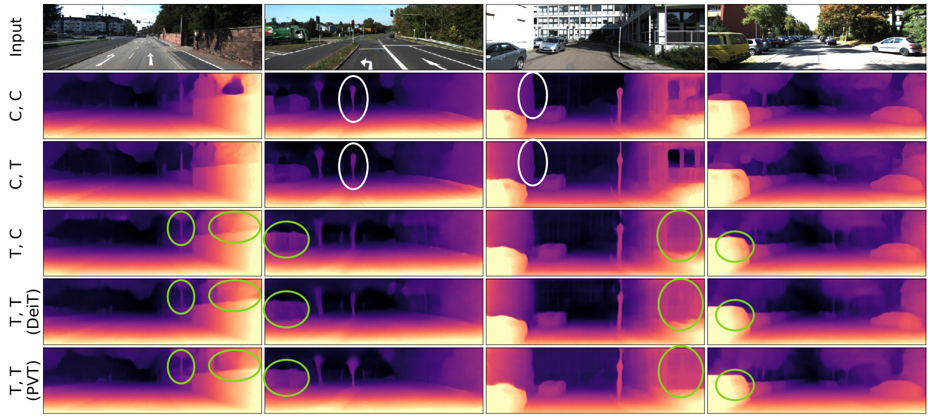
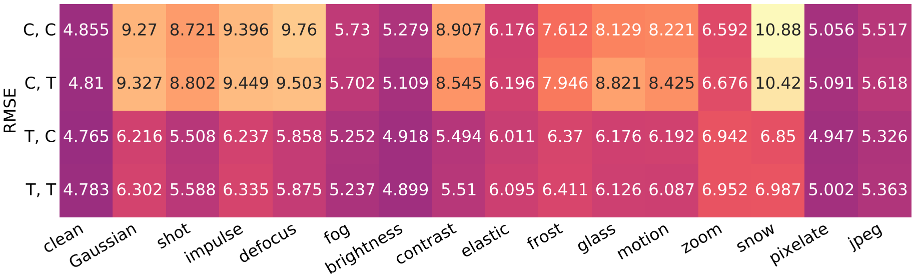


Fig. 2. Disparity maps on KITTI Eigen for qualitative comparison of all four architecture combinations of depth and pose networks. Example regions where the global receptive field of transformers is advantageous are highlighted in green. Example areas where local receptive field of CNNs is advantageous are highlighted in white. Source: [53].

4.5 Impact of Architecture

Since unsupervised monocular depth estimation networks simultaneously train a pose network (see Equation 1), we investigate the impact of each network's architecture on depth estimation. We consider CNN-based (C) and Transformer-based (T) networks for depth and pose estimation. The four resulting combinations of (Depth Network, Pose Network) architectures, in ascending order of impact of transformers on depth estimation, are (C, C), (C, T), (T, C), and (T, T). To compare the transformer-based architecture fairly with CNN-based networks, we utilize Monodepth2 [17] with ResNet-101 [19] in the depth and pose encoders. All four combinations are trained thrice on the KITTI Eigen split using the settings described in Section 4.3 and the known ground-truth camera intrinsics.

On Performance. Table 3 shows the best results on depth estimation for each architecture combination of depth and pose networks. We observe that MT-SfMLearner, i.e. the combination of transformer-based depth and pose networks, performs best under two of



RMSE	clean	Gaussian	shot	impulse	defocus	fog	brightness	contrast	elastic	frost	glass	motion	zoom	snow	pixelate	jpeg
C, C	4.855	9.27	8.721	9.396	9.76	5.73	5.279	8.907	6.176	7.612	8.129	8.221	6.592	10.88	5.056	5.517
C, T	4.81	9.327	8.802	9.449	9.503	5.702	5.109	8.545	6.196	7.946	8.821	8.425	6.676	10.42	5.091	5.618
T, C	4.765	6.216	5.508	6.237	5.858	5.252	4.918	5.494	6.011	6.37	6.176	6.192	6.942	6.85	4.947	5.326
T, T	4.783	6.302	5.588	6.335	5.875	5.237	4.899	5.51	6.095	6.411	6.126	6.087	6.952	6.987	5.002	5.363

Fig. 3. RMSE for natural corruptions of KITTI Eigen test set for all four combinations of depth and pose networks. The i.i.d evaluation is denoted by *clean*. Source: [53].

the *error* metrics and two of the *accuracy* metrics. The remaining combinations show comparable performance on all metrics. Figure 2 also shows more uniform estimates for larger objects, such as vehicles, vegetation, and buildings, when the depth is learned using transformers. Transformers also estimate depth more coherently for reflections 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 due to the inherent inductive bias for spatial locality present in convolutional layers.

On Robustness. We saw in the previous section that the different architecture combinations perform comparably on the independent and identically distributed (i.i.d) test set. However, networks that perform well on an i.i.d test set may still learn shortcut features that reduce robustness on out-of-distribution (o.o.d) datasets [15]. Therefore, we study the robustness of each architecture combination. We report the mean RMSE across three training runs on the KITTI Eigen split test set for all experiments in this section.

Natural Corruptions: Following [20] and [39], we generate 15 corrupted versions of the KITTI i.i.d test set at the highest severity($= 5$). These corruptions are changes to images that correspond to variations expected in nature, such as those due to *noise* (Gaussian, shot, impulse), *blur* (defocus, glass, motion, zoom), *weather* (snow, frost, fog, brightness) and *digital* (contrast, elastic, pixelate, JPEG). We observe in Figure 3 that learning depth with transformer-based architectures instead of CNN-based architecture leads to a significant improvement in the robustness to all natural corruptions.

Untargeted Adversarial Attack: Untargeted adversarial attacks make changes to input images that are imperceptible to humans to generate adversarial examples that can induce general prediction errors in neural networks. We employ Projected Gradient Descent (PGD) [38] to generate untargeted adversarial examples from the test set at attack strength $\varepsilon \in \{0.25, 0.5, 1.0, 2.0, 4.0, 8.0, 16.0\}$. The gradients are calculated with respect to the appearance-based training loss. Following [29], the adversarial perturbation is

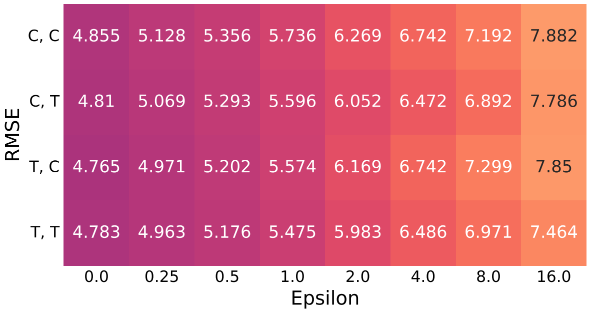


Fig. 4. RMSE for adversarial corruptions of KITTI Eigen test set generated using the PGD attack at all attack strengths (0.0 to 16.0) for the four combinations of depth and pose networks. Attack strength 0.0 refers to i.i.d evaluation. Source: [53].

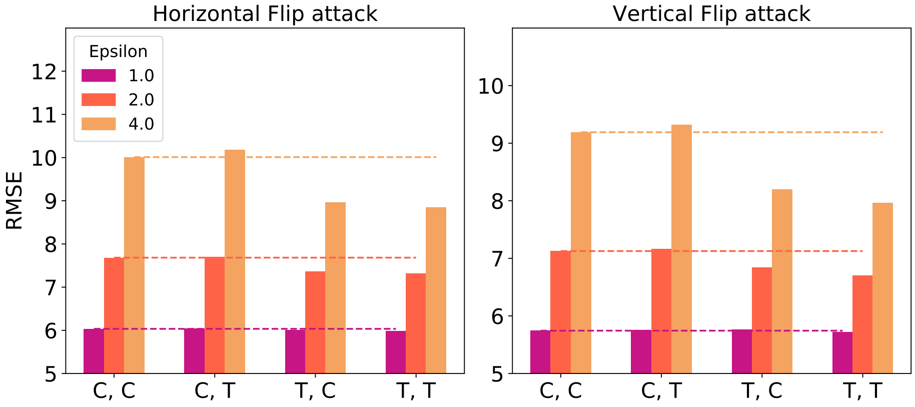


Fig. 5. RMSE for adversarial corruptions of KITTI Eigen test set generated using targeted horizontal and vertical flip attacks for all four combinations of depth and pose networks. Source: [53].

computed 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 sequence, the training loss is only calculated for the feasible pair of images. Figure 4 demonstrates a general improvement in untargeted adversarial robustness when learning depth or pose with a transformer-based architecture instead of a CNN-based architecture.

Targeted Adversarial Attack: Finally, targeted adversarial attacks make changes imperceptible for humans to generate adversarial examples that can induce *specific* prediction errors in neural networks. Deriving from [56], we generate targeted adversarial examples to fool the networks into predicting horizontally and vertically flipped estimates. To this end, we use the gradients with respect to the RMSE loss, where the targets are symmetrical horizontal and vertical flips of the predictions on the clean test set images. This evaluation is conducted at attack strength $\varepsilon \in \{1.0, 2.0, 4.0\}$. Figure 5 shows an

Table 4. Quantitative results on DDAD (complete). The best results are shown in bold.

Architecture	**Error↓**				**Accuracy↑**		
	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Packnet [18]	0.178	7.521	14.605	0.254	0.831	0.928	0.963
C, C	**0.151**	**3.346**	14.229	0.243	0.814	0.929	0.967
T,T	**0.151**	3.821	**14.162**	**0.237**	**0.820**	**0.935**	**0.970**

improvement in robustness to targeted adversarial attacks when depth is learned using transformer-based architectures instead of CNN-based architectures. Furthermore, the combination where both depth and pose are learned using transformer-based architectures is the most robust.

Therefore, MT-SfMLearner, where depth and pose are learned with transformer-based architectures, provides the highest robustness against natural corruptions and untargeted and targeted adversarial attacks, according to studies on image classification [2, 42]. 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.

4.6 Generalizability

The previous section showed that the use of transformers in the depth and pose estimation network contributes to the improved performance and robustness of depth estimation compared to their convolutional counterparts. We now examine whether this conclusion holds when examining on a different dataset or with a different transformer-based encoder. Note that the experiments herein directly show the comparison between (C,C) and (T,T). All comparisons assume known ground-truth camera intrinsics.

Different Dataset We compare on the DDAD dataset, which has scenes from a variety of locations, and has a ground truth depth of a longer range than KITTI. As earlier, we compare both the performance of depth estimation on i.i.d test set as well as the robustness to natural corruptions and adversarial attacks.

Performance: We report the best performance for each architecture on i.i.d in Table 4. Note that (T,T) outperforms (C,C) in almost all metrics. This confirms the generalizability of i.i.d performance of transformers across datasets. Additionally, we note that transformers also outperform the 3D convolutional PackNet architecture designed to handle long-range depth as curated within DDAD, despite a lack of inductive bias for the same in transformers.

Robustness: Next, we compare the robustness of the architectures against natural corruptions and adversarial attacks on the DDAD dataset. Note that this analysis is performed on a subset of the DDAD test set, consisting of a randomly selected subset of 11 of the 50 test sequences. In particular, these sequences are #{151, 154, 156, 167, 174, 177, 179, 184, 192, 194, 195}. For robustness evaluation, we report the mean RMSE across three training runs.

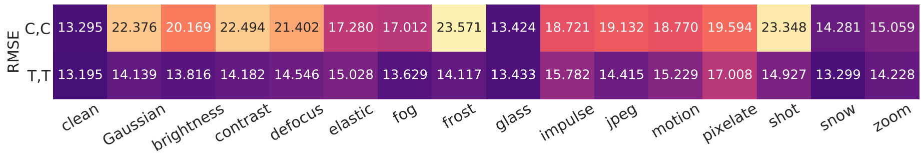


Fig. 6. RMSE for natural corruptions of DDAD. The i.i.d evaluation is denoted by *clean*.

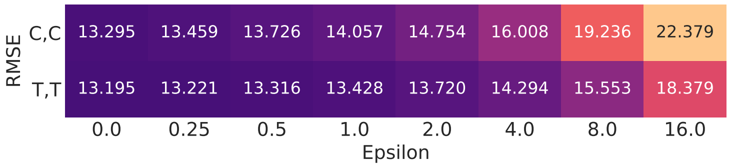


Fig. 7. RMSE for untargeted attacks on DDAD. The i.i.d evaluation is denoted by *clean*.

Figure 6 compares the robustness of the architectures to natural corruptions. We find that transformers are significantly better at handling across natural corruptions. Figure 7 further compares the robustness of architectures with untargeted adversarial perturbations. We find that transformers are also better against untargeted adversarial attacks, with the difference becoming more pronounced as the attack strength increases. Finally, Figure 8 compares the robustness of the architectures with the targeted horizontal and vertical flip attacks. Again, we find that transformers are also more robust to targeted attacks.

The above experiments confirm the generalizability of robustness of transformers across datasets.

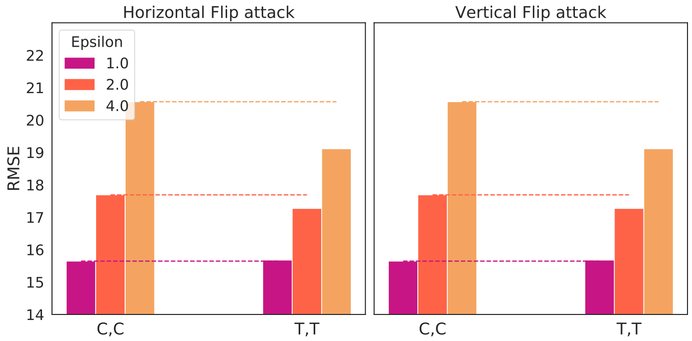


Fig. 8. RMSE for adversarial corruptions of DDAD generated using horizontal and vertical flip attacks. The i.i.d evaluation is denoted by *clean*.

Table 5. Quantitative results on KITTI Eigen with modified encoder backbone. The best results are shown in bold.

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
T,T (DeiT)	0.112	0.838	4.771	0.188	0.879	0.960	**0.982**
T,T (PVT)	**0.107**	**0.780**	**4.537**	**0.183**	**0.890**	**0.963**	**0.982**

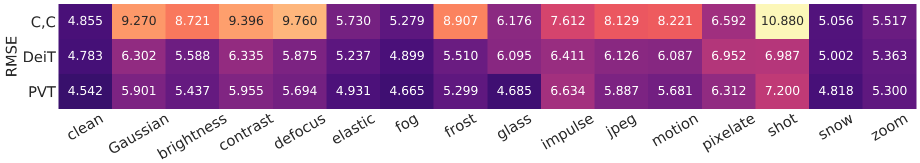
Different Encoder Backbone While we utilized the Data Efficient Image Transformer (DeiT) backbone for the encoder in previous experiments, we now evaluate MT-SfMLearner with the Pyramid Vision Transformer (PVT) backbone. As earlier, we compare both the performance of depth estimation on i.i.d test set as well as the robustness to natural corruptions and adversarial attacks for the KITTI Eigen split.

Performance: We report the best performance for each architecture on i.i.d in Table 5. Note that (T,T) with PVT outperforms not only (C,C) but also (T,T) with DeiT in all metrics. This can be attributed to the design of PVT, built particularly for dense prediction tasks, such as depth estimation. Figure 2 further shows continued globally coherent predictions for large objects and reflections for PVT. Furthermore, PVT improves on thin structures over DeiT, likely due to its overlapping patch embedding, which allows for better learning of local information. This confirms the generalizability of i.i.d performance of transformers across different encoder backbones.

Robustness: Next, we compare the robustness of the architectures against natural corruptions and adversarial attacks. For robustness evaluation, we report the mean RMSE across three training runs.

Figures 9, 10, and 11 compare the robustness of the architectures to natural corruptions, untargeted adversarial attack, and targeted horizontal and vertical flip attack. We find that both transformer architectures are significantly better at handling natural corruptions as well as untargeted and targeted adversarial attacks. We hypothesize that PVT has much higher robustness than even DeiT due to its spatial feature pyramid and overlapping patch embedding that helps to maintain local continuity in the image.

Therefore, the above experiments also confirm the generalizability of the robustness of transformers across encoder backbones.



	clean	Gaussian	brightness	contrast	defocus	elastic	fog	frost	glass	impulse	jpeg	motion	pixelate	shot	snow	zoom
C,C	4.855	9.270	8.721	9.396	9.760	5.730	5.279	8.907	6.176	7.612	8.129	8.221	6.592	10.880	5.056	5.517
DeiT	4.783	6.302	5.588	6.335	5.875	5.237	4.899	5.510	6.095	6.411	6.126	6.087	6.952	6.987	5.002	5.363
PVT	4.542	5.901	5.437	5.955	5.694	4.931	4.665	5.299	4.685	6.634	5.887	5.681	6.312	7.200	4.818	5.300

Fig. 9. RMSE for natural corruptions of KITTI Eigen including PVT. The i.i.d evaluation is denoted by *clean*.

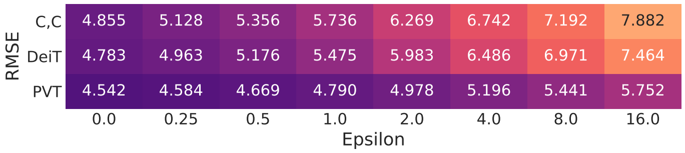


Fig. 10. RMSE for untargeted attacks on KITTI Eigen, including T, T (PVT). The i.i.d evaluation is denoted by *clean*.

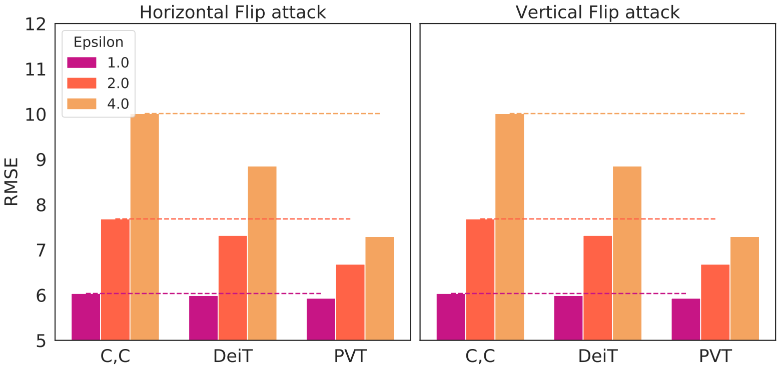


Fig. 11. RMSE for adversarial corruptions of KITTI Eigen including T, T (PVT) generated using horizontal and vertical flip attacks. The i.i.d evaluation is denoted by *clean*.

4.7 Auxiliary Tasks

Unsupervised monocular SfM requires access to relative pose between image pairs and camera intrinsics corresponding to the input, in addition to depth. As discussed in Section 3, a network is simultaneously trained for the pose and (optionally) camera intrinsics estimation along with the depth estimation network. While we have studied depth estimation in detail in the previous subsections, here we examine if the improved performance with transformers comes at the expense of the performance on auxiliary tasks.

Intrinsics Estimation: In Table 6, we examine the accuracy of our proposed intrinsics estimation network on the KITTI Eigen split. We observe that both the CNN-based and transformer-based architectures result in a low percentage error on the focal length

Table 6. Percentage error for intrinsics prediction. Source: [53]

Network	Error(%) ↓			
	f_x	c_x	f_y	c_y
C,C	-1.889	-2.332	2.400	-9.372
T,T	-1.943	-0.444	3.613	-16.204

Table 7. Impact of estimating intrinsics on pose estimation for the KITTI Odom split.

Network	Intrinsics	Seq 09		Seq 10	
		$t_{err}(\%)\downarrow$	$r_{err}(^{\circ}/100m)\downarrow$	$t_{err}(\%)\downarrow$	$r_{err}(^{\circ}/100m)\downarrow$
C,C	Given	10.376	3.237	7.784	2.444
	Learned	14.052	4.047	11.780	3.338
T,T	Given	6.998	2.181	8.983	3.666
	Learned	7.624	2.099	9.537	3.962

and principal point. The performance and robustness of depth estimation with learned camera intrinsics are discussed in Section 4.8.

Pose Estimation: In Table 7, we examine the accuracy of the pose estimation network on the KITTI Odom split, including when the intrinsics are unknown. We observe that both the translation and rotation errors for sequence 09 are lower with (T,T) than with (C,C) when the camera intrinsics are given. However, the opposite is true for Sequence 10. We also observe that the translation and rotation errors for both sequences are similar to when the ground truth intrinsics are known a priori. Figure 12 visualizes the predicted trajectories with both architectures, including when the intrinsics are unknown.

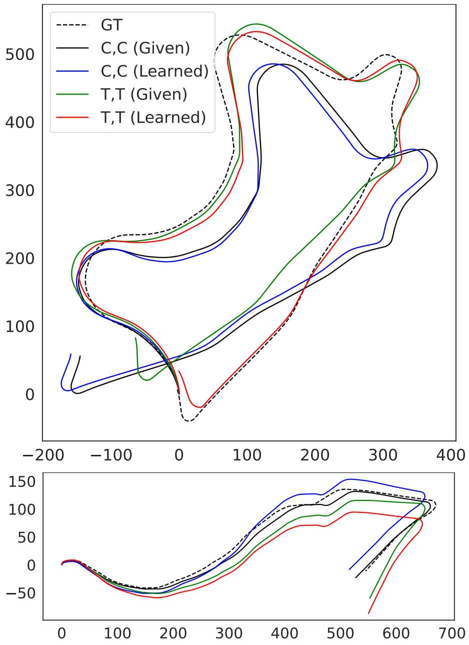


Fig. 12. Origin-aligned trajectories on KITTI Odom split for (C,C) and (T,T) showing the impact of learning intrinsics.

Table 8. Impact of estimating intrinsics on depth estimation for KITTI Eigen split. Source: [53].

Network	Intrinsics	Depth Error↓				Depth Accuracy↑		
		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	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	0.112	0.809	4.734	0.188	0.878	0.960	0.982

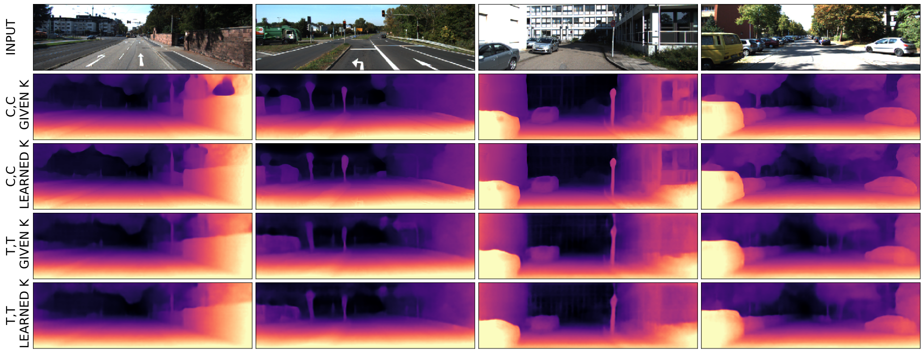


Fig. 13. Disparity maps for qualitative comparison on KITTI, when trained with and without intrinsics (K). The second and fourth rows are same as the second and the fifth rows in Figure 2. Source: [53].

Overall, we conclude that the benefits of using transformers do not come at the expense of its performance on intrinsics or pose estimation. Note that while auxiliary tasks perform well, traditional methods continue to dominate with several methods combining deep learning approaches with the traditional methods [3, 8]

4.8 Depth Estimation with Learned Camera Intrinsics

We analyze the impact on performance and robustness of depth estimation when the camera intrinsics are unknown a priori and the network is trained to estimate it. The experiments are performed on the KITTI Eigen split.

Performance: Table 8 compares the accuracy and error for depth estimation when intrinsics are learned and when they are given a priori. We observe that depth error and accuracy for transformer-based architectures continue to be better than those for CNN-based architectures, even when the intrinsics are learned. Additionally, we observe that the depth error and accuracy metrics are both similar to those when the ground truth intrinsics are known. This is also substantiated in Figure 13 where learning of the intrinsics does not cause artifacts in depth estimation.

Robustness: As before, we also evaluate networks trained with learned intrinsics for their robustness against natural corruptions, untargeted attack, and targeted adversarial

Table 9. Mean RMSE (μRMSE) for natural corruptions, horizontal (H) and vertical (V) adversarial flips of KITTI, when trained with and without ground truth intrinsics. Source: [53].

		Natural corruptions	Adversarial attack	
Architecture	**Intrinsics**	**μRMSE$\downarrow$**	**μRMSE$\downarrow$ (H)**	**μRMSE$\downarrow$ (V)**
C, C	Given	7.683	7.909	7.354
	Learned	7.714	7.641	7.196
T, T	Given	5.918	7.386	6.795
	Learned	5.939	7.491	6.929

attacks. We report the mean RMSE (μRMSE) across all corruptions and for all attacks strengths in targeted adversarial attacks in Table 9, the RMSE on the untargeted adversarial attack in Figure 14 averaged over three runs. We observe that both the architectures maintain similar robustness against natural corruptions and adversarial attacks when the intrinsics are learned simultaneously as opposed to when intrinsics are known a priori. Additionally, similar to the scenario with known ground truth intrinsics, transformers with learned intrinsics is more robust than its convolutional counterpart.

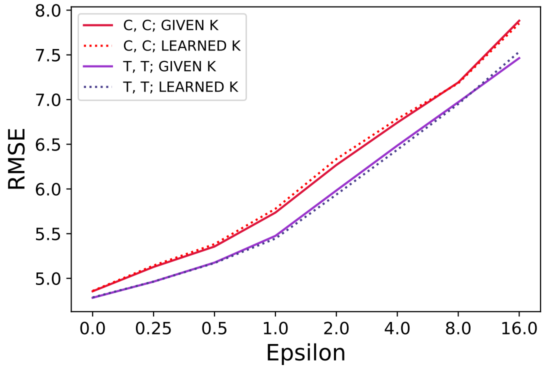


Fig. 14. RMSE for adversarial corruptions of KITTI generated using untargeted PGD attack, when trained with and without ground truth intrinsics (K). Source: [53].

4.9 Efficiency

In order to deploy the architectures for use in robots and autonomous driving systems, it is important to examine their suitability for real-time application. Thus, we evaluate the networks on their computational and energy efficiency.

Table 10 reports the average speed and the average energy consumption during inference for depth, as well as pose and intrinsics networks for both architectures. These metrics are computed over 10,000 forward passes on NVidia GeForce RTX 2080 Ti. We observe that both architectures run in real-time with an inference speed $>$ 30 fps.

Table 10. Inference Speed and Energy Consumption for depth, pose, and intrinsics estimation using CNN- and transformer-based architectures. Source: [53].

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

Nevertheless, the energy consumption and computational costs for transformer-based architecture are higher than those of its CNN-based counterpart.

4.10 Comparing Performance

Having established the benefits of transformers for depth estimation, we now evaluate MT-SfMLearner, where both depth and pose networks are transformer-based, with contemporary neural networks for their error and accuracy on unsupervised monocular depth estimation. Note that we do not compare with methods that use ground-truth depth or semantic labels during training. We also do not compare against methods that use multiple frames for depth estimation. In Table 11, we observe that MT-SfMLearner (DeiT) achieves comparable performance against other methods including those with a heavy encoder such as ResNet-101 [24] and PackNet with 3D convolutions [18]. We also observe that MT-SfMLearner (PVT) outperforms these methods, including contemporary transformer-based methods such as MonoFormer [1].

Table 11. 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. Table modified from [53].

Methods	**Resolution**	**Error↓**				**Accuracy↑**		
		Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
CC [48]	832×256	0.140	1.070	5.326	0.217	0.826	0.941	0.975
SC-SfMLearner [4]	832×256	0.137	1.089	5.439	0.217	0.830	0.942	0.975
Monodepth2 [17]	640×192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
SG Depth [27]	640×192	0.117	0.907	4.844	0.194	0.875	0.958	0.980
PackNet-SfM [18]	640×192	0.111	0.829	4.788	0.199	0.864	0.954	0.980
Poggi et. al [44]	640×192	0.111	0.863	4.756	0.188	0.881	0.961	0.982
Johnston & Carneiro [24]	640×192	**0.106**	0.861	4.699	0.185	0.889	0.962	0.982
HR-Depth [37]	640×192	0.109	0.792	4.632	0.185	0.884	0.962	**0.983**
G2S [7]	640×192	0.112	0.894	4.852	0.192	0.877	0.958	0.981
MonoFormer [1]	640×192	0.108	0.806	4.594	0.184	0.884	**0.963**	**0.983**
MT-SfMLearner (DeiT)	640×192	0.112	0.838	4.771	0.188	0.879	0.960	0.982
MT-SfMLearner (PVT)	640×192	0.107	**0.780**	**4.537**	**0.183**	**0.890**	**0.963**	0.982

5 Conclusion

This work investigates the impact of transformer-based architecture on the unsupervised monocular Structure-from-Motion (SfM). We demonstrate that learning both depth and pose using transformer-based architectures leads to highest performance and robustness in depth estimation across multiple datasets and transformer encoders. We additionally establish that this improvement in depth estimation doesn't come at the expense of auxiliary tasks of pose and intrinsics estimation. We also show that transformer-based architectures predict uniform and coherent depths, especially for larger objects, whereas CNN-based architectures provide local spatial-bias, especially for thinner objects and around boundaries. Moreover, our proposed intrinsics estimation module predicts intrinsics with low prediction error while maintaining performance and robustness on depth estimation. However, transformer-based architectures are more computationally demanding and have lower energy efficiency than their CNN-based counterpart. Thus, we contend that this work assists in evaluating the trade-off between performance, robustness, and efficiency of unsupervised monocular SfM for selecting the suitable architecture.

References

1. Bae, J., Moon, S., Im, S.: Deep digging into the generalization of self-supervised monocular depth estimation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 187–196 (2023) 3, 19
2. Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., Veit, A.: Understanding robustness of transformers for image classification. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10231–10241 (2021) 2, 3, 12
3. Bian, J.W., Zhan, H., Wang, N., Li, Z., Zhang, L., Shen, C., Cheng, M.M., Reid, I.: Unsupervised scale-consistent depth learning from video. International Journal of Computer Vision **129**(9), 2548–2564 (2021) 3, 17
4. Bian, J., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M.M., Reid, I.: Unsupervised scale-consistent depth and ego-motion learning from monocular video. In: Advances in Neural Information Processing Systems. pp. 35–45 (2019) 19
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision. pp. 213–229. Springer (2020) 1, 3
6. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021) 2
7. Chawla, H., Varma, A., Arani, E., Zonooz, B.: Multimodal scale consistency and awareness for monocular self-supervised depth estimation. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE (2021) 19
8. Chawla, H., Jukola, M., Brouns, T., Arani, E., Zonooz, B.: 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). pp. 4750–4757. IEEE (2020) 17
9. Croce, F., Hein, M.: On the interplay of adversarial robustness and architecture components: patches, convolution and attention. arXiv preprint arXiv:2209.06953 (2022) 1

10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 8
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021) 1, 3
12. Eigen, D., Puhrsch, C., 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) 7, 8
13. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. The International Journal of Robotics Research **32**(11), 1231–1237 (2013) 7
14. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012) 8
15. Geirhos, R., Jacobsen, J.H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A.: Shortcut learning in deep neural networks. Nature Machine Intelligence **2**(11), 665–673 (2020) 10
16. Godard, C., Mac Aodha, O., 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) 2, 3
17. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3828–3838 (2019) 6, 7, 9, 19
18. Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., 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) 2, 3, 4, 7, 12, 19
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 9
20. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. Proceedings of the International Conference on Learning Representations (2019) 10
21. Huang, H., Wang, Y., Erfani, S., Gu, Q., Bailey, J., Ma, X.: Exploring architectural ingredients of adversarially robust deep neural networks. Advances in Neural Information Processing Systems **34**, 5545–5559 (2021) 1
22. Jeeveswaran, K., Kathiresan, S., Varma, A., Magdy, O., Zonooz, B., Arani, E.: A comprehensive study of vision transformers on dense prediction tasks. In: Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - VISAPP,. INSTICC, SciTePress (2022) 2, 3
23. Jiang, B., Sun, P., Luo, B.: Glmnet: Graph learning-matching convolutional networks for feature matching. Pattern Recognition **121**, 108167 (2022) 3
24. Johnston, A., Carneiro, G.: 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. pp. 4756–4765 (2020) 2, 19
25. Kästner, L., Frasineanu, V.C., Lambrecht, J.: A 3d-deep-learning-based augmented reality calibration method for robotic environments using depth sensor data. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 1135–1141. IEEE (2020) 1
26. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 8

27. Klingner, M., Termöhlen, J.A., Mikolajczyk, J., Fingscheidt, T.: Self-Supervised Monocular Depth Estimation: Solving the Dynamic Object Problem by Semantic Guidance. In: ECCV (2020) 19
28. Koch, T., Liebel, L., Fraundorfer, F., Korner, M.: Evaluation of cnn-based single-image depth estimation methods. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. pp. 0–0 (2018) 8
29. Kurakin, A., Goodfellow, I., Bengio, S., et al.: Adversarial examples in the physical world (2016) 10
30. Ladicky, L., Shi, J., Pollefeys, M.: Pulling things out of perspective. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 89–96 (2014) 8
31. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998) 1
32. Li, C., Kowdle, A., Saxena, A., Chen, T.: Towards holistic scene understanding: Feedback enabled cascaded classification models. Advances in Neural Information Processing Systems **23** (2010) 8
33. Li, J., Dai, Y., Wang, J., Su, X., Ma, R.: Towards broad learning networks on unmanned mobile robot for semantic segmentation. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 9228–9234. IEEE (2022) 1
34. Lin, G., Milan, A., Shen, C., Reid, I.: Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1925–1934 (2017) 5
35. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021) 3
36. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 8
37. Lyu, X., Liu, L., Wang, M., Kong, X., Liu, L., Liu, Y., Chen, X., Yuan, Y.: Hr-depth: high resolution self-supervised monocular depth estimation. CoRR abs/2012.07356 (2020) 2, 3, 19
38. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: 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 (2018), https://openreview.net/forum?id=rJzIBfZAb 10
39. Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M., Brendel, W.: Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv preprint arXiv:1907.07484 (2019) 10
40. Ming, Y., Meng, X., Fan, C., Yu, H.: Deep learning for monocular depth estimation: A review. Neurocomputing **438**, 14–33 (2021) 2
41. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32**, 8026–8037 (2019) 8
42. Paul, S., Chen, P.Y.: Vision transformers are robust learners. arXiv preprint arXiv:2105.07581 (2021) 3, 12
43. Peng, M., Gong, Z., Sun, C., Chen, L., Cao, D.: Imitative reinforcement learning fusing vision and pure pursuit for self-driving. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 3298–3304. IEEE (2020) 1
44. Poggi, M., Aleotti, F., Tosi, F., Mattoccia, S.: On the uncertainty of self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3227–3237 (2020) 19

45. Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., Dosovitskiy, A.: Do vision transformers see like convolutional neural networks? arXiv preprint arXiv:2108.08810 (2021) 2
46. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. ArXiv preprint (2021) 4
47. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2020) 2, 3
48. Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., Black, M.J.: 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. pp. 12240–12249 (2019) 19
49. Saxena, A., Sun, M., 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) 8
50. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016) 2, 3
51. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7262–7272 (2021) 3
52. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning. pp. 10347–10357. PMLR (2021) 2, 3, 8
53. Varma., A., Chawla., H., Zonooz., B., Arani., E.: Transformers in self-supervised monocular depth estimation with unknown camera intrinsics. In: Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,. pp. 758–769. INSTICC, SciTePress (2022). https://doi.org/10.5220/0010884000003124 2, 3, 4, 5, 6, 9, 10, 11, 15, 17, 18, 19
54. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017) 1, 5
55. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pvt v2: Improved baselines with pyramid vision transformer. Computational Visual Media **8**(3), 415–424 (2022) 8
56. Wong, A., Cicek, S., Soatto, S.: Targeted adversarial perturbations for monocular depth prediction. In: Advances in neural information processing systems (2020) 11
57. Yang, G., Tang, H., Ding, M., Sebe, N., Ricci, E.: Transformer-based attention networks for continuous pixel-wise prediction. In: ICCV (2021) 2, 3
58. Yao, J., Fidler, S., Urtasun, R.: Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 702–709. IEEE (2012) 1
59. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6881–6890 (2021) 1, 3
60. Zhou, T., Brown, M., Snavely, N., 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) 2, 3, 7