Two Stream Networks for Self-Supervised Ego-Motion Estimation

Rares Ambrus **Vitor Guizilini** **Jie Li** **Sudeep Pillai** **Adrien Gaidon**
Toyota Research Institute (TRI)
firstname.lastname@tri.global

Abstract: Learning depth and camera ego-motion from raw unlabeled RGB video streams is seeing exciting progress through self-supervision from strong geometric cues. To leverage not only appearance but also scene geometry, we propose a novel self-supervised two-stream network using RGB and inferred depth information for accurate visual odometry. In addition, we introduce a sparsity-inducing data augmentation policy for ego-motion learning that effectively regularizes the pose network to enable stronger generalization performance. As a result, we show that our proposed two-stream pose network achieves state-of-the-art results among learning-based methods on the KITTI odometry benchmark, and is especially suited for self-supervision at scale. Our experiments on a large-scale urban driving dataset of 1 million frames indicate that the performance of our proposed architecture does indeed scale progressively with more data.

Keywords: Self-Supervised Learning, Ego-Motion Estimation, Visual Odometry

1 Introduction

Visual ego-motion estimation is a fundamental capability in mobile robots, used in many tasks such as perception, navigation, and planning. While visual ego-motion estimation has been well-studied in the Structure-from-Motion [1] and Visual-SLAM [2, 3] literature, recent work has shown exciting progress in self-supervised learning methods [4, 5, 6, 7]. These methods are versatile and scalable, as they learn directly from raw data, typically using the proxy photometric loss as a supervisory signal [4, 8]. Similar to Zhou et al. [4], we jointly learn a monocular depth and camera ego-motion network in a self-supervised manner. While recent works in self-supervised monocular depth and pose estimation have mostly focused on engineering the loss function [9, 10, 11, 12], we show that performance in this self-supervised SfM regime critically depends on the model architecture, in line with the observations of Kolesnikov et al. [13] and Guizilini et al. [14].

In this work, we specifically address the current limitations of self-supervised ego-motion learning architectures, namely their exclusive reliance on dense appearance changes, ignoring sparse structures that make the strength of more traditional SfM algorithms. We also investigate the strong interdependence between depth and pose in this self-supervised learning regime. We make three main contributions in this work. First, we propose **a novel two-stream network combining images and inferred depth for accurate camera ego-motion estimation**. Our architecture, inspired by action recognition models [15], efficiently leverages appearance and scene geometry, reaching state-of-the-art performance among learning-based methods on the KITTI odometry benchmark.

Second, while most learning-based methods tend to rely on generic model regularization policies to avoid overfitting, we introduce **a sparsity-inducing image augmentation scheme specifically targeted at regularizing camera ego-motion learning**. Through experiments, we show that our aggressive augmentation policy indeed reduces overfitting in this self-supervised regime, providing a simple-yet-effective mechanism to learn a sufficiently-sparse network for pose estimation.

Third, we quantify **the performance benefits and scalability of self-supervised pre-training on large datasets**. We introduce an urban driving dataset of 1 million frames, and show that by pre-training the network with large amounts of data we are able to improve monocular ego-motion estimation performance on a target dataset such as KITTI [16].

3rd Conference on Robot Learning (CoRL 2019), Osaka, Japan.

2 Related Work

Self-supervised methods for depth and ego-motion estimation have become popular, as accurate ground-truth measurements rely heavily on more expensive and specialized equipment such as LiDAR and Inertial Navigation Systems (INS). One of the earliest works in self-supervised depth estimation [8] used the photometric loss as a proxy for supervision to learn a monocular depth network from stereo imagery. In this work, the authors leverage differentiable view-synthesis [17] to geometrically synthesize the left stereo image from the right image pair and the predicted left disparity, permitting a proxy loss to be imposed between the geometrically synthesized image and the actual image captured in a stereo camera. Zhou et al. [4] extend this self-supervision to the generalized multi-view case, and leverage constraints typically incorporated in Structure-from-Motion to simultaneously learn depth and camera ego-motion from monocular image sequences. Several works have extended this work further - engineering the loss function to handle errors in the photometric loss via flow [9, 18, 10], robustly handling outliers in the loss [19, 20], incorporating 3D constraints [11], explicitly modelling dynamic object motion [21], and employing stereo and monocular constraints in the same framework [5, 22, 23]. Teed and Deng [18] proposed an iterative method to regress dense correspondences from pairs of depth frames and compute the 6-DOF estimate using the Perspective-n-Point (PnP) [24] algorithm. Instead, in this work we show that performance in the self-supervised SfM regime critically depends on the choice of the model architecture and the specific ego-motion optimization task at hand. By drawing insights through ablation studies, we introduce a sparsity-inducing image augmentation scheme to effectively regularize ego-motion learning, instead of only limiting such modifications to the underlying loss-function.

Multi-Stream architectures for multi-modal learning While recent works in self-supervised SfM learning have focused on tailoring the loss function [9, 10, 19], a few methods following [13] have explored the space of network architectures for such tasks [22, 14, 19]. Godard et al. [19] used a novel architecture relying on a ResNet [25] backbone which is shared between the depth and the ego-motion network. In the context of multi-modal and multi-task learning, multi-stream architectures have been shown to perform remarkably well in different challenging problems such as object detection and classification [26, 27, 28], semantic segmentation [29, 30], action recognition [15, 31], and image enhancement [32]. Chen et al. [26] uses a region proposal branch to construct a zenithal view of the LiDAR point cloud that is then applied over a depth and RGB stream. In [28], the authors classify objects by using six separate convolutional branches, each receiving a different depth map view of the object under consideration. Chung et al. [27] approach multi-modal learning by feeding two siamese networks with RGB and optical flow features for the task of person re-identification. In other works [15, 32, 33], multi-stream architectures have been designed for multi-task learning where one of the streams guide the learning for the other parallel stream by providing aggregated context and additional conditioning information. Simonyan and Zisserman [15] employ a two-stream architecture separately utilizing single frames and multi-frame optical flow over each branch for action recognition and video classification. Following it, in [31], authors study different architectures for the same objective, including inflated 3D convolutional ones. Inspired by both multi-task and multi-modal network architectures, we treat the RGB image and predicted monocular depth as two separate input modalities and introduce a two-stream architecture tailored for self-supervised ego-motion learning. Through experiments, we show that the proposed network architecture is able to extract and appropriately fuse RGB-D information from each of its branches to enable accurate ego-motion estimation.

3 Self-supervised Two-Stream Ego-motion Estimation

3.1 Method overview

As originally proposed in their work, Zhou et al. [4] define the task of simultaneously learning depth and pose from a monocular image stream and utilize the proxy photometric loss introduced in [8] to self-supervise both tasks. While Zhou et al. [4] and others [19, 10, 11, 12] have mostly limited to operating purely in the RGB domain, we note that depth prediction tasks naturally permit multi-modal and multi-task reasoning for further downstream processing.

To this end, we extend the formulation in [4] to consider the fusion of RGB and depth information within the pose network, via a two-stream network architecture. Figure 1 illustrates our overall self-supervised learning method with architecture details of our two-stream pose network. Our mod-

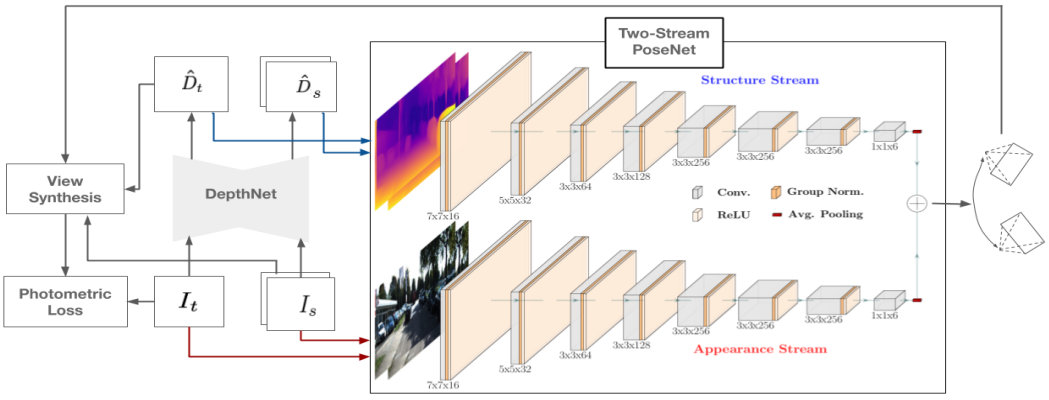

Figure 1: Our proposed self-supervised depth and camera ego-motion learning method. Contrary to previous self-supervised depth and pose estimation methods [4, 10], the RGB (I_s, I_t) and predicted depth $(\hat{D}_s, \hat{D}_t)$ images from the source and target frames are used in two separate network streams in our modified PoseNet architecture. The resulting two-stream pose network is self-supervised, achieving state-of-the-art performance on the KITTI odometry benchmark [16].

ified pose network $f_{\mathbf{x}} : (I_t, D_t, I_S, D_S) \rightarrow \mathbf{x}_{t \rightarrow S}$ estimates the 6-DOF ego-motion transformation $\mathbf{x}_{t \rightarrow s} = \left(\begin{smallmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{smallmatrix}\right) \in \text{SE(3)}$ between target frame I_t and (temporally adjacent) source frames I_s for $s \in S$, with the additional predicted depth information as inputs. This allows the proposed pose network to effectively fuse multi-modal RGB-D information for the task of ego-motion estimation, and is able to further decompose the task beyond reasoning over raw input image streams, as typically done in [4, 10, 5]. Our depth estimation network $f_d : I \rightarrow D$ is based on the DispNet network architecture [34], which is a baseline commonly used in the literature [4, 22]. The network employs a decoder with skip connections from the encoder's activation blocks and outputs depths at 4 scales. Depth at each scale is upsampled by a factor of 2 and concatenated with the decoder features to help resolve the depth and the next scale. A more detailed description of the our depth estimation network can be found in the appendix.

3.2 Two-stream ego-motion network

The proposed architecture for the ego-motion estimation task is shown in Figure 1. Inspired by work in action recognition [31, 15] we propose to augment the commonly used ego-motion estimation network which relies only on RGB input [4, 8, 11] with a second modality by passing the estimated depth, along with the RGB, as inputs to the network. This modification allows the network to learn both appearance and geometry features, leading to better results.

The architecture consists of two towers, each processing one of the modalities as shown in Figure 1. Each tower contains 8 convolutional layers plus a final average pooling layer and outputs the 6-DOF transformation between the input frames. The 6-DOF transformation is represented as 6 numbers (x, y, z) for the translation and (α, β, γ) for the rotation using the Euler parameterization. We experimented with other parameterizations for the rotation (e.g. log-quaternion [35], or quaternion [36]) but did not see any improvement.

We note that unlike related methods in the literature which process multiple frames (most methods usually process 3 frames to estimate ego-motion), our network is designed to process only two frames at a time. We show through experiments that this simple architecture is powerful enough to capture the complex dynamics of outdoor environments.

3.3 Self-supervised objective for depth and ego-motion learning

Following [4], we formulate the self-supervised objective as the minimization of the proxy photometric loss imposed between the target image I_t and the synthesized target image $\hat{I}_t$ generated from the source view. Notably, both the depth and pose networks are trained jointly via the same self-supervised proxy measure, rendering the two tasks strongly coupled with each other.

Figure 2: **Proposed augmentation method:** We show the original image and augmentation results with random 21x21 noise patches covering up to 10%, 20%, 40%, 60% and 80% respectively.

The overall loss is composed of a robust appearance loss term estimated via Structural Similarity [37], and a depth regularization term [8]. The robustness incorporated helps account for errors incurred due to occlusions and dynamic objects [19].

Robust appearance-based loss Following [8, 4, 22, 14] we define the appearance-based matching loss between two images as the linear combination between an $L1$ loss and the Structural Similarity (SSIM) loss [37] given by:

$$\mathcal{L}_p\left(I_t, \hat{I}_t\right) = \alpha \, \frac{1 - \text{SSIM}\left(I_t, \hat{I}_t\right)}{2} + (1-\alpha) \, \|I_t - \hat{I}_t\| \tag{1}$$

The SSIM component of the loss is further described in the appendix. The photometric loss as defined in Equation 1 is susceptible to errors induced by occlusions or dynamic objects. While the authors in [38] suggest clipping the photometric errors above a percentile to filter out errors, in practice we found that the *auto-masking* approach of [19] yields better results. Given target image I_t, source image I_s, and synthesized image $\hat{I}_t$ we define the masking term $\mathcal{M}_r$:

$$\mathcal{M}_r\left(I_t, I_s, \hat{I}_t\right) = \mathcal{L}_p\left(I_t, I_s\right) < \mathcal{L}_p\left(I_t, \hat{I}_t\right) \tag{2}$$

$\mathcal{M}_r$ is a robust mask that allows us to filter out stationary pixels and pixels with little photometric variation [19]. Finally, we define the robust appearance matching loss between target image I_t and context images I_S as:

$$\mathcal{L}_r\left(I_t, I_S\right) = \min_{s \in S} \mathcal{M}_r\left(I_t, I_s, \hat{I}_t\right) \cdot \mathcal{L}_p\left(I_t, \hat{I}_t\right) \tag{3}$$

We note that $\mathcal{L}_r$ is a per-pixel loss - we denote the term $\mathcal{M}_r$ that filters out static pixels between I_t and I_s, and subsequently select the loss term with the lowest value across all context images in I_S.

Depth smoothness loss In addition to the photometric term, we incorporate a multi-scale edge-aware term to regularize the depth in texture-less regions [8]:

$$\mathcal{L}_s(\hat{D}_t) = |\delta_x \hat{D}_t| e^{-|\delta_x I_t|} + |\delta_y \hat{D}_t| e^{-|\delta_y I_t|} \tag{4}$$

Finally, the loss we optimize is:

$$\mathcal{L}(I_t, I_S) = \mathcal{L}_r(I_t, I_S) \odot \mathcal{M}_r + \lambda \, \mathcal{L}_s(\hat{D}_t) \tag{5}$$

Prior to computing $\mathcal{L}$, we upsample the depth maps across all the scales to the resolution of the input image I_t, following insights from [22, 38, 19].

3.4 Sparsity-inducing data augmentation for ego-motion estimation

Our two-stream ego-motion model is more expressive, using dense information from both appearance and depth streams, and thus tends to overfit, as we show in the ablation section of our experiments (see Section 4.4). However, we know that only a subset of the input pixels are reliable and needed for direct robust visual odometry [39], although it is challenging to estimate the optimal subset.

Consequently, we propose to leverage this insight in the form of an implicit spatial sparsity prior to regularize our ego-motion network. Our approach is based on data augmentation via sampling random noise patches to obfuscate certain parts of the input images and inferred depth maps. This training methodology induces the following hyperparameters: (i) the percentage of the image to be covered by random noise, and (ii) the size of each noise patch. Figure 2 shows an example of an image covered to varying degrees of obfuscation. We present details of the effects of this data augmentation step on training and test performances in Section 4.4, showing that it indeed reduces overfitting by learning more robust sparse features from the input images. Interestingly, we also find this to be at odds with depth estimation performance, as decribed in more details in Section 4.4.

Method	Supervision	Snippet	Seq. 09	Seq. 10
SfMLearner (Zhou et al. [4])	Mono	5-frame	0.021 ± 0.017	0.020 ± 0.015
DF-Net [10]	Mono	5-frame	0.017 ± 0.007	0.015 ± 0.009
Godard et al. [19]v3	Mono	5-frame	0.017 ± 0.008	0.015 ± 0.010
Klodt et al. [40]	Mono	5-frame	0.014 ± 0.007	0.013 ± 0.009
EPC++(mono) [21]	Mono	5-frame	0.013 ± 0.007	0.012 ± 0.008
GeoNet (Yin et al. [9])	Mono	5-frame	0.012 ± 0.007	0.012 ± 0.009
Struct2Depth [12]	Mono	5-frame	0.011 ± 0.006	0.011 ± 0.010
Ours	Mono	5-frame	$\mathbf{0.010 \pm 0.002}$	$\mathbf{0.009 \pm 0.002}$
Vid2Depth [11]	Mono	3-frame	0.013 ± 0.010	0.012 ± 0.011
Shen et al [41]	Mono	3-frame	$\mathbf{0.009 \pm 0.005}$	$\mathbf{0.008 \pm 0.007}$
Ours	Mono	3-frame	$\mathbf{0.009 \pm 0.004}$	$\mathbf{0.008 \pm 0.007}$

Table 1: **Average Absolute Trajectory Error (ATE) in meters on the KITTI Odometry Benchmark [16]**: All methods are trained on Sequences 00-08 and evaluated on Sequences 09-10. The ATE numbers are averaged over all overlapping 5-frame, respectively 3-frame, snippets.

4 Experiments

4.1 Datasets

To validate our contributions we use the standard KITTI dataset [16]. We compare against state-of-the-art methods on the KITTI odometry benchmark which consists of 11 sequences (00-10), and we use the training protocol described by [4], i.e. we train on sequences 00-08 and test on sequences 09 and 10. Following related work [4, 19], we use the ground truth camera translation to scale our predictions; specifically, we compute the scaling factor for each two-frame prediction using a 5-frame window. We stack the two-frame predictions to obtain trajectories for each test sequence. We report the *Absolute Trajectory Error (ATE)* [42] averaged over all overlapping 3-frame and 5-frame snippets of the test sequences as shown in Table 1. In addition, we report t_{rel} - average translational RMSE drift (%) on trajectories of length 100-800m, and r_{rel} - average rotational RMSE drift ($\mathrm{deg}/100m$) on trajectories of length 100-800m, as described by [16]. We present these metrics and compare against other learning based monocular and stereo methods in Table 2. In addition, in Section 4.4 we also evaluate the performance of the depth estimation component. We use the quantitative depth evaluation to illustrate the counter-intuitive dependency between jointly optimizing depth and pose. The depth evaluation is done using the Eigen test split [43] of the KITTI raw dataset, which consists of 697 depth images. To explore self-supervised learning of monocular SfM at scale, we consider two additional urban driving datasets in this work. We experiment with pre-training our model on the publicly available CityScapes dataset [44] (88K images), and introduce a new urban driving dataset consisting of 24 sessions and 1 million images(the data will be made available upon request). We first filter the dataset of redundant sequential images by simply thresholding on the difference of JPEG payload size, and use the remaining images as training data. Following the pre-training step, we fine-tune on the KITTI dataset, this time using sequences 01, 02, 06, 08, 09 and 10 for training and 00, 03, 04, 05 and 07 for testing, to facilitate comparison with other methods (e.g. DVSO [23, 22]). Finally, we present our results as well as comparisons with other approaches based on direct methods, stereo, or lidar-based in Table 4.

4.2 Implementation details

All our models are implemented in PyTorch [47], using the Adam optimizer [48] with β_1 = 0.9 and β_2 = 0.999. Both the depth and the pose networks receive as input images of size 320×320 pixels. We set the SSIM weight $\alpha = 0.85$ and the depth smoothness weight $\lambda = 0.1$. We use a batch size of 8 images and train for 200 epochs. We start with learning rates of $1e^{-3}$ and $5e^{-4}$ for the depth and pose networks respectively, and reduce the learning rates by a factor of 0.5 every 80 epochs. For the regularization and sparsity-inducing data augmentation, we performed an ablation study (more details in Section 4.4) and found a threshold of 20%-40% of the image size and square blocks with sides of 80-100 pixels to work best.

4.3 Results on the KITTI odometry benchmark

Table 1 shows our ATE results on Sequences 09 and 10, evaluated on 3 and 5-frame snippets. As can be seen, our method achieves state-of-the-art results compared to other learning-based methods, and on par with Shen et al. [41]. While Shen et al. [41] propose to augment the photometric error

Method	Supervision	00*	01*	02*	03*	04*	05*	06*	07*	08*	09†	10†	Train Avg	Test Avg	
t_{rel} - Average Translational RMSE drift (%) on trajectories of length 100-800m.															
ORB-SLAM-M [42]	Mono	25.29	-	-	-	-	26.01	-	24.53	32.40	-	-	-	27.05	-
VISO2-M [45]	Mono	18.24	-	4.37	-	-	19.22	-	23.61	24.18	-	-	-	17.93	-
SfMLearner [4]	Mono	66.4	35.2	58.8	10.8	4.49	18.7	25.9	21.3	21.9	18.8	14.3	29.28	16.55	
Zhan et al [46]	Mono	-	-	-	-	-	-	-	-	-	11.9	12.6	-	12.30	
EPC++(mono) [21]	Mono	-	-	-	-	-	-	-	-	-	8.84	8.86	-	8.85	
UnDeepVO [5]	Stereo	4.14	69.1	5.58	5.00	4.49	3.40	6.20	3.15	4.08	7.01	10.6	11.68	8.81	
Zhu et al [38]	Mono	4.95	45.5	6.40	4.83	**2.43**	**3.97**	**3.49**	**4.50**	**4.08**	**4.66**	**6.30**	8.91	**5.48**	
Ours‡	Mono	**4.88**	**12.61**	**4.19**	**4.01**	3.2	5.26	8.18	6.33	7.34	6.72	9.52	**6.22**	8.12	
Ours	*Mono*	***1.29***	***1.63***	***1.06***	***1.84***	***0.55***	***1.58***	***0.91***	***2.25***	***1.84***	***3.51***	***2.32***	***1.44***	***2.92***	
r_{rel} - Average Rotational RMSE drift ($^\circ/100m$) on trajectories of length 100-800m.															
ORB-SLAM-M [42]	Mono	7.37	-	-	-	-	10.62	-	10.83	12.13	-	-	-	10.23	-
VISO2-M [45]	Mono	2.69	-	1.18	-	-	3.54	-	4.11	2.47	-	-	-	2.80	-
SfMLearner [4]	Mono	6.13	2.74	3.58	3.92	5.24	4.1	4.8	6.65	2.91	3.21	3.30	4.45	3.26	
Zhan et al [46]	Mono	-	-	-	-	-	-	-	-	-	3.60	3.43	-	3.52	
EPC++(mono) [21]	Mono	-	-	-	-	-	-	-	-	-	3.34	3.18	-	3.26	
UnDeepVO [5]	Stereo	1.92	1.60	2.44	6.17	2.13	1.5	1.98	2.48	1.79	3.61	4.65	2.45	4.13	
Zhu et al [38]	Mono	1.39	1.78	1.92	2.11	1.16	1.2	1.02	1.78	1.17	1.69	1.59	1.50	1.64	
Ours	Mono	**0.55**	**0.48**	**0.45**	**0.94**	**0.45**	**0.67**	**0.34**	**1.15**	**0.70**	**1.57**	**1.48**	**0.64**	**1.53**	

Table 2: Comparison to self-supervised learning methods on the KITTI odometry benchmark. We report the following metrics: t_{rel} - average translational RMSE drift (%) and r_{rel} - average rotational RMSE drift ($^\circ/100m$). The methods are trained on Sequences 00-08 (*) and tested on Sequences 09 and 10 (†); The results of the methods trained with monocular data were scaled using the scale from the ground truth translation. ‡ denotes global scale alignment while *italics* denote iterative scaling of snippet trajectories. The numbers for [45, 42, 5] are reported from [5].

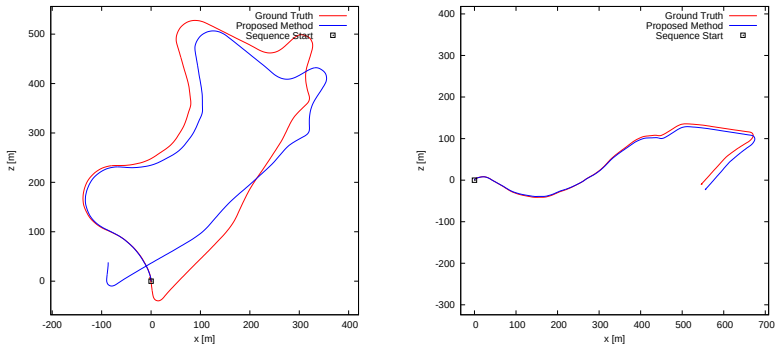

Figure 3: Qualitative trajectory results of the proposed method on test sequences 09 and 10 of the KITTI odometry benchmark.

between two frames through an additional loss term imposed via geometric constraints derived from epipolar geometry, our work shows that the early fusion of depth and RGB information coupled with an appropriate training scheme can achieve competitive results without the need of additional loss terms or external information.

In addition, we also report the average translational RMSE drift (t_{rel}) and average rotational RMSE drift (r_{rel}) on trajectories of length 100-800m in Table 2. As described in 4.1, we compute the scale for our incremental predictions over 5-frame snippets, denoted by the *italic* font in Tables 2 and 2. We also provide results when computing a global alignment factor [49], denoted by ‡ in Table. 2. Our t_{rel} and r_{rel} results are better than related monocular or stereo based learning methods, even though we estimate the transformations on a frame-to-frame basis and do not explicitly account for the errors induced by dynamic objects or occlusions when computing the photometric loss. Our model achieves state-of-the-art results when compared to similar methods due to the use of two complementary modalities: appearance and geometry. In addition, by explicitly regularizing our model during training using the proposed method, we also force the model to learn sparse features which further increase its performance. We note that our globally aligned trajectories fall short of Zhu et al. [38] in terms of the t_{rel} metric - our method suffers from the standard scale drift present in monocular self-supervised methods, while [38] use RANSAC to filter outliers and the pose estimate is computed externally and not learned. We present qualitative results of our method on the test Sequences 09 and 10 in Figure 3, with more qualitative results on the training sequences in the supplementary materials.

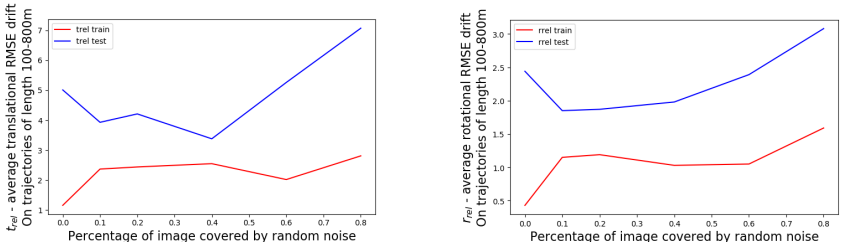

Figure 4: **Ablation study on the amount of noise augmentation:** We show test and train t_{rel} (left) and r_{rel} (right) computed when training with varying levels of the proposed augmentation method. We obtain the best results when covering 20% of the input images with random noise.

4.4 Ablation study

We present ablation results in Table 3. The first row represents our baseline, which uses only RGB images as input. The third row shows results using the proposed two-stream network architecture, described in Section 3.2, but without the proposed regularization and sparsity-inducing training methodology. As we can see, the increased modeling power of the pose network leads to even more overfitting during training and increased depth performance at test time. This shows that when optimizing for depth we are effectively interested in letting the pose network overfit as much as possible, at the expense of its test-time generalization performance. Rows 2 and 4 of Table 3 show the performance of the baseline and of the proposed two-stream network architecture with the proposed training methodology: the ego-motion network overfits less at train time, which leads to much better performance at test time. Evaluating depth shows that due to reduced performance of the ego-motion component during training, the performance of the depth network suffers as well.

We also perform an ablative analysis to better understand how our proposed augmentation technique affects performance. Specifically, we vary 2 hyperparameters: the percentage of pixels to cover with noise (10%, 20%, 40%, 60%, 80%) and the size noise patches to apply (we experiment with square patches of size 21x21, 41x41, 61x61, 81x81 and 101x101 pixels, see Figure 2). The results are summarized in Figure 4: the proposed augmentation technique has the desired effect of regularizing the solution and reducing overfitting. We get the best test results when employing an augmentation-level of 20%-40%, and when using the larger size patches (81x81 and 101x101). When going above 40% augmentation the solution degrades; however we note that, surprisingly, even when replacing 80% of the input image with random noise we are still able to regress the ego-motion.

4.5 Does self-supervised learning of ego-motion improve with more data?

To further evaluate our method, we study the effects of self-supervised pre-training on large datasets. We introduce an urban driving dataset of 1 million frames, and show that by pre-training the network with additional data we are able to gradually improve monocular pose estimation performance on a target dataset such as KITTI [16]. Furthermore, we compare these results to pre-training on CityScapes [44], a dataset similar to KITTI. We show that large amounts of unlabeled driving data can be a scalable alternative to highly curated datasets in the same domain.

As evidenced by previous works [8], pretraining on the CityScapes dataset [44] (CS) can substantially improve depth and pose estimation performance on the KITTI dataset. In addition, in this work we also investigate how these models scale and perform when self-supervised with larger amounts of unlabelled video data. For all self-supervised pre-training experiments, we first pretrain our models either on the approximately 80K images of CityScapes (CS) or the 1M urban driving dataset (D1M)

	Pose				**Depth**						
	Train		Test								
Method	t_{rel}	r_{rel}	t_{rel}	r_{rel}	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Ours - RGB, w/o Depth, w/o aug	1.39	0.59	5.93	2.64	0.138	1.084	5.336	0.220	0.823	0.937	0.973
Ours - RGB + aug, w/o Depth	2.61	1.10	4.62	2.15	0.143	1.110	5.335	0.220	0.811	0.937	0.974
Ours - RGB + Depth, w/o aug	1.16	0.43	4.91	2.44	0.135	1.031	5.260	0.216	0.826	0.940	0.974
Ours	1.44	0.64	2.92	1.53	0.139	1.063	5.349	0.221	0.817	0.937	0.974

Table 3: **Ablation study on the KITTI odometry benchmark.** We report t_{rel} & r_{rel} for different versions of our method. All methods are trained on Sequences 00-08, tested on Sequences 09 & 10.

Method	Sensor	01*	02*	06*	08*	09*	10*	00†	03†	04†	05†	07†	Train Avg	Test Avg
t_{rel} - Average Translational RMSE drift (%) on trajectories of length 100-800m.														
SuperDepth [22]	Stereo	13.48	3.48	1.81	2.25	3.74	2.26	6.12	7.90	11.80	4.58	7.60	4.50	7.60
DeepVO [51]	Mono+Pose	-	-	5.42	-	-	8.11	-	8.49	7.19	2.62	3.91	-	5.96
Velas et al. [52]	Lidar+IMU	4.44	3.42	1.88	2.89	4.94	3.27	3.02	4.94	1.77	2.35	1.77	3.11	3.22
VISO2_S [45]	Stereo	-	-	1.48	-	-	1.17	-	3.21	2.12	1.53	1.85	-	1.89
LO-NET [50]	Lidar	1.36	1.52	**0.71**	2.12	1.37	1.80	1.47	1.03	0.51	1.04	1.70	1.09	1.75
LOAM [53]	Lidar	**0.78**	1.43	0.92	**0.86**	**0.71**	**0.57**	**0.65**	**0.63**	1.12	0.77	0.79	-	1.33
ORB-SLAM2 [42]	Stereo	1.38	**0.81**	0.82	1.07	0.82	0.58	0.83	0.71	0.45	0.64	0.78	-	0.81
DVSO [23]	Stereo	1.18	0.84	**0.71**	1.03	0.83	0.74	0.71	0.77	**0.35**	**0.58**	**0.73**	**0.89**	**0.63**
Ours - KITTI‡	Mono	17.59	6.82	8.93	8.38	6.49	9.83	7.16	7.66	3.8	6.6	11.48	9.67	7.34
Ours - KITTI + CS‡	Mono	11.36	3.41	6.41	6.67	4.48	8.84	5.85	5.99	2.69	4.79	7.06	6.86	5.28
Ours - KITTI + D1M‡	Mono	9.04	5.15	4.21	5.07	3.7	6.9	5.42	6.92	2.87	5.07	4.28	5.68	4.91
Ours - KITTI	*Mono*	*4.74*	*2.6*	*0.97*	*1.72*	*1.98*	*2.56*	*3.83*	*5.74*	*1.45*	*1.54*	*2.94*	*2.43*	*3.1*
Ours - KITTI + CS	*Mono*	*1.45*	*1.32*	*0.78*	*1.87*	*1.20*	*1.14*	*3.55*	*4.19*	*1.52*	*2.29*	*3.08*	*1.29*	*2.93*
Ours - KITTI + D1M	*Mono*	*0.91*	*1.22*	*0.79*	*1.57*	*1.28*	*0.84*	*3.04*	*3.88*	*1.40*	*2.16*	*2.57*	*1.10*	*2.61*
r_{rel} - Average Rotational RMSE drift ($^\circ/100m$) on trajectories of length 100-800m.														
SuperDepth [22]	Stereo	1.97	1.10	0.78	0.84	1.19	1.03	2.72	4.30	1.90	1.67	5.17	1.15	3.15
DeepVO [51]	Mono+Pose	-	-	5.82	-	-	8.83	-	6.89	6.97	3.61	4.60	-	6.12
VISO2_S [45]	Stereo	-	-	1.58	-	-	1.30	-	3.25	2.12	1.60	1.91	-	1.96
LO-NET [50]	Lidar	0.47	0.71	0.50	0.77	0.58	0.93	0.72	0.66	0.65	0.69	0.89	0.63	0.79
ORB-SLAM2 [42]	Stereo	0.20	0.28	0.25	0.31	0.25	0.28	0.29	0.17	0.18	0.26	0.42	-	0.26
DVSO [23]	Stereo	**0.11**	**0.22**	**0.20**	**0.25**	**0.21**	**0.21**	**0.24**	**0.18**	**0.06**	**0.22**	**0.35**	**0.20**	**0.21**
Ours - KITTI	Mono	1.01	0.87	0.39	0.61	0.86	0.98	1.70	3.49	0.42	0.90	2.05	0.79	1.71
Ours - KITTI + CS	Mono	0.59	0.57	0.46	0.66	0.47	0.56	1.55	2.82	0.78	1.08	1.97	0.55	1.64
Ours - KITTI + D1M	Mono	0.35	0.56	0.39	0.58	0.48	0.58	1.32	2.97	0.62	1.04	1.75	0.49	1.54

Table 4: Comparison to direct, feature based, and lidar based methods on the KITTI odometry benchmark. We report the following metrics t_{rel} and r_{rel} averaged over trajectories of length 100-800m. † and $*$ represent test and respectively train seq. for our method, as well as for [22, 23]. [50] and [52] are trained on Seq. 00-06 and tested on Seq. 07-10. DeepVO [51] is trained on Seq. 00, 02, 08 and 09. The numbers for [52] are taken from [50]. The results of the methods trained only with Mono data were scaled using ground truth depth scale. ‡ denotes global scale alignment while *italics* denote iterative scaling of snippet trajectories. Our self-supervised method is not yet competitive with stereo and lidar, but shows a clear trend of improvement with more data, towards closing the gap with these complex methods.

(i.e. approximately 50 epochs on CS or 5 epochs on the D1M driving dataset). Subsequently, we fine-tune the model on the target KITTI dataset, using the same protocol as described in Section 4.2. For a fair comparison we use the KITTI odometry sequences *01, 02, 06, 08, 09, 10* for training and test on sequences *00, 03, 04, 05, 07*. When training with the 1M dataset, we notice the benefits of self-supervision at scale and observe noticeable improvements in pose estimation performance. Interestingly, CityScapes (CS) pre-training performs quite well in the ego-motion benchmark despite having pre-trained on a smaller dataset compared to the 1M driving dataset (D1M). This can be attributed to the similarity in domains between CityScapes (CS) and the target KITTI dataset that were both captured in geographically similar regions. These results, as well as comparisons against direct, traditional and lidar based methods are summarized in Table 4. As in Table 2, we denote our results when computing a global alignment step with ‡. As before, we notice a performance drop in the t_{rel} metric which we ascribe to scale inconsistency of our self-supervised model. However, we record a significant improvement in the t_{rel} metric with data for the globally scaled models, from which we conclude that training on more data has the additional effect of regularizing the scale of the model across the dataset. We note that our method falls short of the state-of-the-art results obtained by direct methods such as DVSO [23] or ORB-SLAM2 [42] (i.e. without loop closure and global optimization), however, we outperform the learning-based method of Pillai et al. [22] which relies on stereo supervision, and our orientation estimates outperform [45] (stereo) and [50] (Lidar).

5 Conclusion

This paper addresses the problem of learning monocular ego-motion estimation in a self-supervised setting. We explore the inter-dependence between depth regression and ego-motion estimation in the self-supervised regime, gaining insights into training methodologies when optimizing for ego-motion. Leveraging our insights, we propose a new two-stream network architecture along with a sparsity-inducing image augmentation technique that reduces pose overfitting, allowing the network to better generalize. We validate our contributions through extensive comparisons on the standard KITTI benchmark and we show that our method achieves state-of-the-art results. In addition, we also investigate the ability of self-supervised learning methods to scale with unlabeled data by training our method on an urban driving dataset containing 1 million images. We show that through self-supervised pre-training we are able to achieve additional gains, further narrowing the performance gap between learned and direct ego-motion estimation methods.

References

[1] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE TPAMI*, (6):1052–1067, 2007.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Transactions on Robotics*, 31(5):1147–1163, Oct 2015.

[4] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017.

[5] R. Li, S. Wang, Z. Long, and D. Gu. UnDeepVO: Monocular visual odometry through unsupervised deep learning. *arXiv preprint arXiv:1709.06841*, 2017.

[6] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *AAAI*, 2017.

[7] S. Pillai and J. J. Leonard. Towards visual ego-motion learning in robots. In *IROS*, 2017.

[8] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017.

[9] Z. Yin and J. Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. In *CVPR*, 2018.

[10] Y. Zou, Z. Luo, and J.-B. Huang. DF-Net: Unsupervised Joint Learning of Depth and Flow using Cross-Task Consistency. In *ECCV*, 2018.

[11] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *CVPR*, 2018.

[12] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, 2019.

[13] A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019.

[14] V. Guizilini, R. Ambrus, S. Pillai, and A. Gaidon. Packnet-sfm: 3d packing for self-supervised monocular depth estimation. *arXiv preprint arXiv:1905.02693*, 2019.

[15] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

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

[17] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015.

[18] Z. Teed and J. Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018.

[19] C. Godard, O. Mac Aodha, and G. Brostow. Digging into self-supervised monocular depth estimation. *arXiv preprint arXiv:1806.01260*, 2018.

[20] L. Zhou, J. Ye, M. Abello, S. Wang, and M. Kaess. Unsupervised learning of monocular depth estimation with bundle adjustment, super-resolution and clip loss. *arXiv preprint arXiv:1812.03368*, 2018.

[21] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *arXiv preprint arXiv:1810.06125*, 2018.

[22] S. Pillai, R. Ambrus, and A. Gaidon. SuperDepth: Self-Supervised, Super-Resolved Monocular Depth Estimation. In *ICRA*, 2019.

[23] N. Yang, R. Wang, J. Stückler, and D. Cremers. Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry. In *ECCV*, 2018.

[24] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate o (n) solution to the PnP problem. *IJCV*, 2009.

[25] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[26] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, 2017.

[27] D. Chung, K. Tahboub, and E. J. Delp. A two stream siamese convolutional neural network for person re-identification. In *ICCV*, 2017.

[28] P. Zanuttigh and L. Minto. Deep learning for 3d shape classification from multiple depth maps. In *ICIP*, 2017.

[29] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision*, pages 213–228. Springer, 2016.

[30] A. Valada, R. Mohan, and W. Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *arXiv preprint arXiv:1808.03833*, 2018.

[31] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[32] H. Zhang and V. M. Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, 2018.

[33] N. Radwan, A. Valada, and W. Burgard. Vlocnet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robotics and Automation Letters*, 3(4):4407–4414, 2018.

[34] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

[35] S. Brahmbhatt, J. Gu, K. Kim, J. Hays, and J. Kautz. Geometry-aware learning of maps for camera localization. In *CVPR*, 2018.

[36] A. Kendall, R. Cipolla, et al. Geometric loss functions for camera pose regression with deep learning. In *Proc. CVPR*, volume 3, page 8, 2017.

[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 2004.

[38] A. Z. Zhu, W. Liu, Z. Wang, V. Kumar, and K. Daniilidis. Robustness meets deep learning: An end-to-end hybrid pipeline for unsupervised learning of egomotion. *arXiv preprint arXiv:1812.08351*, 2018.

[39] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *IEEE TPAMI*, 2018.

[40] M. Klodt and A. Vedaldi. Supervising the New with the Old: Learning SFM from SFM. In *ECCV*, 2018.

[41] T. Shen, Z. Luo, L. Zhou, H. Deng, R. Zhang, T. Fang, and L. Quan. Beyond photometric loss for self-supervised ego-motion estimation. *arXiv preprint arXiv:1902.09103*, 2019.

[42] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[43] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, pages 2366–2374, 2014.

[44] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[45] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968. Ieee, 2011.

[46] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *CVPR*, 2018.

[47] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

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

[49] M. Grupp. evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo, 2017.

[50] Q. Li, S. Chen, C. Wang, X. Li, C. Wen, M. Cheng, and J. Li. LO-Net: Deep Real-time Lidar Odometry. In *CVPR*, 2019.

[51] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *ICRA*, 2017.

[52] M. Velas, M. Spanel, M. Hradis, and A. Herout. Cnn for imu assisted odometry estimation using velodyne lidar. In *ICARSC*, 2018.

[53] J. Zhang and S. Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41(2): 401–416, 2017.