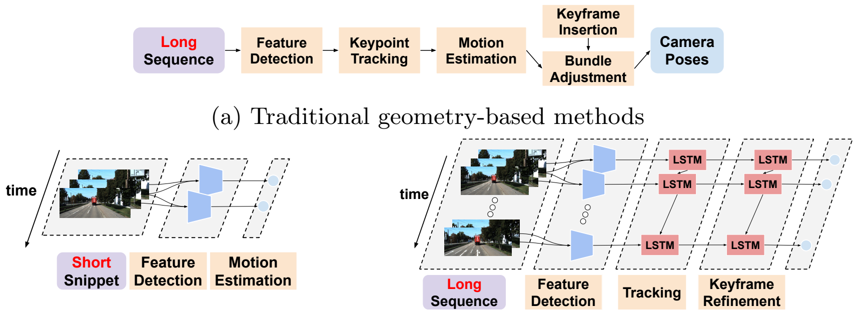
Learning Monocular Visual Odometry via Self-Supervised Long-Term Modeling

Yuliang Zou$^{1[0000-0002-8374-6662]}$, Pan Ji$^{2[0000-0001-6213-554X]}$, Quoc-Huy Tran$^{2[0000-0003-1396-6544]}$, Jia-Bin Huang$^{1[0000-0002-0536-3658]}$, and Manmohan Chandraker$^{2,3[0000-0003-4683-2454]}$

[1]Virginia Tech [2]NEC Labs America [3]UCSD



(a) Traditional geometry-based methods

(b) Existing self-supervised methods

(c) Our proposed method

Fig. 1: **Learning monocular visual odometry with long-term modeling.** Existing self-supervised methods only see *short* snippets during the *training* time, which makes it hard to learn to leverage temporal consistency over *long* sequences. Our method, in contrast, inspired by geometry-based visual odometry methods, combines the best of both the geometry and learning to aggregate long-term temporal information.

Abstract. Monocular visual odometry (VO) suffers severely from error accumulation during frame-to-frame pose estimation. In this paper, we present a self-supervised learning method for VO with special consideration for consistency over longer sequences. To this end, we model the long-term dependency in pose prediction using a pose network that features a two-layer convolutional LSTM module. We train the networks with purely self-supervised losses, including a cycle consistency loss that mimics the loop closure module in geometric VO. Inspired by prior geometric systems, we allow the networks to see beyond a small temporal window during training, through a novel a loss that incorporates temporally distant (e.g., $O(100)$) frames. Given GPU memory constraints, we propose a stage-wise training mechanism, where the first stage operates in a local time window and the second stage refines the poses with a "global" loss given the first stage features. We demonstrate competitive results on several standard VO datasets, including KITTI and TUM RGB-D.

1 Introduction

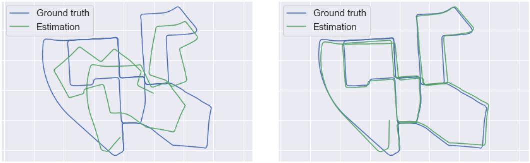
Most existing VO systems are either *geometric* or *learning-based*. In this paper, we argue that a truly robust VO system should combine the best of both worlds (i.e., geometry and learning). In particular, we propose a self-supervised method to learn monocular VO with long-term modeling, where the training scheme is directly inspired by traditional geometric methods (see Figure 1).

At the heart of the state-of-the-art VO systems [8,9,10,24] is the incorporation of several long-studied geometric modules, including keypoint tracking, motion estimation, keyframe insertion, and bundle adjustment (BA) [38]. With all these modules, a key insight is to optimize the states (e.g., 6-DoF camera poses) over *long-term* observations such that the system suffers less from error accumulation [27]. While being robust in normal scenarios, monocular VO still suffers from the difficulty in initialization for slow motions [23], and the tracking tends to fail miserably in unconstrained environments with large texture-less regions, fast movements, or other adverse factors [48] such as rolling shutter effect [28,53] and unknown camera intrinsics [3,54].

In contrast, learning-based VO methods [44,45,46,47] have the potential of being more robust to the aforementioned challenges by harnessing the rich priors from data. However, training neural networks in a supervised way involves collecting large-scale, diverse datasets with ground truth annotations, which could be labor-extensive and time-consuming. Recently, self-supervised methods [13,20,26,42,50,52,55] have been proposed to tackle this task. Instead of supervising the networks with ground truth labels, the idea is to couple the depth and pose networks with photometric errors across adjacent frames and jointly train them in an end-to-end manner. Nonetheless, the performance of these methods still falls behind that of geometric methods [23] for general scenarios.

One of the potential reasons for their performance gap is that the pose networks do not exploit the temporal coherence over long sequences. During training, these networks receive *short* snippets (e.g., 3-frame or 5-frame) as input and predict the ego-motions that are optimized *locally* for the current snippet. When evaluating these methods in short snippets, they compare favorably even with the state-of-the-art geometric methods [23]. However, if we concatenate all the predictions to form the full trajectory, it is often the case that the learning-based methods generate much larger pose errors, as illustrated in Figure 2.

In this paper, we argue that learning VO requires explicit *long-term* modeling to infuse the insights from geometric methods [8,9,10,24]. To this end, we propose a novel self-supervised VO learning framework that draws inspiration from geometric modules. Specifically, we build our learning framework upon a depth network of an auto-encoder structure with skip-connections [13] and a pose network with a two-layer LSTM module [47]. In contrast to the supervised method by Xue et al. [47], our method incorporates extra depth information and uses a completely different training scheme, leading to a purely self-supervised learning framework. To mitigate error accumulation, we propose a cycle consistency constraint between the two-layer predictions, mimicking a mini *loop closure* module, which improves the pose consistency over the sequence. In order to model long-term dependency in VO, we propose a two-stage training strategy,



(a) SfMLearner [52] (b) Our method

Evaluation	5-frame	video-level
SfMLearner [52]	0.019	97.81
Our method	0.015	13.13

(c) RMSE (m)

Fig. 2: **5-frame v.s. video-level evaluation.** Evaluating visual odometry requires having a global picture of recovered trajectories. However, most self-supervised methods only evaluate the trajectories within a short snippet, which may not reflect the holistic performance. "5-frame" means that we evaluate the results using 5-frame snippets, and "video-level" means that we evaluate on the entire trajectory.

which considers both short-term and long-term constraints. The proposed two training stages correspond to the *local* and *global* bundle adjustment modules in the geometric VO, allowing us to refine the poses within a large temporal range.

In summary, our contributions are:

- We propose a novel self-supervised VO learning framework that explicitly models long-term temporal dependency.
- We build connections between our method and key building blocks of geometric VO systems and demonstrate well-motivated designs.
- We evaluate the *full* pose trajectories by our method, against the state-of-the-art geometric and learning-based baselines, and achieve competitive results on standard VO datasets, including KITTI and TUM RGB-D.

To the best of our knowledge, our method is the first of the kind that is able to learn from "truly" long sequences (e.g., ∼100 frames) in the training stage. Our experiments show that our proposed method gives rise to significant empirical benefits by explicitly considering long-term modeling.

2 Related Work

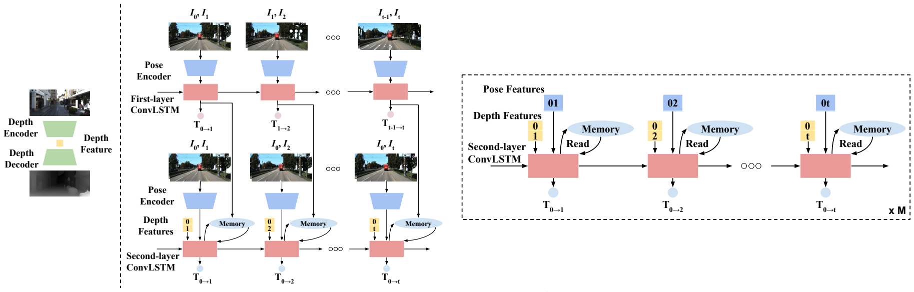
Geometric Methods. Visual odometry is a long-standing problem that estimates the ego-motion incrementally [25,27] using visual input. A conventional geometric VO system usually consists of the following components [27]: feature detection, feature matching (or tracking), motion estimation (e.g., triangulation [16]), and local optimization (e.g., bundle adjustment). A keyframe mechanism [18] is also adopted for improved robustness in motion estimation. Incorporated with a mapping system that reconstructs the 3D scene structures, a VO system turns to a system called Simultaneous Localization and Mapping (SLAM) [4]. The key to the robustness of the modern VO/SLAM systems [23,37] lies in their capability to extract reliable image measurements and optimize the states (e.g., 6-DoF camera poses) over a large number of frames. In this work, we leverage these geometric insights to design a robust learning-based VO system.

Fully-Supervised Methods. With the success of deep neural networks, end-to-end learning-based methods [44,45,46,47] have been proposed to tackle the visual odometry problem. These methods often rely on a supervised loss using the ground-truths to regress the 6-DoF camera relative pose from a pair of consecutive images. Recently, some methods [2,35,36,39,51] exploit CNNs to predict the scene depth and camera pose jointly, utilizing the geometric connection between the structure and the motion. This corresponds to learning Structure-from-Motion (SfM) in a supervised manner. Although the methods above achieve good performance, they require ground-truth annotations to train the networks. In contrast, our method is self-supervised, requiring nothing but the monocular video frames.

Self-Supervised Methods. To mitigate the requirement of data annotations, self-supervised methods [13,20,26,42,50,52] have been proposed to tackle the SfM task. The main supervisory signal of these methods comes from the photometric-consistency between corresponding pixels of neighboring frames. While they achieve good performance on single-view depth estimation, the performance of ego-motion estimation still lags behind the traditional SLAM/VO methods. Recently, Bian et al. [1] argue that the pose networks cannot provide full camera trajectories over long sequences due to the inconsistent scale of per-frame estimations and thus propose a geometry consistency constraint. However, their method only enforces the globally consistent trajectories by propagating the consistency on overlapping *short* snippets during training. In contrast, our method directly optimizes over *long* sequences via long-term modeling. Inspired by the keyframe mechanism in geometric methods, Sheng et al. [30] propose to jointly learn depth, ego-motion, and keyframe selection simultaneously in a self-supervised manner. Similarly, the training of this method considers only *short* snippets and thus is unable to model long-term dependency.

Sequential Modeling. Sequential modeling based on recurrent neural networks (RNNs) has been successfully applied to many applications, such as speech recognition [5], machine translation [14], and video prediction [33,40]. Aiming to estimate the full trajectory over a long sequence of frames, VO can be naturally formulated as a sequential learning problem and thus modeled with RNNs [42,44,45,46]. Recently, Xue et al. [47] propose to use a two-layer LSTM network for pose estimation, where the first layer estimates the relative motion between consecutive frames, and the second layer estimates global absolute poses.

Despite using a similar pose network, our method differs from Xue et al. [47] in being self-supervised v.s. full-supervised and the associated training strategies. First of all, our method further incorporates depth information, while the method in [47] relies only on pose features. Apart from the photometric discrepancy as to the supervisory signal, we enforce a cycle consistency between the predictions from the two-layer LSTM modules, which serves as a mini “loop closure” module that mimics the geometry VO system. More importantly, we decouple our network training into two stages, allowing our method to optimize over *long* sequences (more than 90 frames) during training, whereas the method in [47] only trains with 11-frame snippets. To our knowledge, this is the *first* deep learning approach for visual odometry that takes *long* sequences as input in the training stage.



(a) 1st stage of training (*short* snippet) (b) 2nd stage of training (*long* sequence)

Fig. 3: **Overview.** Our method adopts a stage-wise training strategy. (a) In the first stage of training, we jointly train all the components: the depth encoder, the depth decoder, the pose encoder, the first and second layer of ConvLSTM (Sec. 3.2); (b) In the second stage of training, we pre-extract features as input and fine-tune the second layer of the ConvLSTM module only (Sec. 3.3).

3 Method

Figure 3 provides a high-level overview of the proposed monocular VO system. Our system has two major components: a depth network and a pose network.[1] The single-image depth network employs an encoder-decoder structure with skip-connections [13]. The pose network consists of a FlowNet backbone [7], a two-layer LSTM module [47], and two pose prediction heads (with one after each of the LSTM layers). In the two-layer recurrent architecture, the first-layer focuses on predicting consecutive frame motions, while the second-layer refines the estimations from the first-layer [47].

3.1 Background

We formulate the monocular visual odometry task as a view synthesis problem, by training the networks to predict a target image from the source image with the estimated depth and camera pose. Such a system typically consists of two components: a depth network which takes a single RGB image as input to predict the depth map, and a pose network which takes a concatenation of two consecutive frames as input to estimate the 6-DoF ego-motion.

Given two input images I_t and I_{t+1}, the estimated depth map $\hat{D}_t$ and camera pose $\hat{T}_{t\to t+1}$, we can then compute the per-pixel correspondence between the two input images. Assume a known camera intrinsic matrix K, and let p_t represent the 2D homogeneous coordinate of a pixel in I_t. We can find the corresponding point of p_t in I_{t+1} following the equation [52]:

$$p_{t+1} \sim K\hat{T}_{t\to t+1}\hat{D}_t(p_t)K^{-1}p_t \ . \tag{1}$$

[1] Since accurate pose prediction is the primary focus of this paper, we name our method as VO instead of SfM or SLAM.

Appearance loss. For a self-supervised visual odometry system, the primary supervision comes from the appearance dissimilarity between the synthesis image and the target image. To effectively handle occlusion, we use three consecutive frames to compute the per-pixel minimum photometric reprojection loss [13], i.e.,

$$L_A = \frac{1}{N-2} \sum_{t=1}^{N-2} \min_{t' \in \{t-1, t+1\}} \rho(I_t, \hat{I}_{t' \to t}) \ , \tag{2}$$

where ρ is a weighted combination of the L2 loss and the Structured SIMilarity (SSIM) loss, $\hat{I}_{t' \to t}$ denotes the frame synthesized from $I_{t'}$ using Eq. (1). To handle static pixels, we adopt the auto-masking mechanism following Godard et al. [13].

Smoothness loss. Since the appearance loss cannot provide meaningful supervision for texture-less or homogeneous regions of the scene, a smoothness prior of disparity is incorporated. We here use the edge-aware smoothness loss L_S [41].

Remark 1. The appearance loss in Eq. (2) *corresponds to a local photometric bundle adjustment objective, which is also commonly used in the geometric direct VO/SLAM systems [8,9,43].*

3.2 Cycle consistency within memory-aided sequential modeling

With the above setting, current state-of-the-art self-supervised methods estimate the ego-motion within a *local* range, discarding the sequential dependence and dynamics in the *long* sequences. Such information, however, is essential for a pose network to recover the entire trajectory in a consistent manner. We thus adopt a recurrent structure of our pose network to utilize the temporal information.

Sequential modeling. To learn to utilize the temporal information, we adopt the recurrent network structure with a convolution LSTM (ConvLSTM) module [31]. Previously, the pose network takes the concatenation of two frames and outputs the 6-DoF camera pose directly. After incorporating the ConvLSTM module, the pose network also takes the previous estimation information into account when predicting the output. Formally, we have

$$F_t = \mathrm{PEnc}(I_t, I_{t-1}) \ , \tag{3}$$

$$O_t, H_t = \mathrm{ConvLSTM}(F_t, H_{t-1}) \ , \tag{4}$$

$$\hat{T}_{t-1 \to t} = g_1(O_t) \ , \tag{5}$$

where $\mathrm{PEnc}(\cdot)$ is the pose encoder, O_t, H_t denotes the output and hidden state of ConvLSTM at time t, $g_1(\cdot)$ is a linear layer to predict the 6-DoF motion $\hat{T}_{t-1 \to t}$. By doing this, the network implicitly learns to aggregate temporal information and learns the motion pattern.

Memory buffer and refinement. In the sequential modeling setting above, the pose network estimates the relative pose for every two consecutive frames. However, the motions between consecutive frames are often tiny, which results in difficulties in extracting good features for relative pose estimation. Thus, predicting the camera pose from a non-adjacent “anchor” frame to the current frame could be a better option. Note that many traditional SLAM systems [23,24]

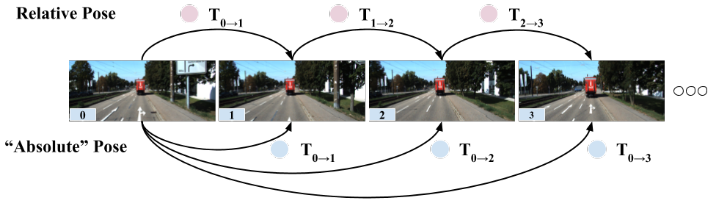


Fig. 4: **Cycle-consistency over two-layer poses.** In our model, the first layer ConvLSTM estimates the relative pose between consecutive frames, and the second layer ConvLSTM predicts the "absolute" pose within the current snippet. By exploiting the transitivity of camera poses, we incorporate a cycle-consistency constraint.

adopt a keyframe mechanism and always compute camera poses from the current frame to the most recent keyframe.

Inspired by the keyframe mechanism, we incorporate the second-layer ConvLSTM and adopt the memory module proposed by Xue et al. [47]. After each step in the first-layer ConvLSTM, we store the hidden state tensor in a memory buffer, whose length is set to the length of the input snippet. When we read out from the memory buffer, we compute the weighted average of all the memory slots in the memory buffer as in [47].

We also compute the depth and pose features for the first frame and the current frame as additional input to the second-layer ConvLSTM. This can be formally written as

$$E_t = \mathrm{DEnc}(I_t)\ , \tag{6}$$

$$F_{t,abs} = \mathrm{PEnc}(I_0, I_t)\ , \tag{7}$$

$$O_{t,abs}, H_{t,abs} = \mathrm{ConvLSTM}(F_{t,abs}, E_0, E_t, M_t, H_{t-1,abs})\ , \tag{8}$$

$$\hat{T}_{0\to t} = g_2(O_{t,abs})\ , \tag{9}$$

where $\mathrm{DEnc}(\cdot)$ is the depth encoder, M_t is the read-out memory, $O_{t,abs}$, $H_{t,abs}$ denote the output and hidden state from the second-layer at time t, and $g_2(\cdot)$ is another linear layer predicting the absolute pose in the current snippet.

Remark 2. The ConvLSTMs explicitly model the sequential nature of the VO problem and meanwhile facilitate the implementation of a keyframe mechanism. Compared to the memory module by Xue et al. [47], which only considers pose features, our memory module accommodates both depth and pose features. As verified in our experiments [2]*, incorporating the additional depth information in memory improves the overall performance.*

Cycle consistency over two-layer poses. To train the second-layer ConvLSTM, we utilize the photometric error between the first frame and the other

[2] Table 1 in the supplementary material.

frames of the input snippet, i.e.,

$$L_{A,abs} = \frac{1}{N-1} \sum_{t=1}^{N-1} \rho(I_0, \hat{I}_{t \to 0}) \ , \tag{10}$$

where N is the number of frames for the input snippet, which is set to 7 in our model.

Also, according to the transitivity of the camera transformation, we have an additional constraint to ensure the consistency between the first and second layer ConvLSTM (as shown in Figure 4), i.e.,

$$L_P = \frac{1}{N-1} \sum_{t=1}^{N-1} ||\hat{T}_{0 \to t} - \hat{T}_{t-1 \to t} \hat{T}_{0 \to t-1}||_2^2 \ . \tag{11}$$

Thus, the overall objective is

$$L_{\text{full}} = L_A + \lambda_1 L_S + L_{A,abs} + \lambda_2 L_P \ , \tag{12}$$

where λ_1, λ_2 are the hyper-parameters to balance the scale of different terms, which are empirically set to 0.001.

Remark 3. The loss in Eq. (11) *can be thought of as a mini "loop closure" module that enforces the cycle-consistency between the outputs of two ConvLSTM layers. Note that our method is also compatible with the existing full loop closure techniques [19], which we will consider in the future work.*

3.3 Long-range constraints via stage-wise training

Although we adopt a recurrent network structure to aggregate temporal information for better performance, the network has never seen *long* sequences but only *short* snippets during the training time. Thus, the network may not learn how to fully utilize the long-term temporal context. The hurdle that prevents us from taking long sequences as input is the limited memory volume of modern GPUs. To tackle this problem for training a long-term model, we propose a two-stage training strategy. We first train our whole model with the full objective L_{full} using short snippets.

Once the first stage of training is finished, we run this model on each sequence in the dataset separately, to extract the required input for the second-layer ConvLSTM and store them. After that, we only fine-tune the lightweight second-layer ConvLSTM, without the heavy feature extraction and depth networks, which saves us a lot of memories. By doing this, we can now feed long sequences into the network during the training time, allowing the network to better learn how to utilize the temporal context. Since only the second-layer ConvLSTM is optimized, our loss for the second stage of training is

$$L_{long} = \frac{1}{M} \sum_{m=0}^{M-1} \frac{1}{N-1} \sum_{t=m(N-1)+1}^{m(N-1)+N-1} \rho(I_{m(N-1)}, \hat{I}_{t \to m(N-1)}) \ , \tag{13}$$

where N is the number of frames of each snippet, which is set to 7; M is the number of snippets in the input sequence, which is set to 16. Note that consecutive snippets have one frame in common, and thus the total number of frames in the input sequence is 97. The synthesized image $\hat{I}_{t\rightarrow m(N-1)}$ is a function of depth and pose, where pose encodes long-range constraints through hidden states of ConvLSTMs, yielding an effective window of 97 frames.

Remark 4. The second training stage can be viewed as a motion-only bundle adjustment module [23] that considers long-term modeling.

4 Experimental Results

4.1 Settings

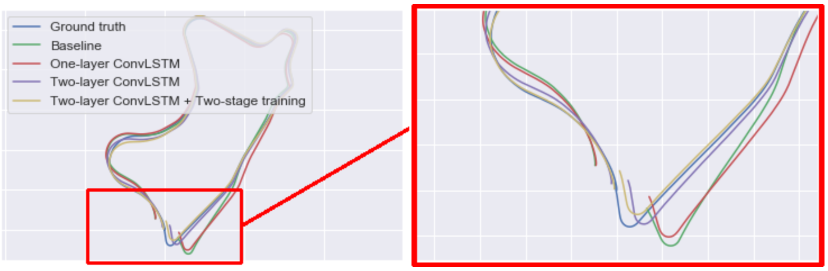
Datasets. To evaluate our method, we conduct the main experiments on the KITTI dataset [11,12], which consists of urban and highway driving sequences for road scene understanding [32,6]. The odometry split of KITTI is a widely used benchmark for odometry/SLAM evaluation. It contains 22 sequences, among which Sequence 00-10 have ground truth trajectory labels, and the annotations of the remaining sequences are not publicly available. Following Zhou et al.[52], we use Sequence 00-08 as our training set and validate the models on Sequence 09 and 10. Besides, we select 18 more sequences from KITTI raw data, which have no overlaps with the odometry split, for further evaluation. Since the ground truth trajectories of Sequence 11-21 are not available, we run ORB-SLAM2 (stereo version) to get predictions as (pseudo) ground truth for evaluation. In addition to these outdoor scenes, we also train and evaluate our model on the TUM RGB-D dataset [34]. This dataset is collected by hand-held cameras in indoor environments with challenging conditions. We use the same train/test split as in Xue et al. [47].

Evaluation metrics. For the KITTI dataset, we adopt the absolute trajectory RMSE and relative translation/rotation errors for all possible subsequences of length (100, 200, ..., 800 meters). For the TUM RGB-D dataset, we use the translational RMSE as our evaluation metrics. For self-supervised monocular methods, since the absolute scale is unknown, we align the trajectory globally using the evo toolbox [15].

Implementation details. We use ImageNet pretrained ResNet-18 as our depth encoder. Our depth decoder structure is the same as Godard et al. [13]. For the pose encoder, we take the encoder of FlowNet-S structure until the last layer, which is pre-trained on FlyingChairs [7] for optical flow estimation. We implement our system using the publicly available PyTorch framework and conduct all our experiments with a single TitanXP GPU. For both stages, we train the network with the Adam optimizer [17] for 20 epochs. The learning rate is set to 5e-5 for the first 15 epochs and drops to 5e-6 for the remaining epochs. The input size is 640×192, and the batch size is set to 2. In the first stage of training, the number of frames is set to 7, while the number of frames is set to 97 in the second stage. Note that the long-term optimization only happens in the training time. At the test time, our model runs at 14.3 frames per second.

Table 1: **Ablation study.** We evaluate different variants of the proposed method on sequences 09 and 10 of the KITTI Odometry dataset [12]. The best performance is in **bold** and the second best is underlined.

	Seq. 09			Seq. 10		
Method	RMSE (m)	Rel. trans. (%)	Rel. rot. (deg/m)	RMSE (m)	Rel. trans. (%)	Rel. rot. (deg/m)
Baseline	22.71	7.55	0.028	17.87	10.43	0.046
One-layer ConvLSTM	23.45	5.59	0.016	11.93	7.23	0.023
Two-layer ConvLSTM	**9.77**	4.23	0.013	12.68	6.02	0.023
Two-layer ConvLSTM + Two-stage training	11.30	**3.49**	**0.010**	**11.80**	**5.81**	**0.018**



(a) Full trajectories (b) Zoom-in

Fig. 5: **Visual comparison on Sequence 09.** We compare different variants of our method to validate the design choice. As we can see, adding each of the components gradually improves the overall performance. Best viewed in color.

4.2 Results

Ablation study. To validate our design choice, we perform an ablation study on Sequence 09 and 10 of the KITTI Odometry dataset. We consider the following variants. 1) Baseline: the pose network takes as input the concatenation of two consecutive frames to generate pose estimation; 2) One-layer ConvLSTM: we incorporate a one-layer ConvLSTM for the pose network; 3) Two-layer ConvLSTM: we use the full model, but only conduct the first stage of training; 4) Two-layer ConvLSTM + Two-stage training: our final model with the two-stage training strategy.

As shown in Table 1, the performance gradually improves as we add more components. Specifically, adding a recurrent module improves the overall performance over the baseline; adding the second layer LSTM leads to further improvement, which validates the effectiveness of the second layer in the self-supervised learning setting; applying our second stage long-term training again boosts the performance, achieving a new state-of-the-art for self-supervised methods. We show a visual comparison in Figure 5.

Comparison with the state-of-the-art methods. For comparison, we select several state-of-the-art methods, including the monocular version of ORB-SLAM2 [24] (denoted as ORB-SLAM2-M) (with and without loop closure optimization), several supervised learning methods [36,44,45,46,47], and some other self-supervised methods [1,13,20,21,22,26,29,42,49,50,52]. Note that all supervised methods are trained on Sequence 00, 02, 08, 09 of the KITTI Odometry

Table 2: **Comparison with the state-of-the-art.** The results of ORB-SLAM2-M methods are the medians of 5 runs. '-' means the results are not available from that paper. For DeepV2D [36], SfMLearner [52], GeoNet [49], CC [26], DeepMatchVO [29], and MonoDepth2 [13], we take the pre-trained models and run on the sequences to get the results. The best performance of each block is in **bold**, and the second best is underlined.

		Seq. 09			Seq. 10		
	Method	RMSE (m)	Rel. trans. (%)	Rel. rot. (deg/m)	RMSE (m)	Rel. trans. (%)	Rel. rot. (deg/m)
Geo.	ORB-SLAM2-M (w/o LC) [24]	44.10	9.67	**0.003**	**6.43**	**4.04**	**0.003**
	ORB-SLAM2-M [24]	**8.84**	**3.22**	0.004	8.51	4.25	**0.003**
Sup.	DeepVO [44]	-	-	-	-	8.11	0.088
	ESP-VO [45]	-	-	-	-	9.77	0.102
	GFS-VO [46]	-	-	-	-	6.32	0.023
	GFS-VO-RNN [46]	-	-	-	-	7.44	0.032
	BeyondTracking [47]	-	-	-	-	**3.94**	**0.017**
	DeepV2D [36]	**79.06**	**8.71**	**0.037**	**48.49**	12.81	0.083
Self-Sup.	SfMLearner [52]	24.31	8.28	0.031	20.87	12.20	0.030
	GeoNet [49]	158.45	28.72	0.098	43.04	23.90	0.090
	Depth-VO-Feat [50]	-	11.93	0.039	-	12.45	0.035
	vid2depth [22]	-	-	-	-	21.54	0.125
	UnDeepVO [20]	-	7.01	0.036	-	10.63	0.046
	Wang et al. [42]	-	9.88	0.034	-	12.24	0.052
	CC [26]	29.00	6.92	0.018	13.77	7.97	0.031
	DeepMatchVO [29]	27.08	9.91	0.038	24.44	12.18	0.059
	PoseGraph [21]	-	8.10	0.028	-	12.90	0.032
	MonoDepth2 [13]	55.47	11.47	0.032	20.46	7.73	0.034
	SC-SfMLearer [1]	-	11.2	0.034	-	10.1	0.050
	Ours	**11.30**	**3.49**	**0.010**	**11.80**	**5.81**	**0.018**

dataset [12], except DeepV2D [36], which is trained on the Eigen split of KITTI raw dataset [11]. As we can see in Table 2, our final model outperforms other self-supervised methods by a significant margin. In particular, our method outperforms the recent proposed SC-SfMLearner [1], which aims to reconstruct the scale-consistent camera trajectory. This indicates that the explicit long-term modeling used in our approach is more effective than propagating the geometric constraint among overlapping short snippets in SC-SfMLearner [1]. Our model also compares favorably with the geometric method and outperforms all supervised methods except Xue et al. [47] on Sequence 10.

Results on additional KITTI sequences. From the raw data of the KITTI dataset, we select 18 short sequences that have no overlaps with either the training or test split of the KITTI Odometry dataset.[3] We then apply the same pre-trained models on these test sequences. As we can see in Table 3, our method outperforms other learning-based methods and even compares favorably with ORB-SLAM2-M methods in terms of RMSE and relative translation error.

Results on KITTI test sequences. Since the ground truth trajectories of Sequence 11-21 on the KITTI Odometry dataset are not available, we cannot directly recover the global scale using similarity transformation. Thus, we choose to run the stereo version of ORB-SLAM2 (denoted as ORB-SLAM2-S), which is one of the state-of-the-art methods on these sequences. To compare with other methods, we treat the estimations from ORB-SLAM2-S as pseudo ground truth. As we can see in Table 4, our method achieves state-of-the-art performance among

[3] The sequence names are available in the supplementary material.

Table 3: **Average results on 18 additional KITTI sequences.** The results of ORB-SLAM2-M methods are the medians of 5 times. The best performance of each block is in **bold**, and the second best is underlined.

	Method	RMSE (m)	Rel. trans (%)	Rel. rot. (deg/m)
Geo.	ORB-SLAM2-M (w/o LC) [24]	**7.17**	**9.41**	**0.008**
	ORB-SLAM2-M [24]	8.12	10.64	**0.008**
Sup.	DeepV2D [36]	10.94	11.81	0.028
Self-Sup.	SfMLearner [52]	10.79	13.82	0.041
	GeoNet [49]	14.41	18.99	0.076
	CC [26]	7.51	10.49	0.024
	DeepMatchVO [29]	8.53	12.76	0.033
	MonoDepth2 [13]	7.51	11.99	0.028
	Ours	**6.47**	**9.99**	**0.019**

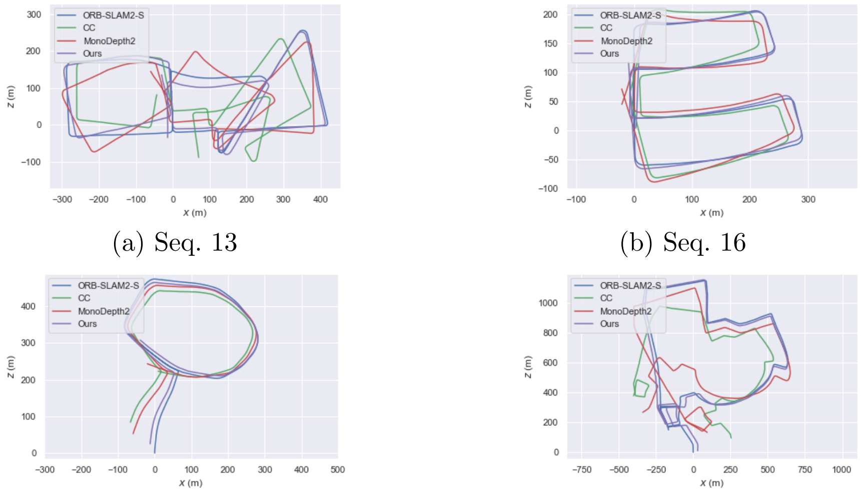
Table 4: **Results on KITTI Odometry official test split.** Since ground truth trajectories are not publicly available, we use estimations from the stereo version of ORB-SLAM2 as pseudo ground truth. The best performance of each block is in **bold**, and the second best is underlined.

	Method	RMSE (m)	Rel. trans (%)	Rel. rot. (deg/m)
Geo.	ORB-SLAM2-M (w/o LC) [24]	81.20	19.60	0.009
	ORB-SLAM2-M [24]	**44.09**	**12.96**	**0.007**
Sup.	DeepV2D [36]	221.33	24.61	0.041
Self-Sup.	SfMLearner [52]	75.00	26.54	0.045
	GeoNet [49]	94.98	29.11	0.062
	CC [26]	**55.44**	16.65	0.032
	DeepMatchVO [29]	95.79	17.31	0.038
	MonoDepth2 [13]	99.36	12.28	0.031
	Ours	71.63	**7.28**	**0.014**

the learning-based methods and even compares favorably with ORB-SLAM2-M in terms of the relative translation error. We visualize trajectories of four sequences in Figure 6. As we can see, our method better aligns with the reference trajectories from ORB-SLAM2-S. We also submit the globally aligned results to the KITTI evaluation server and get similar numbers, which indicates using ORB-SLAM2-S as pseudo ground truth for evaluation is reasonable. Please refer to the supplementary material for more details.

Results on the TUM RGB-D dataset. The TUM RGB-D dataset was created to evaluate the performance of RGB-D SLAM and is thus very challenging for monocular methods. To test our model in indoor environments, we instead compare our model with several strong baselines. For traditional methods, we choose the monocular version of ORB-SLAM2 (denoted as ORB-SLAM2-M) and DSO [8]. For learning-based methods, we choose the BeyondTracking method from Xue et al. [47] and the recent DeepV2D [36]. For DeepV2D, since it is trained on another indoor dataset, in case it cannot generalize its scale to TUM RGB-D, we provide two options: with and without global scale alignment.

Table 5 shows that traditional methods perform well on some of the sequences, but they failed to produce results from the remaining ones due to tracking failure. Our method outperforms the supervised baseline DeepV2D [36] on most of the sequences, but compares less favorably than the supervised VO method in [47]. We conjecture that this is due to the limited amount of training data available



(a) Seq. 13

(b) Seq. 16

(c) Seq. 18

(d) Seq. 19

Fig. 6: **Visual comparison on the KITTI Odometry test set.** We show the trajectories of ORB-SLAM2-S, CC [26], MonoDepth2 [13] and our method. Our method aligns best with the reference ORB-SLAM2-S trajectories. Best viewed in color.

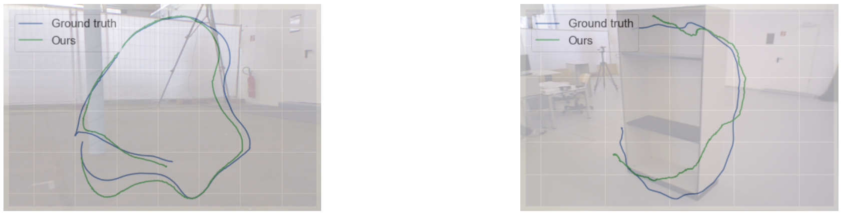


Fig. 7: **Qualitative Results on the TUM RGB-D dataset.** We overlay the first frame of each sequence with the trajectory. Best viewed in color.

on the TUM RGB-D dataset. Currently, we use the same amount of training data as supervised methods. Adding more unlabeled video data to the training might lead to better performance for our method. We also notice that the rolling shutter issue in this dataset makes the photometric consistency assumption less accurate, which could potentially hurt the performance of both the proposed method and DSO.

Discussions. Our learning framework is motivated by geometric VO methods. The FlowNet backbone mimics the *tracking module* to extract pair-wise image features, and the LSTMs model the *sequential nature* of the VO problem. The design of the two-layer LSTM module also resembles the *keyframe mechanism* of geometric VO in the sense that the second LSTM predicts the motion between a keyframe and a non-keyframe, refining the initial consecutive estimations from

Table 5: **Results on the TUM-RGBD dataset** [34]**.** '-' means that traditional method fails in that sequence. The best performance of each block is in **bold**, and the second best is underlined.

	Method	Seq. 1	Seq. 2	Seq. 3	Seq. 4	Seq. 5	Seq. 6	Seq. 7	Seq. 8	Seq. 9	Seq. 10	Avg.
Geo.	ORB-SLAM2-M [24]	**0.041**	**0.184**	-	-	-	-	-	**0.057**	-	**0.018**	-
	DSO [8]	-	0.197	-	**0.737**	-	**0.082**	-	0.093	**0.543**	0.040	-
Sup.	BeyondTracking [47]	0.153	0.208	**0.056**	**0.070**	0.172	0.015	**0.123**	**0.007**	**0.035**	**0.042**	**0.088**
	DeepV2D [36]	0.232	0.651	0.186	0.167	**0.171**	0.029	0.435	0.106	0.085	0.082	0.214
	DeepV2D (aligned) [36]	**0.087**	0.300	0.114	0.106	0.181	**0.013**	0.380	0.110	0.094	0.098	0.148
Self-Sup.	Ours	0.192	0.190	0.083	0.122	0.177	0.016	0.219	0.102	0.179	0.107	0.139

the first LSTM. The cycle consistency constraint between the two-layer LSTM estimations serves as a mini *loop closure* to enforce the transitivity consistency of poses. The second stage of training allows our network to explicitly optimize over *long* sequences, which resembles the *motion-only bundle adjustment module*. We combine the best of both geometry and learning by building a self-supervised VO framework whose components (network, loss, training scheme) are fully inspired by the well-studied geometric modules. As verified in our experiments, these geometry inspired designs lead to significantly better results than the existing self-supervised baselines.

Limitations. Although the proposed system achieves a good camera pose estimation performance in terms of translation error, the improvement on the rotation prediction is not as substantial as the translation. We conjecture that the large rotation error may be due to the bias within the training data. Specifically, for driving scenarios, the translational motions occur much more frequently than the rotational ones. Training on more diverse video sequences or synthetic data could potentially alleviate the inherent bias in the existing datasets. Also, we observe that our system fails under the over-exposure scenarios since our method still relies on visual input to extract information.

5 Conclusions

In this work, we learn a monocular visual odometry system in a self-supervised manner to mimic critical modules in traditional geometric methods. We first adopt a two-layer convolutional LSTM module to model the long-term dependency in the pose estimation. To allow the network to see beyond *short* snippets (e.g., 3 or 5 frames) during the training time, we propose a stage-wise training strategy. Combining the recurrent architecture and the proposed decoupled training scheme, our system achieves state-of-the-art performance among self-supervised methods. In the current form, we do not have a mechanism to detect loops and perform full loop closure. In the future, we plan to study how to incorporate it into our learning framework.

Acknowledgment. This work was part of Y. Zou's internship at NEC Labs America, in San Jose. Y. Zou and J.-B. Huang were also supported in part by NSF under Grant No. (#1755785).

References

1. Bian, J.W., 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: NeurIPS (2019) 4, 10, 11
2. Bloesch, M., Czarnowski, J., Clark, R., Leutenegger, S., Davison, A.J.: Codeslam—learning a compact, optimisable representation for dense visual slam. In: CVPR (2018) 4
3. Bogdan, O., Eckstein, V., Rameau, F., Bazin, J.C.: Deepcalib: a deep learning approach for automatic intrinsic calibration of wide field-of-view cameras. In: CVMP (2018) 2
4. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on Robotics **32**(6), 1309–1332 (2016) 3
5. Chorowski, J., Bahdanau, D., Cho, K., Bengio, Y.: End-to-end continuous speech recognition using attention-based recurrent nn: First results. arXiv preprint arXiv:1412.1602 (2014) 4
6. Dhiman, V., Tran, Q.H., Corso, J.J., Chandraker, M.: A continuous occlusion model for road scene understanding. In: CVPR (2016) 9
7. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: ICCV (2015) 5, 9
8. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. TPAMI **40**(3), 611–625 (2017) 2, 6, 12, 14
9. Engel, J., Schöps, T., Cremers, D.: Lsd-slam: Large-scale direct monocular slam. In: ECCV (2014) 2, 6
10. Forster, C., Pizzoli, M., Scaramuzza, D.: Svo: Fast semi-direct monocular visual odometry. In: ICRA (2014) 2
11. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. IJRR **32**(11), 1231–1237 (2013) 9, 11
12. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012) 9, 10, 11
13. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.: Digging into self-supervised monocular depth estimation. In: ICCV (2019) 2, 4, 5, 6, 9, 10, 11, 12, 13
14. Graves, A., Jaitly, N.: Towards end-to-end speech recognition with recurrent neural networks. In: ICML (2014) 4
15. Grupp, M.: evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo (2017) 9
16. Hartley, R.I., Sturm, P.: Triangulation. CVIU **68**(2), 146–157 (1997) 3
17. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2014) 9
18. Klein, G., Murray, D.: Parallel tracking and mapping for small ar workspaces. In: ISMAR (2007) 3
19. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g 2 o: A general framework for graph optimization. In: ICRA (2011) 8
20. Li, R., Wang, S., Long, Z., Gu, D.: Undeepvo: Monocular visual odometry through unsupervised deep learning. In: ICRA (2018) 2, 4, 10, 11
21. Li, Y., Ushiku, Y., Harada, T.: Pose graph optimization for unsupervised monocular visual odometry. In: ICRA (2019) 10, 11
22. Mahjourian, R., Wicke, M., Angelova, A.: Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In: CVPR (2018) 10, 11

23. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE Transactions on Robotics **31**(5), 1147–1163 (2015) 2, 3, 6, 9
24. Mur-Artal, R., Tardós, J.D.: ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017) 2, 6, 10, 11, 12, 14
25. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry. In: CVPR (2004) 3
26. 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: CVPR (2019) 2, 4, 10, 11, 12, 13
27. Scaramuzza, D., Fraundorfer, F.: Visual odometry [tutorial]. IEEE Robotics & Automation Magazine **18**(4), 80–92 (2011) 2, 3
28. Schubert, D., Demmel, N., Usenko, V., Stuckler, J., Cremers, D.: Direct sparse odometry with rolling shutter. In: ECCV (2018) 2
29. Shen, T., Luo, Z., Zhou, L., Deng, H., Zhang, R., Fang, T., Quan, L.: Beyond photometric loss for self-supervised ego-motion estimation. In: ICRA (2019) 10, 11, 12
30. Sheng, L., Xu, D., Ouyang, W., Wang, X.: Unsupervised collaborative learning of keyframe detection and visual odometry towards monocular deep slam. In: ICCV (2019) 4
31. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: NeurIPS (2015) 6
32. Song, S., Chandraker, M.: Robust scale estimation in real-time monocular sfm for autonomous driving. In: CVPR (2014) 9
33. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: ICML (2015) 4
34. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: IROS (2012) 9, 14
35. Tang, C., Tan, P.: Ba-net: Dense bundle adjustment network. In: ICLR (2019) 4
36. Teed, Z., Deng, J.: Deepv2d: Video to depth with differentiable structure from motion. In: ICLR (2020) 4, 10, 11, 12, 14
37. Tiwari, L., Ji, P., Tran, Q.H., Zhuang, B., Anand, S., Chandraker, M.: Pseudo rgb-d for self-improving monocular slam and depth prediction. In: ECCV (2020) 3
38. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: International Workshop on Vision Algorithms (1999) 2
39. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. In: CVPR (2017) 4
40. Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., Lee, H.: Learning to generate long-term future via hierarchical prediction. In: ICML (2017) 4
41. Wang, C., Miguel Buenaposada, J., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. In: CVPR (2018) 6
42. Wang, R., Pizer, S.M., Frahm, J.M.: Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In: CVPR (2019) 2, 4, 10, 11
43. Wang, R., Schworer, M., Cremers, D.: Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In: ICCV (2017) 6
44. Wang, S., Clark, R., Wen, H., Trigoni, N.: Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In: ICRA (2017) 2, 4, 10, 11

45. Wang, S., Clark, R., Wen, H., Trigoni, N.: End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. IJRR **37**(4-5), 513–542 (2018) 2, 4, 10, 11
46. Xue, F., Wang, Q., Wang, X., Dong, W., Wang, J., Zha, H.: Guided feature selection for deep visual odometry. In: ACCV (2018) 2, 4, 10, 11
47. Xue, F., Wang, X., Li, S., Wang, Q., Wang, J., Zha, H.: Beyond tracking: Selecting memory and refining poses for deep visual odometry. In: CVPR (2019) 2, 4, 5, 7, 9, 10, 11, 12, 14
48. Yang, N., Wang, R., Gao, X., Cremers, D.: Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect. RA-L **3**(4), 2878–2885 (2018) 2
49. Yin, Z., Shi, J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In: CVPR (2018) 10, 11, 12
50. Zhan, H., Garg, R., Saroj Weerasekera, C., Li, K., Agarwal, H., Reid, I.: Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In: CVPR (2018) 2, 4, 10, 11
51. Zhou, H., Ummenhofer, B., Brox, T.: Deeptam: Deep tracking and mapping. In: ECCV (2018) 4
52. Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: CVPR (2017) 2, 3, 4, 5, 9, 10, 11, 12
53. Zhuang, B., Tran, Q.H., Ji, P., Cheong, L.F., Chandraker, M.: Learning structure-and-motion-aware rolling shutter correction. In: CVPR (2019) 2
54. Zhuang, B., Tran, Q.H., Lee, G.H., Cheong, L.F., Chandraker, M.: Degeneracy in self-calibration revisited and a deep learning solution for uncalibrated slam. In: IROS (2019) 2
55. Zou, Y., Luo, Z., Huang, J.B.: DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In: ECCV (2018) 2