

# Learning by Distillation: A Self-Supervised Learning Framework for Optical Flow Estimation

Pengpeng Liu <sup>1</sup>, Michael R. Lyu, *Fellow, IEEE*, Irwin King <sup>2</sup>, *Fellow, IEEE*, and Jia Xu, *Fellow, IEEE*

**Abstract**—We present DistillFlow, a knowledge distillation approach to learning optical flow. DistillFlow trains multiple teacher models and a student model, where challenging transformations are applied to the input of the student model to generate hallucinated occlusions as well as less confident predictions. Then, a self-supervised learning framework is constructed: confident predictions from teacher models are served as annotations to guide the student model to learn optical flow for those less confident predictions. The self-supervised learning framework enables us to effectively learn optical flow from unlabeled data, not only for non-occluded pixels, but also for occluded pixels. DistillFlow achieves state-of-the-art unsupervised learning performance on both KITTI and Sintel datasets. Our self-supervised pre-trained model also provides an excellent initialization for supervised fine-tuning, suggesting an alternate training paradigm in contrast to current supervised learning methods that highly rely on pre-training on synthetic data. At the time of writing, our fine-tuned models ranked 1st among all monocular methods on the KITTI 2015 benchmark, and outperform all published methods on the Sintel Final benchmark. More importantly, we demonstrate the generalization capability of DistillFlow in three aspects: framework generalization, correspondence generalization and cross-dataset generalization. Our code and models will be available on <https://github.com/ppliuboy/DistillFlow>.

**Index Terms**—Optical flow, knowledge distillation, unsupervised learning, self-supervised learning, and stereo matching

## 1 INTRODUCTION

OPTICAL flow, which describes the dense pixel correspondence between two adjacent images, has a wide range of applications such as autonomous driving [1], object tracking [2] and video-related tasks [3], [4], [5], [6]. Traditional variational approaches [7], [8], [9] formulate optical flow estimation as an energy minimization problem, which are usually computationally expensive [10] and not applicable for real-time applications.

Benefited from the development of deep learning, convolutional neural networks (CNNs) have been successfully applied to optical flow estimation [11], [12], [13], which achieve comparable or even better performance compared with traditional methods, while running at real-time. Similar to other deep learning tasks, a large amount of labeled training data is required to train these fully supervised CNNs with high performance. However, it is extremely difficult to obtain the ground truth of optical flow for real-world image pairs, especially when there are occlusions. Due to lacking large-scale real-world annotations, existing methods highly rely on pre-training on synthetic labeled datasets [14], [15]. However, there usually exists a large domain gap between the distribution of synthetic data and natural scenes. In order to train a flow

model stably, we have to carefully follow specific training schedules across different datasets [11], [12], [13], [16], [17], [18]. This raises our first question: *can we simplify the training procedure and remove the need of synthetic dataset for supervised optical flow estimation?*

Another promising direction is to develop unsupervised methods [19], [20], [21], [22], [23], where unlimited image sequences are readily available. The basic idea is based on image warping, where the target image is warped toward the reference image according to the estimated flow, then a photometric loss is defined to minimize the difference between the reference image and the warped target image. However, brightness constancy assumption does not hold for occluded regions. Therefore, such photometric loss only works well for non-occluded pixels, but turns to provide misleading information for occluded pixels. To alleviate the above issue, recent methods propose to exclude occluded pixels when computing the photometric loss or employ additional spatial and temporal smoothness terms to regularize the flow estimations [21], [22], [23]. However, all these methods rely on hand-crafted energy terms to regularize flow estimation, lacking the key capability to effectively learn optical flow of occluded pixels. As a result, there is still a large performance gap comparing these methods with state-of-the-art fully supervised methods. This raises our second question: *can we effectively learn optical flow with occlusions without relying on any labeled data and reduce the performance gap?*

In this paper, we address the above two questions with a two-stage self-supervised learning approach based on knowledge distillation. As shown in Fig. 1, the proposed framework contains two kinds of models: teacher model and student model, and two kinds of distillation: model distillation (stage 1) and data distillation (stage 2). In stage 1, we train teacher

• Pengpeng Liu, Michael R. Lyu, and Irwin King are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. E-mail: {ppliuboy, lyu, king}@cse.cuhk.edu.hk.

• Jia Xu is with the Huya AI, Guangzhou, Guangdong 511442, China. E-mail: xujia@huya.com.

Manuscript received 6 July 2020; revised 30 March 2021; accepted 14 May 2021.

Date of publication 1 June 2021; date of current version 4 August 2022.

(Corresponding author: Pengpeng Liu.)

Recommended for acceptance by E. B. Sudderth.

Digital Object Identifier no. 10.1109/TPAMI.2021.3085525

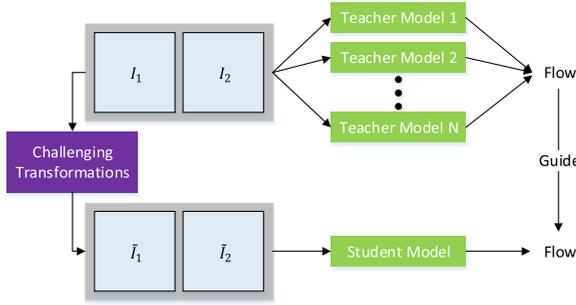


Fig. 1. Framework illustration. We distill confident optical flow estimations from teacher models (stage 1) to guide the flow learning of the student model (stage 2) under different challenging transformations.

models following a similar training protocol as previous methods [21], [22], [23] with occlusion handling, aiming at estimating accurate optical flow for those non-occluded pixels. We employ model distillation to ensemble predictions from multiple teacher models, which can reduce the variance of single teacher model prediction. The confident predictions from teacher models serve as annotations to guide the learning of the student model in the second stage. For data distillation, we create challenging transformations to the input image pairs, which are used to generate hallucinated occlusions as well as less confident predictions for self-supervision. As shown in Fig. 2, we show occlusion hallucination techniques that generate hand-crafted occlusions, e.g., pixel  $p_1$  is non-occluded from  $I_1$  to  $I_2$  but becomes occluded from  $\tilde{I}_1$  to  $\tilde{I}_2$ . Then we can let teacher models guide the student model to learn the optical flow of these hand-crafted occluded pixels. As a result, the distillation scheme enables our student model to effectively learn the optical flow of occluded pixels. The occlusion hallucination techniques also create less confident predictions, e.g., although  $p_2$  is non-occluded, the patch similarity of pixel  $p_2$  between  $I_1$  and  $\tilde{I}_2$  is smaller than the patch similarity between  $I_1$  and  $I_2$ , due to the partially missing regions. In this case, the distillation scheme helps our student model learn optical flow of non-occluded pixels more effectively. Apart from these occlusion

hallucination techniques, we create other challenging transformations, including geometric transformations (e.g., scaling) and color transformations (e.g., overexposure and underexposure). Overall, our distillation scheme lets confident predictions from teacher models to supervise less confident predictions from the student model, enabling the student model to have the improved ability to learn optical flow of both occluded and non-occluded pixels. Our method, termed as DistillFlow, outperforms all previous unsupervised methods by a large margin on all datasets, and is comparable with fully supervised methods, which significantly reduces the performance gap. Hence, the problem in question 2 is successfully addressed.

More importantly, our self-supervised pre-trained model provides an excellent initialization for supervised fine-tuning. We can use self-supervised pre-training on unlabeled data to replace pre-training on multiple synthetic datasets. We also extend the distillation idea from unsupervised learning to semi-supervised learning, which further improves the fine-tuning performance. At the time of writing, our fine-tuned models outperform all monocular methods on the KITTI 2015 benchmark, and outperform all published methods on the Sintel Final benchmark. Hence, the problem in question 1 is successfully addressed.

Furthermore, we demonstrate the generalization capability of DistillFlow in three aspects: framework generalization, correspondence generalization and cross-dataset generalization. In framework generalization, we first show that our knowledge distillation framework is applicable to different network structures (e.g., PWC-Net [12], FlowNetS [14] and FlowNetC [14]), then extend the knowledge distillation idea to semi-supervised learning. For correspondence generalization, we directly use our self-supervised flow model trained on monocular videos to estimate stereo disparity. Surprisingly, DistillFlow achieves comparable performance with state-of-the-art unsupervised stereo matching methods on KITTI datasets. For cross-data generalization, we evaluate the performance of the model trained on another dataset

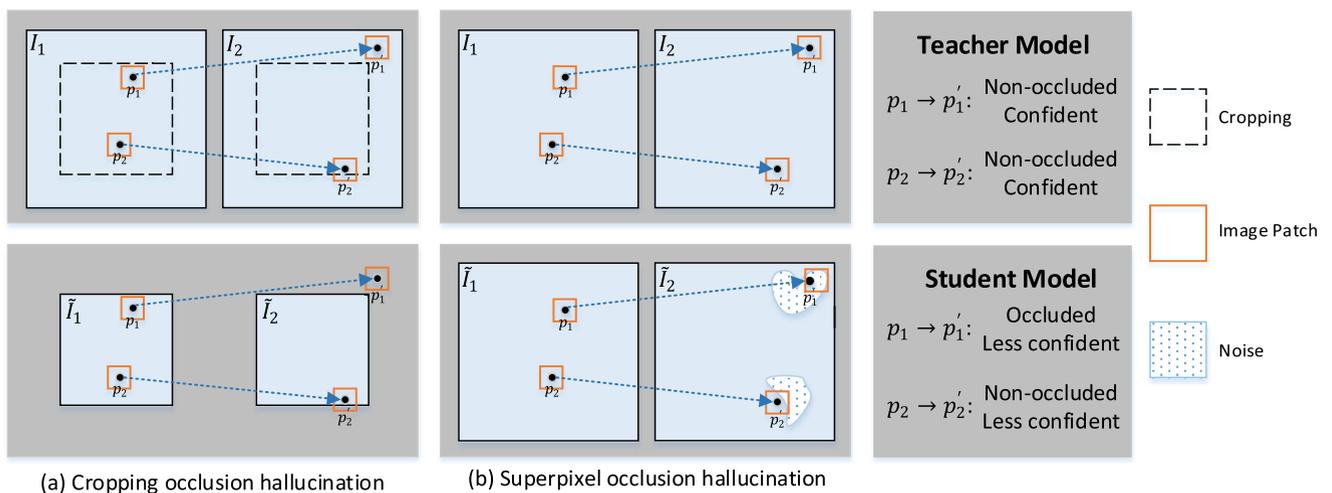


Fig. 2. Occlusion hallucination scheme. The scheme creates hand-crafted occlusions, e.g., pixel  $p_1$  is non-occluded from  $I_1$  to  $I_2$  but becomes occluded from  $\tilde{I}_1$  to  $\tilde{I}_2$  ( $p_1$  moves out of the image boundary for case (a) and is covered by noise for case (b)). The scheme also creates less confident predictions, e.g., though pixel  $p_2$  is non-occluded, its patch similarity between  $\tilde{I}_1$  and  $\tilde{I}_2$  is smaller than patch similarity between  $I_1$  and  $I_2$  due to the partially missing regions.

(e.g., Sintel  $\rightarrow$  KITTI and KITTI  $\rightarrow$  Sintel), and show that DistillFlow still achieves comparable performance with previous methods.

We summarize our main contributions as follows:

- We present a two-stage self-supervised learning approach based on knowledge distillation to effectively learn optical flow of both occluded and non-occluded pixels from unlabeled data. Our method significantly outperforms previous unsupervised methods, especially for occluded pixels.
- We improve the training protocol compared with our previous works DDFlow [24], SelfFlow [25] and Flow2-Stereo [26] by adding spatial regularizer and model distillation.
- Our self-supervised pre-trained models provide excellent initializations for supervised fine-tuning, which removes the reliance of pre-training on multiple synthetic datasets. After fine-tuning, we achieve state-of-the-art supervised learning performance.
- We demonstrate the generalization capability of DistillFlow in three aspects: framework generalization, correspondence generalization and cross-dataset generalization.

## 2 RELATED WORK

*Traditional Flow Methods.* Traditional variational methods formulate optical flow estimation as an energy minimization problem based on brightness constancy and spatial smoothness [7], [27]. These methods work well for small motion, but usually fail when displacements are large. Later works [9], [28] integrate feature matching to tackle this issue. Specifically, they first find sparse feature correspondences to initialize flow estimation and then refine it in a coarse-to-fine manner. The seminal work EpicFlow [29] interpolates dense flow from sparse matches and has become a widely used post-processing pipeline. There are also some works that use temporal information over multiple frames to improve the robustness and accuracy by adding temporal constraints, such as constant velocity [30], [31], [32], constant acceleration [33], [34] and so on. Recently, [10], [35] use convolutional neural networks (CNNs) to learn a feature embedding for better matching and have demonstrated superior performance. However, these methods are often computationally expensive and can not be trained end-to-end. In this paper, we use CNNs to directly estimate optical flow in an end-to-end manner, which is very efficient.

*Supervised Flow Methods.* Inspired by the development of deep neural networks, CNNs have been successfully applied to optical flow estimation. The pioneering work FlowNet [14] proposes two types of CNN, FlowNetS and FlowNetC, which take two consecutive images as input and output a dense optical flow map. The follow-up FlowNet 2.0 [11] stacks several basic FlowNet models and refines the flow iteratively, which significantly improves accuracy. SpyNet [36] proposes a lightweight network architecture by employing image warping at different scales in a coarse-to-fine manner. However, its performance is behind the state-of-the-art. PWC-Net [12] and LiteFlowNet [13] propose to warp CNN features instead of images at different scales and introduce cost volume construction,

achieving state-of-the-art performance with compact model size. They were further improved by only using a single network block with shared weights to iteratively refine flow at different scales and adding occlusion reasoning [16]. VCN [37] introduces efficient volumetric networks for dense 2D correspondence matching by exploring high-dimensional invariance during cost volume computation. MaskFlowNet [18] proposes an asymmetric occlusion-aware feature matching module, which masks out those occluded regions after feature warping. ScopeFlow [17] introduces an improved training protocol by fully utilizing cropping randomly sized scene scopes. However, due to lacking of real-world ground truth optical flow, all above supervised learning methods highly rely on pre-training on synthetic datasets (e.g., FlyingChairs [14] and FlyingThings3D [15]) and follow specific training schedules. In this paper, we propose to employ self-supervised pre-training on unlabeled image sequences to achieve excellent initializations, which remove the reliance of pre-training on synthetic datasets.

*Unsupervised Flow Methods.* Due to lacking ground truth optical flow for natural image sequences, recent studies turn to formulate optical flow estimation as an unsupervised learning problem based on the brightness constancy and spatial smoothness assumption [19], [20]. The basic idea is based on image warping to achieve view synthesis, and then define a photometric loss measure the difference between the reference image and the synthesized warped target image. However, brightness constancy does not hold for occluded pixels, therefore [21], [22], [23] propose to detect occlusion and exclude occluded pixels when computing photometric loss. EpipolarFlow [38] proposes to incorporate global geometric epipolar constraints into network learning to improve performance. There are also works that propose to jointly learn flow and depth from monocular videos [39], [40], [41], [42], [43] or jointly learn flow and disparity from stereoscopic videos [44], [45]. Despite promising progress, they still lack the key ability to effectively learn optical flow of occluded pixels and their performance is far behind state-of-the-art supervised learning methods. In this paper, we propose a knowledge distillation based self-supervised learning method to learn optical flow of both occluded and non-occluded pixels in an unsupervised manner.

*Unsupervised Stereo Methods.* Optical flow estimation and stereo matching can be viewed as a unified problem: correspondence matching. For rectified stereo image pairs, the stereo disparity can be regarded as a special case of optical flow in the horizontal direction. Therefore, our method is also related to a large body of unsupervised stereo learning methods, including image synthesis and warping with depth estimation [46], left-right consistency [47], [48], [49], employing additional semantic information [50], cooperative learning [51], self-adaptive fine-tuning [52], [53], [54]. Different from all these methods that train stereo models on stereo image pairs, we directly use our optical flow model trained on monocular videos to estimate disparity (only keep the horizontal component of flow). We demonstrate that our self-supervised flow model has the strong generalization capability to find correspondences.

*Knowledge Distillation and Self-Supervised Learning.* Our work is closely related to the family of self-supervised learning methods, where the supervision signal is purely generated

from the data itself. It is widely used for learning feature representations from unlabeled data [55]. A pretext task is usually employed, such as image inpainting [56], image colorization [57], solving Jigsaw puzzles [58]. Pathak *et al.* [59] propose to explore low-level motion-based cues to learn feature representations without manual supervision. Doersch *et al.* [60] combine multiple self-supervised learning tasks to train a single visual representation. To obtain more reliable supervision signals, we employ model distillation [61] to ensemble predictions of several teacher models. Then we make use of the domain knowledge of optical flow and employ data distillation for self-supervision. Our data distillation is different from [62], which ensembles predictions from a single model applied to multiple transformations of unlabeled image pairs as annotations. Instead, we create challenging image transformations to create hallucinated occlusions and less confident predictions to enable effective self-supervision.

### 3 METHOD

In this section, we illustrate our self-supervised learning framework based on knowledge distillation. We train two types of CNNs (multiple teacher models and a student model) with the same network architecture. Only the student model is needed during testing. The self-supervised learning framework contains two stages: in stage 1, we train teacher models to obtain confident flow predictions; in stage 2, we distill confident predictions from the teacher models to guide the student model to learn optical flow of both occluded and non-occluded pixels. We introduce two variants in stage 2, where variant 1 is from the occlusion view and variant 2 is from the confidence view. In principle, DistillFlow can use any backbone network to learn optical flow. In our implementation, we adopt IRR-PWC [16], which is a variant of PWC-Net [12] with weight sharing across different levels. We additionally add dilated convolution to increase the receptive field. Before describing our method in detail, we first define our notations.

#### 3.1 Notation

For our teacher models, we denote  $I_1, I_2 \in \mathbb{R}^{H \times W \times 3}$  for two consecutive RGB images, where  $H$  and  $W$  are height and width respectively. Our goal is to estimate the forward optical flow  $\mathbf{w}_f \in \mathbb{R}^{H \times W \times 2}$  from  $I_1$  to  $I_2$ . After obtaining  $\mathbf{w}_f$ , we can warp  $I_2$  towards  $I_1$  to get the warped image  $I_{2 \rightarrow 1}^w$ . Here, we also estimate the backward optical flow  $\mathbf{w}_b$  from  $I_2$  to  $I_1$  and a backward warped image  $I_{1 \rightarrow 2}^w$ . Since there are many cases where one pixel is only visible in one image but not visible in the other image, namely occlusion, we denote  $O_f, O_b \in \mathbb{R}^{H \times W \times 1}$  as the forward and backward occlusion map respectively. The occlusion map is a binary mask, where value 1 denotes that the pixel in that location is occluded and value 0 denotes non-occluded.

After creating challenging transformations to the input image pairs, the input images to the student model are denoted as  $\tilde{I}_1, \tilde{I}_2 \in \mathbb{R}^{h \times w \times 3}$ . Similarly, optical flow, occlusion map and confidence map from teacher models need to perform corresponding transformations as input images. We use  $\tilde{\mathbf{w}}_f^T, \tilde{\mathbf{w}}_b^T, \tilde{O}_f^T, \tilde{O}_b^T, \tilde{M}_f^T$  and  $\tilde{M}_b^T$  to denote their transformed results.

Our student model follows similar notations. The student network takes  $\tilde{I}_1, \tilde{I}_2$  as input, and produces optical flow  $\tilde{\mathbf{w}}_f, \tilde{\mathbf{w}}_b$ , warped images  $\tilde{I}_{2 \rightarrow 1}^w, \tilde{I}_{1 \rightarrow 2}^w$ , occlusion maps  $\tilde{O}_f, \tilde{O}_b$ .

For stage 2 variant 1, knowledge distillation is performed from occlusion view. Therefore, we define another occlusion map  $O'_f$  and  $O'_b$  to denote hallucinated occlusions (i.e., hand-crafted occlusions). Hallucinate occlusion map is computed from the transformed occlusion mask  $O^T$  (in our teacher models) and occlusion mask  $\tilde{O}$  (in our student model).

For stage 2 variant 2, knowledge distillation is performed from the confidence view. Therefore, we define confidence maps  $M_f^T$  and  $M_b^T$  to denote which pixels can be accurately estimated by our teacher models after transformations. For confidence maps, value 1 denotes that flow prediction of that pixel is confident and value 0 denotes not confident.

#### 3.2 Stage 1: Unsupervised Flow Learning

To train our teacher models in an unsupervised manner, we swap the image pairs in our input to produce both forward flow  $\mathbf{w}_f$  and backward flow  $\mathbf{w}_b$ . After that, we estimate an occlusion map based on the forward-backward consistency prior [21], [63]. That is, for non-occluded pixels, the forward flow should be the inverse of the backward flow at the corresponding pixel in the second image. We consider pixels as occluded when the mismatch between forward flow and backward flow is too large. Take forward occlusion map as an example, we first compute the reversed forward flow as follow:

$$\hat{\mathbf{w}}_f(\mathbf{p}) = \mathbf{w}_b(\mathbf{p} + \mathbf{w}_f(\mathbf{p})), \quad (1)$$

where  $p$  is a pixel in the first image  $I_1$ . A pixel is considered occluded (i.e.,  $O_f(\mathbf{p}) = 1$ ) when the following constraint is violated:

$$|\mathbf{w}_f(\mathbf{p}) + \hat{\mathbf{w}}_f(\mathbf{p})|^2 < \alpha_1(|\mathbf{w}_f(\mathbf{p})|^2 + |\hat{\mathbf{w}}_f(\mathbf{p})|^2) + \alpha_2, \quad (2)$$

where we set  $\alpha_1 = 0.01$ ,  $\alpha_2 = 0.5$  for all our experiments. Backward occlusion map  $\mathbf{w}_b$  is computed in the same way.

Unsupervised flow estimation is based on brightness constancy and spatial smoothness assumption. We use photometric loss  $L_{pho}$  and edge-aware smoothness loss  $L_{smo}$  for the above two assumptions. Photometric loss measures the difference between the reference image and the warped target image. Take forward flow  $\mathbf{w}_f$  as an example, we can use  $\mathbf{w}_f$  to warp  $I_2$  to reconstruct  $I_1$ :

$$I_{2 \rightarrow 1}^w(\mathbf{p}) = I_2(\mathbf{p} + \mathbf{w}_f(\mathbf{p})). \quad (3)$$

Photometric loss  $L_{pho}$  only makes sense for non-occluded pixels, which is defined as follows:

$$L_{pho} = \sum \psi(I_1 - I_{2 \rightarrow 1}^w) \odot (1 - O_f) / \sum (1 - O_f) + \sum \psi(I_2 - I_{1 \rightarrow 2}^w) \odot (1 - O_b) / \sum (1 - O_b) \quad (4)$$

where  $\psi(x) = (|x| + \epsilon)^q$  is a robust loss function,  $\odot$  denotes the element-wise multiplication. During our experiments, we set  $\epsilon = 0.01$ ,  $q = 0.4$ .

Photometric loss is not informative in homogeneous regions, therefore existing unsupervised methods usually add a smoothness loss to regularize the flow [21], [22]. Smoothness can be regarded as a regularizer for occluded

pixels, since it makes the prediction of occluded pixels similar to the neighborhood pixels. Here we adopt an edge-aware smoothness loss weighted by the image gradient:

$$L_{smo} = \frac{1}{H \times W} \sum_{\mathbf{p}} |e^{-\beta \nabla I_1(\mathbf{p})}|^T \cdot |\nabla \mathbf{w}_f(\mathbf{p})| + \frac{1}{H \times W} \sum_{\mathbf{p}} |e^{-\beta \nabla I_2(\mathbf{p})}|^T \cdot |\nabla \mathbf{w}_b(\mathbf{p})|, \quad (5)$$

where  $\nabla$  denotes gradient,  $T$  represents transpose and  $\beta$  is a factor to control the smoothness effect on edges. We set  $\beta = 10$  in our experiment.

The final loss to train teacher models in stage 1 is the combination of  $L_{pho}$  and  $L_{smo}$ :

$$L_1 = L_{pho} + 0.1 * L_{smo}. \quad (6)$$

### 3.3 Stage 2: Self-Supervised Flow Learning with Knowledge Distillation

Since photometric loss does not make sense for occluded pixels, prior unsupervised methods lack the key ability to effectively learn optical flow of occluded pixels. To tackle this issue, we distill confident predictions from our teacher models, and use them to generate input/output data for our student model by creating challenging transformations. Next, we first introduce our occlusion hallucination techniques, then describe challenging transformations employed in DistillFlow, finally we explain two variants of knowledge distillation.

#### 3.3.1 Occlusion Hallucination

Fig. 2 demonstrates two kinds of occlusion hallucination techniques used in DistillFlow: random cropping and random superpixel noise injection. In both Figs. 2a and 2b, suppose pixel  $p_1$  is non-occluded from  $I_1$  to  $I_2$  and pixel  $p'_1$  is its corresponding pixel. After creating challenging transformation to  $I_1$  and  $I_2$ ,  $p_1$  becomes occluded from  $\tilde{I}_1$  to  $\tilde{I}_2$ , because  $p'_1$  moves out of image boundary in (a) and is covered by noise in (b). We call the above procedures of creating hand-crafted occlusions as occlusion hallucination.

Even though  $p_1$  becomes occluded from  $\tilde{I}_1$  to  $\tilde{I}_2$ , the location of its corresponding pixel  $p'_1$  does not change. As a result, the flow of  $p_1$  does not change during occlusion hallucination. This is the basic assumption of our knowledge distillation idea. Since  $p_1$  is non-occluded from  $I_1$  to  $I_2$ , our teacher models can accurately estimate its flow; however,  $p_1$  becomes occluded from  $\tilde{I}_1$  to  $\tilde{I}_2$ , therefore our student model cannot accurately estimate its flow anymore. Luckily, we can distill confident flow estimation of  $p_1$  from teacher models to guide the student model to learn the flow of  $p_1$  from  $\tilde{I}_1$  to  $\tilde{I}_2$ . This explains why our knowledge distillation approach enables our student model to effectively learn optical flow of occluded pixels.

Strategy in (a) generates hallucinated occlusions near the image boundary. However, for occlusion elsewhere (e.g., motion boundary of objects), it is not so effective. Strategy in (b) can generate hallucinated occlusions in a wider range of cases. This is because the shape of a superpixel is usually random and superpixel edges are often part of object boundaries, which is consistent with the real-world cases. We can choose several superpixels at different locations to cover more occlusion cases. The combination of (a) and (b) can generate a variety of hallucinated occlusions.

#### 3.3.2 Challenging Transformations

In this part, we show that knowledge distillation also helps learn the optical flow of non-occluded pixels. We introduce three kinds of challenging transformations: occlusion hallucination-based transformations, geometric transformations and color transformations.

*Occlusion-hallucination based transformations.* When searching pixel correspondences, we care about not only the color of specific pixels, but also the color of their neighbors or context, that is, image patch similarity. In Fig. 2,  $p_2$  is non-occluded both from  $I_1$  to  $I_2$  and from  $\tilde{I}_1$  to  $\tilde{I}_2$ , and  $p'_2$  is its corresponding pixel. When considering the image patches around pixel  $p_2$  and  $p'_2$ , patch similarity from  $\tilde{I}_1$  to  $I_2$  is obviously smaller than patch similarity from  $I_1$  to  $I_2$ , due to the partially missing regions (part regions move out of the image boundary in (a) and part regions are covered by noise in (b)). In this case, the flow estimation of  $p_2$  by teacher models is more confident than the student model. Then, we can use the confident predictions from teacher models to guide the student model to learn the flow of  $p_2$ . This explains why knowledge distillation also improves the optical flow of non-occluded pixels with occlusion hallucination based transformations.

*Geometric Transformations.* Geometric transformations include cropping, scaling, rotation, translation and so on, which are defined by 6 parameters as in the affine transformation from Spatial Transformer Network [64]. Cropping used in occlusion hallucination is just one kind of geometric transformation. Actually, other kinds of geometric transformations are also effective as long as they can create challenging scenes, where flow predictions become less confident after transformations. Take scaling as an example, suppose we downsample  $I_1$  and  $I_2$  as input to the student model. In general, image information will lose during the downsampling operation, which makes the student model difficult to predict flow. Therefore, the flow prediction by the student model is less confident than teacher models.

*Color Transformations.* Color transformations represent those transformations that change the appearance of images, such as changing contrast, brightness, saturation, hue, etc. Though such transformations do not create hallucinated occlusions, they can create challenging scenes. For example, generating images with overexposure and underexposure changes the image appearance, and decreasing image contrast makes pixels less distinguishable.

Overall, the purpose of creating challenging transformations is to create hallucinated occlusion or less confident predictions, so that knowledge distillation can be effectively employed. Fig. 3 shows a real-world example of different transformations. The raw flow predictions from teacher models have many erroneous pixels, but confident predictions after forward-backward consistency check are mostly correct ((d)  $\rightarrow$  (e) in row 1). After creating challenging transformations, many confident predictions become less confident (e.g., rectangle regions in (d) and (e)).

#### 3.3.3 Knowledge Distillation

Knowledge distillation is performed at stage 2 to train our student model in the self-supervision framework. We first introduce two variants of knowledge distillation, then make comparisons between them.

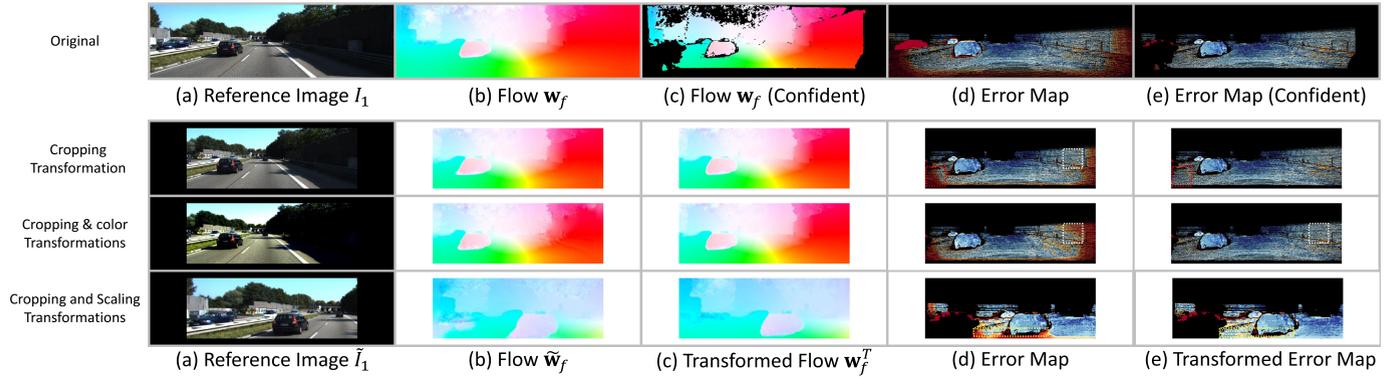


Fig. 3. Knowledge distillation examples. The redder the color in the error map, the greater the error. In the (d) of the first row, flow  $w_f$  has many erroneous pixels; however the confident flow predictions after forward-backward consistency check in (c) are mostly reliable (as shown in (e)). After creating challenging transformations to the input (e.g., row 2-4), the flow predictions by the student model are usually less confident than the transformed predictions  $w_f^T$  from confident flow, e.g., rectangle regions in the error maps. We only use confident predictions in (c) of row 1 to guide the learning of the student model. In general,  $w_f^T$  shall be sparse as in (c) of row 1. For better visual comparison with  $\tilde{w}_f$ , we show transformed results from (b) of row 1.

*Stage 2 Variant 1: From the Occlusion View.* As illustrated in the occlusion hallucination section, there exist new occlusions after creating challenging transformations. Hallucinated occlusion maps  $O'_f$  and  $O'_b$  are computed as follows:

$$\begin{cases} O'_f = \min(\max(\tilde{O}_f - O_f^T, 0), 1) \\ O'_b = \min(\max(\tilde{O}_b - O_b^T, 0), 1). \end{cases} \quad (7)$$

Then the pixels in  $\tilde{I}_1, \tilde{I}_2$  can be divided into three types: old occluded pixels (pixels that are occluded from  $I_1$  to  $I_2$ ), hallucinated occlusions (pixels that are non-occluded from  $I_1$  to  $I_2$  but become occluded from  $\tilde{I}_1$  to  $\tilde{I}_2$ ), and non-occluded pixels both from  $I_1$  to  $I_2$  and from  $\tilde{I}_1$  to  $\tilde{I}_2$ ).

As shown in Fig. 2, we define the loss for occluded pixels on hallucinated occlusions:

$$L_{occ} = \sum \psi(w_f^T - \tilde{w}_f) \odot O'_f / \sum O'_f + \sum \psi(w_b^T - \tilde{w}_b) \odot O'_b / \sum O'_b, \quad (8)$$

where  $w_f^T$  and  $w_b^T$  are the transformed flow of  $w_f$  and  $w_b$  from teacher models.

Photometric loss for non-occluded pixels and edge-aware smoothness loss are computed in the same way as the teacher models. The final loss to train our student model in stage 2 variant 1 is the combination of  $L_{phor}$ ,  $L_{occ}$  and  $L_{smo}$ :

$$L_{2,1} = L_{pho} + L_{occ} + 0.1 * L_{smo}. \quad (9)$$

*Stage 2 variant 2: from the confidence view.* In stage 1, forward-backward consistency check is employed to detect whether the pixel is occluded or not. However, this brings in errors because many pixels are still non-occluded even when they violate this principle, and vice versa. Instead, it would be more proper to call those pixels confident if they pass the forward-backward consistency check. From this point of view, the key point of knowledge distillation is to let confident predictions to supervise those less confident predictions. We define confidence map  $M$  as the reverse of occlusion map  $O$ :

$$\begin{cases} M_f^T = 1 - O_f^T \\ M_b^T = 1 - O_b^T \end{cases} \quad (10)$$

When creating challenging transformations, both occluded regions and non-occluded regions become more challenging. During the self-supervised learning stage, the student model is able to handle more challenging conditions. As a result, its performance improves not only for those occluded pixels, but also for non-occluded pixels.

We define knowledge distillation loss  $L_{dis}$  as follows:

$$L_{dis} = \sum \psi(w_f^T - \tilde{w}_f) \odot M_f^T / \sum M_f^T + \sum \psi(w_b^T - \tilde{w}_b) \odot M_b^T / \sum M_b^T. \quad (11)$$

The final loss to train our student model in stage 2 variant 2 is the combination of  $L_{dis}$  and  $L_{smo}$ :

$$L_{2,2} = L_{dis} + 0.1 * L_{smo}. \quad (12)$$

*Comparison of two variants.*

In our experiments, variant 1 and variant 2 achieve very similar performances. However, it takes more time to train variant 1 than variant 2. This is because for variant 1, apart from  $L_{occ}$ , we still need to compute  $L_{phor}$  which is time-consuming. While for variant 2, we directly train it in a supervised manner after pre-computing pseudo flow and confidence maps. As a result, we set variant 2 as our default knowledge distillation strategy. However, its performance is limited by the prediction of the teacher model. To obtain more reliable predictions from stage 1, we employ model distillation to ensemble flow predictions from multiple teacher models.

In variant 2, challenging transformations create less confident predictions for both occluded and non-occluded pixels, therefore knowledge distillation improves the flow learning of both. However, variant 1 employs  $L_{occ}$  and  $L_{pho}$  to optimize occluded and non-occluded pixels individually. It is surprising how variant 1 achieves performance improvement over non-occluded pixels. We suspect the main reason is that forward-backward consistency check cannot accurately detect whether the pixel is occluded or not (e.g., some non-occluded pixels are regarded as occluded by forward-backward consistency check). In other words, distillation loss  $L_{occ}$  still optimizes a part of non-occluded pixels.

### 3.4 Supervised Fine-Tuning

After pre-training on unlabeled datasets, we use real-world annotated data for fine-tuning. Since there are only annotations for forward flow  $\mathbf{w}_f$ , we do not swap input image pairs. Suppose that the ground truth flow is  $\mathbf{w}_f^{gt}$ , and mask  $V$  denotes whether the pixel has a label, where value 1 means that the pixel has a valid ground truth flow. Then we can obtain the supervised fine-tuning loss as follow:

$$L_{sup} = \sum (\psi(\mathbf{w}_f^{gt} - \mathbf{w}_f) \odot V) / \sum V. \quad (13)$$

During fine-tuning, We first initialize the model with the self-supervised pre-trained student on each dataset, then optimize it using  $L_{sup}$ . Inspired by the knowledge distillation from unsupervised flow learning, we extend the idea of distillation to semi-supervised learning. That is, after supervised fine-tuning, we can compute reliable flow and confidence maps for unlabeled data, which are denoted as self-annotated data. Then we mix the real annotated data and self-annotated data and train our model in a supervised manner. Note that the count of real annotated data is very limited, therefore we make a balance between real annotated data and self-annotated data. Suppose there are  $n_1$  real annotated image pairs,  $n_2$  self-annotated image pairs, we will repeat real annotated image pairs  $\frac{n_2}{n_1}$  times.

## 4 EXPERIMENT

We evaluate and compare our method with state-of-the-art unsupervised and supervised learning methods on standard optical flow benchmarks, including KITTI 2012 [71], KITTI 2015 [1] and MPI Sintel [72].

### 4.1 Implementation Details

*Datasets.* KITTI 2012 and KITTI 2015 datasets consist of real-world road scenes with sparse ground truth flow. We use the combination of their multi-view extensions raw datasets for unsupervised training, similar to [20], [22]. To avoid the mixture of training and testing data, we exclude the image pairs with ground truth flow and their neighboring frames (frame number 9-12) as in [20], [22]. For supervised fine-tuning, we use the combination of their official training pairs.

MPI Sintel is a challenging optical flow dataset that contains naturalistic video sequences. It includes a ‘clean’ version and a ‘final’ version, where the ‘final’ version is more realistic and challenging. We extract images from the Sintel movie and manually split them into 145 scenes as the raw dataset, resulting in 10,990 image pairs. For unsupervised training, we first train our model on the raw dataset, then fine-tune on the official Sintel training dataset (including both ‘clean’ and ‘final’ versions). For supervised fine-tuning, we use the combination of ‘clean’ and ‘final’ training pairs with dense annotations.

*Data Preprocessing.* We rescale the pixel value from [0, 255] to [0, 1] for unsupervised training, while normalizing each channel to be the standard normal distribution for supervised fine-tuning. This is because normalizing the image as input is more robust for illumination changing, which is especially helpful for optical flow estimation. For unsupervised training, we apply Census Transform [73] to

images when computing photometric loss, which has been proved robust for optical flow estimation [21], [74].

We employ similar data augmentation strategies with previous works [12], [13], [14], [18], including geometric augmentations (e.g., random cropping, scaling, flipping, rotation, translation) and color augmentations (e.g., random contrast, brightness, hue, saturation, gamma) for both unsupervised and supervised training. We decrease the degree of augmentations on KITTI datasets. During training, we crop  $320 \times 896$  patches for KITTI and  $384 \times 768$  patches for Sintel in all experiments. During testing, we resize the images to  $384 \times 1280$  for KITTI and  $448 \times 1024$  for Sintel.

*Training procedures.* As shown in Fig. 1, we train multiple teacher models and ensemble their predictions as annotations to obtain more reliable predictions, which will be employed to guide the learning of the student model. However, training too many models will cost a lot of computational resources. In our experiments, we make a compromise and only train two teacher models independently. For each teacher model, we use the last five checkpoints to obtain five flow predictions. As a result, our flow annotations from teacher models are the average of ten predictions.

We train our model with Adam optimizer and set the batch size to be 4 for all experiments. To avoid the trivial solution that all pixels are regarded occluded, we pre-train teacher models for 200k iterations before unsupervised training, where photometric loss is applied to all pixels (including both non-occluded pixels and occluded pixels).

For unsupervised training, we set the initial learning rate as  $10^{-4}$  and disrupt the learning rate as suggested by [66] for a better minimum. In stage 1, we train teacher models with  $L_1$ , the combination of photometric loss with occlusion handling  $L_{pho}$  and edge-aware smoothness loss  $L_{smo}$  for 600k iterations. Then we generate the ensemble flow predictions and confidence maps using teacher models and regard them as annotations (just like KITTI with sparse annotations). In stage 2, we initialize our student model with one of the pre-trained teacher models, and then train the student model using the knowledge distillation variant 2 (from the confidence view) for another 600k iterations. Thanks to the simplicity of our loss functions, there is no need to tune hyper-parameters.

For KITTI datasets, the unsupervised training is only performed on the raw datasets and the pre-trained student model serves as initialization for supervised fine-tuning. For Sintel datasets, we conduct unsupervised training on both the raw dataset and the official training set (as shown in Table 8), where the former serves as initialization for supervised fine-tuning and the latter is used for a fair comparison with previous unsupervised methods.

We extend our knowledge distillation idea from unsupervised learning to semi-supervised learning, therefore our supervised fine-tuning also contains two stages. In the first stage, we train our student model on the official training image pairs for 1,000k iterations, where we disrupt the learning rate in a similar way as [66]. Then we generate flow predictions and confidence maps as self-annotations for raw data. In stage 2, we train our model with the combination of official training pairs with ground truth and raw data with self-annotations for 600k iterations.

*Evaluation Metrics.* We consider two widely-used metrics to evaluate optical flow estimation: average endpoint error

TABLE 1  
Quantitative Evaluation of Optical Flow Estimation on KITTI 2012 and KITTI 2015 Datasets

Method	KITTI 2012						KITTI 2015					
	train		test				train		test			
	EPE-all	EPE-noc	EPE-all	EPE-noc	Fl-all	Fl-noc	EPE-all	EPE-noc	Fl-all	Fl-fg	Fl-bg	
Unsupervised	BackToBasic [19]	11.3	4.3	9.9	4.6	43.15%	34.85%	–	–	–	–	–
	DSTFlow [20]	10.43	3.29	12.4	4.0	–	–	16.79	6.96	39%	–	–
	UnFlow-CSS [21]	3.29	1.26	–	–	–	–	8.10	–	23.30%	–	–
	OccAwareFlow [22]	3.55	–	4.2	–	–	–	8.88	–	31.2%	–	–
	Back2FutureFlow-None [23]*	–	–	–	–	–	–	6.65	3.24	–	–	–
	Back2FutureFlow-Soft [23]*	–	–	–	–	–	–	6.59	3.22	22.94%	24.27%	22.67%
	EpipolarFlow [38]	(2.51)	(0.99)	3.4	1.3	–	–	(5.55)	(2.46)	16.95%	–	–
	Lai <i>et al.</i> [44] (+Stereo)	2.56	1.39	–	–	–	–	7.13	4.31	–	–	–
	UnOS [45] (+Stereo)	1.64	1.04	1.8	–	–	–	5.58	–	18.00%	–	–
	DDFlow [24]	2.35	1.02	3.0	1.1	8.86%	4.57%	5.72	2.73	14.29%	20.40%	13.08%
	SelfFlow [25]*	1.69	0.91	2.2	1.0	7.68%	4.31%	4.84	2.40	14.19%	21.74%	12.68%
	Flow2Stereo [16] (+Stereo)	1.45	<b>0.82</b>	1.7	<b>0.9</b>	7.63%	4.02%	3.54	2.12	11.10%	<b>16.67%</b>	9.99%
	DistillFlow (trained on Sintel)	2.33	1.08	–	–	–	–	8.16	4.20	–	–	–
DistillFlow	<b>1.38</b>	0.83	<b>1.6</b>	<b>0.9</b>	<b>7.18%</b>	<b>3.91%</b>	<b>2.93</b>	<b>1.96</b>	<b>10.54%</b>	16.98%	<b>9.26%</b>	
Supervised	FlowNetS [14]	7.52	–	9.1	–	44.49%	–	–	–	–	–	–
	SpyNet [36]	3.36	–	4.1	2.0	20.97%	12.31%	–	–	35.07%	43.62%	33.36%
	FlowFieldsCNN [35]	–	–	3.0	1.2	13.01%	4.89%	–	–	18.68%	20.42%	18.33%
	DCFlow [10]	–	–	–	–	–	–	–	–	14.86%	23.70%	13.10%
	FlowNet2 [11]	(1.28)	–	1.8	1.0	8.80%	4.82%	(2.3)	–	10.41%	8.75%	10.75%
	UnFlow-CSS [21]	(1.14)	(0.66)	1.7	0.9	8.42%	4.28%	(1.86)	–	11.11%	15.93%	10.15%
	LiteFlowNet [13]	(1.05)	–	1.6	0.8	7.27%	3.27%	(1.62)	–	9.38%	7.99%	9.66%
	LiteFlowNet2 [65]	(0.95)	–	1.4	0.7	6.16%	2.63%	(1.33)	–	7.62%	7.64%	7.62%
	PWC-Net [12]	(1.45)	–	1.7	0.9	8.10%	4.22%	(2.16)	–	9.60%	9.31%	9.66%
	PWC-Net+ [66]	(1.08)	–	1.4	0.8	6.72%	3.36%	(1.45)	–	7.72%	7.88%	7.69%
	ContinualFlow [67]	–	–	–	–	–	–	–	–	10.03%	17.48%	8.54%
	HD <sup>3</sup> Flow [68]	(0.81)	–	1.4	0.7	5.41%	2.26%	(1.31)	–	6.55%	9.02%	6.05%
	IRR-PWC [16]	–	–	1.6	0.9	6.70%	3.21%	(1.45)	–	7.65%	7.52%	7.68%
	MFF [69]*	–	–	1.7	0.9	7.87%	4.19%	–	–	7.17%	<b>7.25%</b>	7.15%
	VCN [37]	–	–	–	–	–	–	(1.16)	–	6.30%	8.66%	5.83%
	SENSE [70]	(1.18)	–	1.5	–	–	3.03%	(2.05)	–	8.16%	–	–
	ScopeFlow [17]	–	–	1.3	0.7	5.66%	2.68%	–	–	6.82%	7.36%	6.72%
	MaskFlowNet-S [18]	–	–	<b>1.1</b>	<b>0.6</b>	5.24%	2.29%	–	–	6.81%	8.21%	6.53%
	MaskFlowNet [18]	–	–	<b>1.1</b>	<b>0.6</b>	<b>4.82%</b>	<b>2.07%</b>	–	–	6.11%	7.70%	5.79%
	SelfFlow [25]*	(0.76)	(0.47)	1.5	0.9	6.19%	3.32%	(1.18)	(0.82)	8.42%	7.61%	12.48%
DistillFlow	(0.79)	(0.45)	1.2	0.6	5.23%	2.33%	(1.14)	(0.74)	<b>5.94%</b>	7.96%	<b>5.53%</b>	

Missing entries (–) indicate that the results are not reported for the respective method. Bold fonts highlight the best results among unsupervised and supervised methods. Parentheses mean that training and testing are performed on the same dataset. *fg* and *bg* denote results of foreground and background regions respectively. (+Stereo) denotes stereo data is used during training, and \* denotes using more than two frames to estimate flow.

(EPE, lower is better), percentage of erroneous pixels (Fl, lower is better). Fl is the ranking metric on KITTI benchmarks, and EPE is the ranking metric on Sintel benchmarks. We also report the results over non-occluded pixels (noc) and occluded pixels (occ) respectively. Stereo matching is the byproduct of our flow models, we use EPE and D1 (share the same definition as Fl) as evaluation metrics. The harmonic average of the precision and recall (F-measure) is used to evaluate the accuracy of occlusion estimation.

## 4.2 Main Results

To alleviate the variance of flow prediction from one single model, we average the results of 10 models (2 independent runs with the last 5 checkpoints for each run). We first run each model to obtain results of different metrics, then average the metric results, which makes our results more reliable especially for ablation studies to analyze the effect of different components. For submission to the benchmark, we average the flow predictions from 2 independent runs.

*KITTI*. As shown in Table 1, DistillFlow achieves the best unsupervised results on both KITTI 2012 and KITTI 2015 datasets and outperforms them by a large margin. Specifically, on the KITTI 2012 training set, DistillFlow achieves EPE-all = 1.38 pixels, outperforming previous best unsupervised monocular method EpipolarFlow [38] by 45 percent.

Note that EpipolarFlow [38] fine-tunes its model on the KITTI 2012 training set, while DistillFlow is only trained on the raw dataset. DistillFlow outperforms UnOS [45] by 16 percent, which utilizes stereo videos and additional constraints during training. On KITTI 2012 benchmark, DistillFlow achieves EPE-all = 1.6 pixels and Fl-all = 7.18 percent, not only outperforms all previous unsupervised methods, but also outperforms some famous fully supervised methods, including FlowNet2 [11], LiteFlowNet [13], PWC-Net [12] and MFF [69]. On the KITTI 2015 dataset, the improvement is also significant. DistillFlow achieves EPE-all = 2.93 pixels on the training set, outperforming Back2FutureFlow [23] (utilizes multiple frames during the training) by 56 percent, EpipolarFlow [38] and UnOS [45] by 42 percent. On the benchmark, DistillFlow achieves Fl-all = 10.54 percent, which is a relative 38 percent improvement compared with previous best EpipolarFlow [38] (Fl-all = 16.95 percent).

After fine-tuning, DistillFlow also achieves state-of-the-art supervised learning performance on KITTI datasets. Specifically, on KITTI 2012, DistillFlow outperforms PWC-Net + [66], LiteFlowNet2 [65], HD<sup>3</sup>Flow [68], IRR-PWC [16] and ScopeFlow [17], is only inferior to MaskFlowNet [18], which incorporates an asymmetric feature matching module and direct occlusion reasoning. On KITTI 2015, DistillFlow achieves Fl-all = 5.94 percent, ranking 4th on the benchmark

TABLE 2  
Quantitative Evaluation of Optical Flow Estimation on Sintel Dataset

Method	Sintel Clean		Sintel Final		
	EPE-train	EPE-test	EPE-train	EPE-test	
Unsupervised	DSTFlow [20]	(6.16)	10.41	(6.81)	11.27
	UnFlow-CSS [21]	–	–	(7.91)	10.22
	OccAwareFlow [22]	(4.03)	7.95	(5.95)	9.15
	Back2FutureFlow-None [23]*	(6.05)	–	(7.09)	–
	Back2FutureFlow-Soft [23]*	(3.89)	7.23	(5.52)	8.81
	EpipolarFlow [38]	(3.54)	7.00	(4.99)	8.51
	DDFlow [24]	(2.92)	6.18	(3.98)	7.40
	SelFlow [25]*	(2.88)	6.56	(3.87)	6.57
	DistillFlow (trained on KITTI)	4.21	–	5.06	–
	DistillFlow	<b>(2.61)</b>	<b>4.23</b>	<b>(3.70)</b>	<b>5.81</b>
Supervised	FlowNet5 [14]	(3.66)	6.96	(4.44)	7.76
	FlowNetC [14]	(3.78)	6.85	(5.28)	8.51
	SpyNet [36]	(3.17)	6.64	(4.32)	8.36
	FlowFieldsCNN [35]	–	3.78	–	5.36
	DCFlow [10]	–	3.54	–	5.12
	FlowNet2 [11]	(1.45)	4.16	(2.01)	5.74
	LiteFlowNet [13]	<b>(1.35)</b>	4.54	(1.78)	5.38
	LiteFlowNet2 [65]	(1.41)	3.48	(1.83)	4.69
	PWC-Net [12]	(2.02)	4.39	(2.08)	5.04
	PWC-Net+ [66]	(1.71)	3.45	(2.34)	4.60
	ContinualFlow [67]	–	3.34	–	4.52
	HD <sup>3</sup> Flow [68]	(1.70)	4.79	<b>(1.17)</b>	4.67
	IRR-PWC [16]	(1.92)	3.84	(2.51)	4.58
	MFF [69]*	–	3.42	–	4.57
	VCN [37]	(1.66)	2.81	(2.24)	4.40
	SENSE [70]	(1.54)	3.60	(2.05)	4.86
	ScopeFlow [17]	–	3.59	–	<b>4.10</b>
	MaskFlowNet-S [18]	–	2.77	–	4.38
MaskFlowNet [18]	–	<b>2.52</b>	–	4.17	
SelFlow [25]*	(1.68)	3.74	(1.77)	4.26	
DistillFlow	(1.63)	3.49	(1.76)	<b>4.10</b>	

(the top three methods all use stereo data), outperforming all monocular optical flow methods (including the most recent MaskFlowNet [18] and ScopeFlow [17]). This is a remarkable result, since we do not require pre-training our model on any labeled synthetic dataset, while all other state-of-the-art supervised learning methods rely on pre-training on synthetic datasets and follow the specific training schedule (FlyingChairs [14] → FlyingThings3D [15]).

DistillFlow consistently outperforms our previous work DDFlow [24], SelFlow [25] and Flow2Stereo [26] (uses stereo data). Specifically, for unsupervised setting, DistillFlow outperforms SelFlow [25] 23 percent on KITTI 2012 benchmark and 26 percent on KITTI 2015 benchmark; for supervised setting, DistillFlow outperforms SelFlow [25] 20 percent on KITTI 2012 and 29 percent on KITTI 2015. The improvements mostly come from edge-aware smoothness regularizer and model distillation.

*MPI Sintel.* Table 2 summarizes the comparison of DistillFlow with existing unsupervised and supervised learning methods on Sintel. DistillFlow outperforms all previous unsupervised methods for all metrics. On the Sintel Clean benchmark, DistillFlow achieves EPE-all = 4.23 pixels, while previous best method EpipolarFlow [38] achieves EPE-all = 7.00 pixels, around 40 percent relative improvement. On the Sintel Final benchmark, DistillFlow achieves 21 percent relative improvement. Our initial data distillation method DDFlow [24] even outperforms all other unsupervised methods (including works that come out later than it, e.g., EpipolarFlow), demonstrating the effectiveness of our proposed knowledge distillation framework. DistillFlow significantly reduces the gap between state-of-the-art supervised learning methods and unsupervised methods.

After supervised fine-tuning, DistillFlow achieves EPE-all = 4.095 pixels on Sintel Final, and outperforms all published method on the benchmark, including our previous winner entry SelFlow [25] and most recent publications e.g., ScopeFlow [17] and MaskFlowNet [18].

Similarly to KITTI, DistillFlow also achieves improvements over our previous method DDFlow [24] and SelFlow [25], with 32 percent relative improvement on Sintel Clean and 12 percent relative improvement on Sintel Final.

*Qualitative Results.* Fig. 4 shows sample unsupervised results from KITTI and Sintel datasets. DistillFlow can estimate accurate flow and occlusion maps in a totally unsupervised manner. Figs. 5 and 6 show the qualitative comparison with state-of-the-art supervised learning methods on KITTI 2015 and Sintel Final benchmarks respectively. DistillFlow achieves better flow prediction, especially for occluded pixels.

*Occlusion Estimation.* Following previous works [22], [23], we evaluate our occlusion estimation performance on both KITTI and Sintel datasets. Note KITTI datasets only have sparse occlusion maps. As shown in Table 3, DistillFlow achieves best occlusion estimation performance on Sintel Clean and Sintel Final datasets over all competing methods. On the KITTI datasets, the ground truth occlusion masks only contain pixels moving out of the image boundary. However, our method will also estimate the occlusions within the image range. Under such settings, our method can achieve comparable performance. DistillFlow consistently outperforms our previous method [24], [25], suggesting better occlusion reasoning

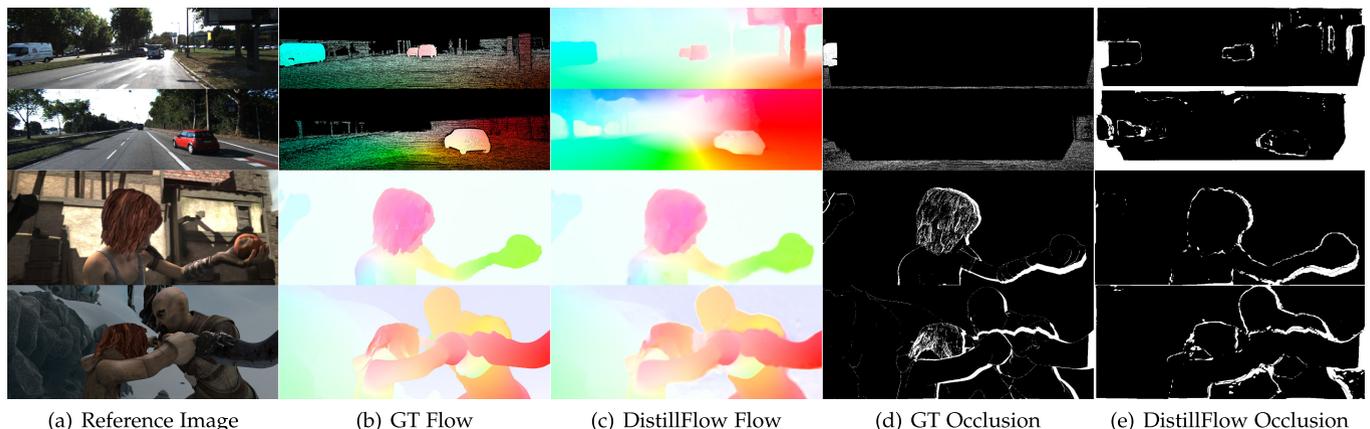


Fig. 4. Sample unsupervised results of DistillFlow on KITTI (top 2) and Sintel (bottom 2) training datasets. DistillFlow estimates both accurate optical flow and occlusion maps. Note that on KITTI datasets, the occlusion maps are sparse, which only contain pixels moving out of the image boundary.

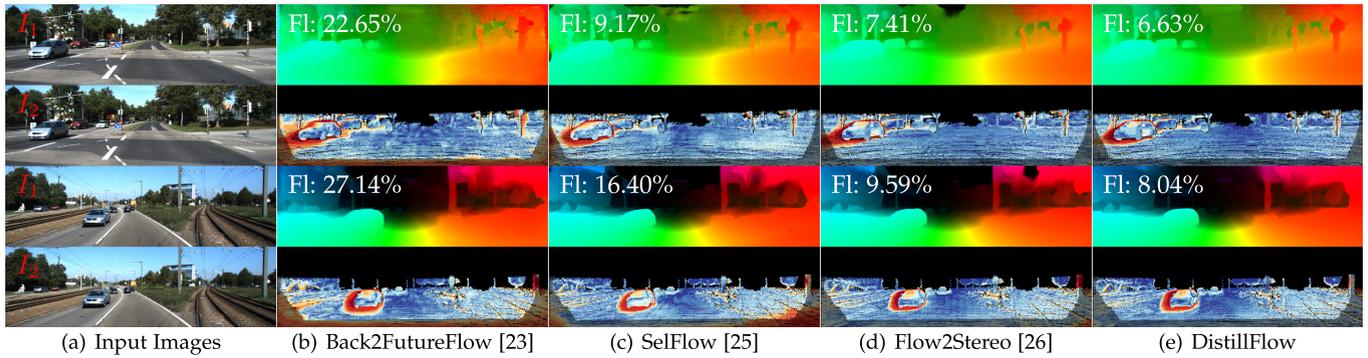


Fig. 5. Qualitative comparison with state-of-the-art unsupervised learning methods on the KITTI 2015 benchmark. For each case, the top row is optical flow and the bottom row is error map. The redder the color in the error map, the greater the error. More examples are available on KITTI 2015 benchmark.

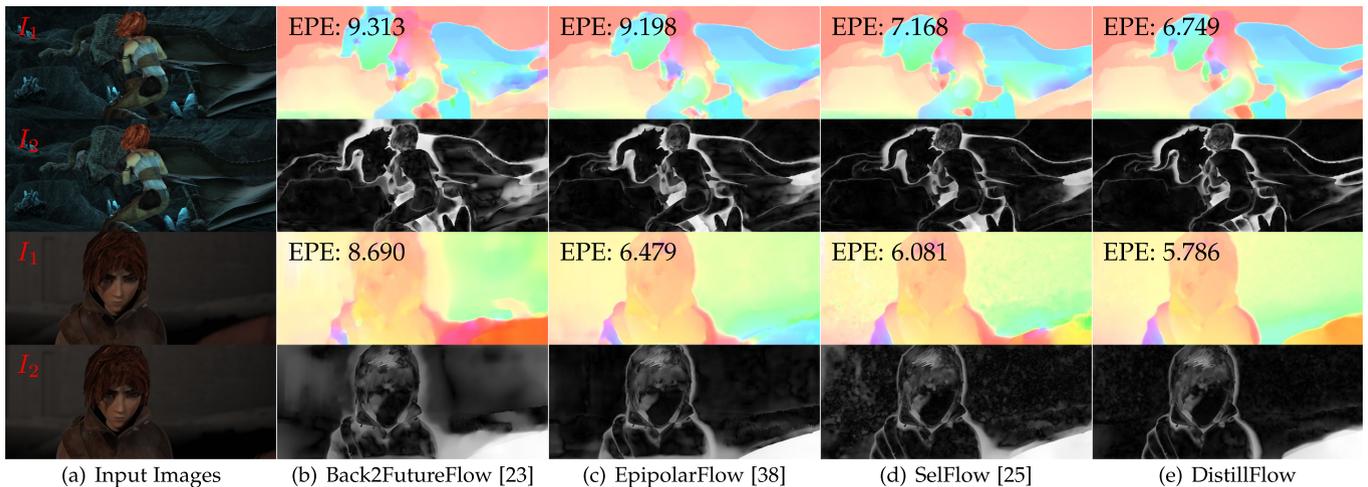


Fig. 6. Qualitative comparison with state-of-the-art unsupervised learning methods on Sintel Final benchmark. For each case, the top row is optical flow and the bottom row is error map. The whiter the color in the error map, the greater the error. More examples are available on the Sintel benchmark.

ability. This also explains why DistillFlow achieves performance improvement.

### 4.3 Generalization

We demonstrate the generalization capability of DistillFlow in three aspects: framework generalization, correspondence generalization and cross-dataset generalization.

*Framework Generalization.* Our proposed knowledge distillation based self-supervised learning framework is effective for different network structures and is applicable to both

unsupervised setting and supervised setting. To verify the former one, apart from PWC-Net based network backbones (as shown in Tables 7 and 8), we also apply our self-supervised learning framework to FlowNetS and FlowNetC (Table 4). With knowledge distillation, we achieve more than 30 percent relative improvement on average on KITTI datasets for both FlowNetS and FlowNetC, and 15 percent relative improvement on Sintel Clean and Final datasets. More importantly, FlowNetS and FlowNetC trained with knowledge distillation achieve EPE-all = 6.33 pixels and 5.47 pixels on KITTI 2015 training dataset, EPE-all = 4.83 pixels and 4.17 pixels on Sintel Final training dataset, which even outperform Back2FutureFlow [23] based on PWC-Net. This also has the same conclusion as [66]: model matters, so does training. Our knowledge distillation approach enables more effective training. All the above results demonstrate the generalization of our distillation framework to different network structures.

We also extend our knowledge distillation idea from unsupervised learning to supervised fine-tuning, resulting in a semi-supervised learning setting. The semi-supervised setting enables us to utilize more data. As shown in Table 5, we achieve improvement on both KITTI and Sintel datasets with knowledge distillation.

*Correspondence Generalization.* Stereo disparity, which describes the pixel displacement between two stereo images, can

TABLE 3  
Comparison of Occlusion Estimation with F-Measure

Method	KITTI 2012	KITTI 2015	Sintel Clean	Sintel Final
MODOF [75]	–	–	–	0.48
OccAwareFlow [22]	0.95	0.88	(0.54)	(0.48)
Back2Future [23]*	–	<b>0.91</b>	(0.49)	(0.44)
DDFlow [24]	0.94	0.86	<b>(0.59)</b>	(0.52)
SelFlow [25]*	0.95	0.88	<b>(0.59)</b>	(0.52)
DistillFlow	<b>0.96</b>	0.89	<b>(0.59)</b>	<b>(0.53)</b>

Note that on KITTI datasets, occlusion only contains pixels moving out of the image boundary and occlusion maps are sparse.

TABLE 4  
Ablation Study for the Generalization Capability of Our Proposed Distillation Framework to FlowNetS and FlowNetC on KITTI and Sintel Datasets

Network	Knowledge	KITTI 2012			KITTI 2015			Sintel Clean			Sintel Final		
		EPE-all	EPE-noc	EPE-occ	EPE-all	EPE-noc	EPE-occ	EPE-all	EPE-noc	EPE-occ	EPE-all	EPE-noc	EPE-occ
FlowNetS	✗	4.26	1.53	22.34	8.85	3.82	40.63	(5.05)	(3.09)	(30.01)	(5.38)	(3.38)	(31.00)
	✓	<b>2.70</b>	<b>1.38</b>	<b>11.44</b>	<b>6.33</b>	<b>3.44</b>	<b>24.59</b>	<b>(4.20)</b>	<b>(2.36)</b>	<b>(27.66)</b>	<b>(4.83)</b>	<b>(2.90)</b>	<b>(29.49)</b>
FlowNetC	✗	3.63	1.26	19.31	8.11	3.45	37.61	(4.20)	(2.36)	(27.66)	(4.83)	(2.90)	(29.49)
	✓	<b>2.18</b>	<b>1.16</b>	<b>8.97</b>	<b>5.47</b>	<b>2.95</b>	<b>21.38</b>	<b>(3.45)</b>	<b>(1.90)</b>	<b>(23.27)</b>	<b>(4.17)</b>	<b>(2.52)</b>	<b>(25.36)</b>

Default experimental settings: census transform (yes), occlusion handling (yes), edge-aware smoothness loss (yes).

TABLE 5  
Ablation Study for the Generalization Capability of Our Proposed Distillation Framework to Semi-Supervised Learning on KITTI and Sintel Datasets

Semi-Supervised Learning	KITTI 2012				KITTI 2015				Sintel Clean			Sintel Final		
	EPE-all	EPE-noc	Fl-all	Fl-noc	EPE-all	EPE-noc	Fl-all	Fl-noc	EPE-all	EPE-noc	EPE-occ	EPE-all	EPE-noc	EPE-occ
✗	1.01	0.58	3.46%	1.69%	1.50	<b>0.94</b>	5.17%	3.54%	1.64	0.77	15.65	2.44	1.51	17.59
✓	<b>0.95</b>	<b>0.57</b>	<b>3.35%</b>	<b>1.65%</b>	<b>1.44</b>	<b>0.94</b>	<b>4.96%</b>	<b>3.51%</b>	<b>1.56</b>	<b>0.72</b>	<b>15.21</b>	<b>2.38</b>	<b>1.47</b>	<b>17.16</b>

In this experiment, we split KITTI and Sintel dataset into **training** and **validation** datasets and evaluate the performance the **validation** part.

be regarded as a special case of optical flow on the epipolar line. They can both be regarded as a correspondence matching problem. From this point of view, if a model can accurately estimate optical flow, it shall have the ability to accurately estimate stereo disparity as well. Then as a byproduct, we directly use our flow model trained on monocular videos to estimate disparity. Surprisingly, our flow model achieves comparable stereo matching performance with current state-of-the-art unsupervised stereo matching methods. As shown in Table 6, DistillFlow achieves D1-all = 4.81 percent on KITTI 2012 training dataset and D1-all = 6.37 percent on KITTI 2015 dataset, outperforming some famous stereo matching methods e.g., SeqStereo *et al.* [50] and Guo *et al.* [49]. On the KITTI 2012 and 2015 benchmarks, DistillFlow achieves D1-all 5.14 and 6.81 percent, which are comparable with previous state-of-the-art methods UnOS [45] and our previous method Flow2Stereo [26]. The results on stereo matching demonstrate the generalization of DistillFlow to find correspondences.

*Cross-Dataset Generalization.* Although deep learning based optical flow methods have outperformed classical methods on challenging benchmarks, their generalization ability is very poor due to limited annotated training data. Therefore, currently learning based methods still cannot apply to many scenes. However, our proposed DistillFlow is a self-supervised learning approach, which can utilize unlimited in-the-wild videos and effectively learn optical flow without requiring any annotations. Since a large collection of unlabeled image sequences can be used, the learned model shall have the strong generalization capability. As shown in Table 1 (DistillFlow (trained on Sintel)) and Table 2 (DistillFlow (trained on KITTI)), we use models trained on Sintel to estimate flow on KITTI and vice versa. Surprisingly, for Sintel → KITTI, DistillFlow achieves EPE = 2.33 pixels on KITTI 2012 training dataset, outperforming previous state-of-the-art unsupervised learning method Back2FutureFlow [23]. On KITTI 2015, DistillFlow outperforms OccAwareFlow [22] is also comparable

TABLE 6  
Quantitative Evaluation of Stereo Disparity on KITTI 2012 and KITTI 2015 Training Datasets (Apart From the Last Columns)

Method	KITTI 2012						KITTI 2015					
	EPE-all	EPE-noc	EPE-occ	D1-all	D1-noc	D1-all (test)	EPE-all	EPE-noc	EPE-occ	D1-all	D1-noc	D1-all (test)
Joung <i>et al.</i> [76]	–	–	–	–	–	13.88%	–	–	–	13.92%	–	–
Godard <i>et al.</i> [47] *	2.12	1.44	30.91	10.41%	8.33%	–	1.96	1.53	24.66	10.86%	9.22%	–
Zhou <i>et al.</i> [48]	–	–	–	–	–	–	–	–	–	9.41%	8.35%	–
OASM-Net [51]	–	–	–	8.79%	6.69%	8.60%	–	–	–	–	–	8.98%
SeqStereo <i>et al.</i> [50] *	2.37	1.63	33.62	9.64%	7.89%	–	1.84	1.46	26.07	8.79%	7.7%	–
Liu <i>et al.</i> [43] *	1.78	1.68	6.25	11.57%	10.61%	–	1.52	1.48	4.23	9.57%	9.10%	–
Guo <i>et al.</i> [49] *	1.16	1.09	4.14	6.45%	5.82%	–	1.71	1.67	4.06	7.06%	6.75%	–
UnOS [45]	–	–	–	–	–	5.93%	–	–	–	<b>5.94%</b>	–	6.67%
Flow2Stereo [26]	<b>1.01</b>	<b>0.93</b>	4.52	5.14%	4.59%	5.11%	1.34	1.31	<b>2.56</b>	6.13%	<b>5.93%</b>	<b>6.61%</b>
DistillFlow (no distillation)	1.25	1.03	10.57	6.67%	4.94%	–	1.44	1.30	9.13	8.19%	6.90%	–
DistillFlow	1.02	0.95	<b>3.72</b>	<b>4.81%</b>	<b>4.40%</b>	5.14%	<b>1.23</b>	<b>1.21</b>	2.78	6.37%	6.17%	6.81%

Our flow model trained on monocular videos achieves comparable performance with state-of-the-art unsupervised stereo learning methods. \* denotes that we use their pre-trained model to compute the numbers, while other numbers are from their paper.

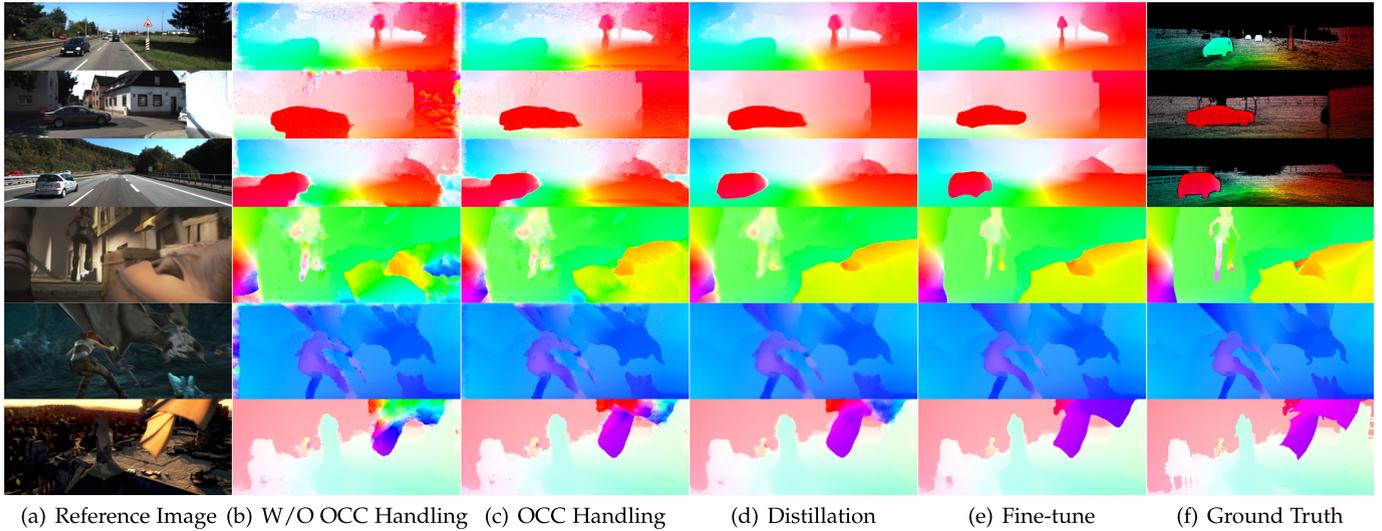


Fig. 7. Ablation study on KITT1 2015 (top 3) and Sintel Final training datasets (bottom 3). (b) and (c) are the results of without and with occlusion handling. (d) shows that results with knowledge distillation and (f) is the supervised fine-tuned results. With knowledge distillation, the flow looks more smooth. After fine-tuning, more details are preserved. Zoom in for details.

with Back2FutureFlow [23]. For KITT1  $\rightarrow$  Sintel, DistillFlow achieves EPE = 5.06 pixels on Sintel Final training dataset, which outperforms Back2FutureFlow [23] and is comparable with EpipolarFlow [38]. This is indeed a remarkable result, since KITT1 datasets only have street views while the Sintel dataset contains many complex scenes. Our model trained only on KITT1 datasets achieves even better or comparable performance with state-of-the-art unsupervised learning methods fine-tuned on Sintel dataset. This fully demonstrates the cross-dataset generalization capability of our model. Moreover, since our knowledge distillation method can work well without requiring any labeled data, we can actually train it on a specific scene to achieve better performance. This makes DistillFlow effective for a wider range of applications.

#### 4.4 Ablation Study

We conduct a thorough ablation study to demonstrate the effectiveness of different components proposed by DistillFlow. Fig. 7 shows visual comparisons.

**Occlusion Handling.** As shown in Table 7 (row 1 versus row 2) and Table 8 (row 1 versus row 2 and row 7 versus row 8), occlusion handling can improve the flow estimation performance on all datasets for all metrics. This is because that brightness constancy assumption does not hold for occluded pixels. Occlusion handling can reduce the misleading guidance information provided by occluded pixels, which makes the model easier to learn good correspondence.

**Edge-Aware Smoothness.** As shown in Table 7 (row 2 versus row 3) and Table 8 (row 2 versus row 3 and row 8 versus row 9), edge-aware smoothness regularizer can also consistently improve the performance on all datasets. This is because photometric loss is not informative in homogeneous regions and cannot handle occlusions. The spatial smooth assumption regularizes the flow to be locally similar, which helps predict flow of those homogeneous or texture-less regions. Smoothness can be regarded as a regularizer for some occluded pixels, since it makes the prediction of occluded pixels similar to the neighborhood of non-occluded pixels. However, it is just a very

weak regularizer, therefore we propose knowledge distillation to more effectively learn optical flow of occluded pixels.

**Data Distillation.** Our proposed knowledge distillation approach contains both data distillation and model distillation. Among them, data distillation is the key point, where we create challenging transformations and let confident predictions to supervise less confident predictions. As shown in Table 7 (row 3 versus row 5), we reduce EPE-all from 2.92 pixels to 1.46 pixels, from 6.45 pixels to 3.20 pixels on KITT1 2012 and KITT1 2015 datasets, with 50 percent relative improvement on average. The improvement over occluded pixels is even more significant, with 62 percent relative improvement on average. This is because our proposed data distillation enables the model to have the ability to effectively learn optical flow of occluded pixels for the first time.

As shown in Table 8 (row 3 versus row 5 and row 9 versus row 11), we also achieve great improvement on Sintel datasets. Specifically, we achieve 9 percent average relative improvement on both Sintel Clean and Sintel Final. All these results demonstrate the effectiveness of our proposed data distillation approach.

**Model Distillation.** Since flow predictions from one single teacher model have large variance, we propose model distillation to ensemble flow predictions of multiple teacher models. Model distillation provides more reliable confident flow predictions, which therefore improves the performance. As shown in Table 7 (row 3 versus row 4) and Table 8 (row 3 versus row 4 and row 9 versus row 10), the average of multiple teacher models has higher accuracy than one teacher model. Consequently, data distillation equipped with model distillation further improves the performance as show in Table 7 (row 5 versus row 6) and Table 8 (row 5 versus row 6 and row 11 versus row 12).

**Knowledge Distillation Strategies.** In Section 3.3.2, we introduce two variants for knowledge distillation: from occlusion view and from confidence view, resulted in two knowledge distillation strategies ‘v3’ and ‘v4’ in Table 9. We also provide two knowledge distillation strategies as in our previous work DDFlow [24] and SelFlow [25], denoted as ‘v1’ and

TABLE 7  
Main Ablation Study on KITTI Training Datasets

Occlusion Handling	Edge-Aware Smoothness	Data Distillation	Model Distillation	KITTI 2012					KITTI 2015				
				EPE-all	EPE-noc	EPE-occ	Fl-all	Fl-noc	EPE-all	EPE-noc	EPE-occ	Fl-all	Fl-noc
×	×	×	×	7.33	1.30	47.26	16.27%	5.97%	12.49	3.59	68.82	23.07%	12.40%
✓	×	×	×	3.22	0.98	18.07	13.57%	4.40%	6.57	2.88	29.87	19.90%	10.01%
✓	✓	×	×	2.92	0.93	16.06	12.44%	3.94%	6.45	2.59	30.90	19.08%	9.48%
✓	✓	×	✓	2.86	0.91	15.84	11.58%	3.85%	6.36	2.52	29.76	18.24%	9.32%
✓	✓	✓	×	1.46	0.85	5.44	5.17%	3.38%	3.20	2.08	10.28	10.05%	8.03%
✓	✓	✓	✓	<b>1.38</b>	<b>0.83</b>	<b>4.98</b>	<b>4.99%</b>	<b>3.25%</b>	<b>2.93</b>	<b>1.96</b>	<b>9.04</b>	<b>9.79%</b>	<b>7.81%</b>

In this experiment, we employ census transform when computing photometric loss. Note that when data distillation is not employed, model distillation (row 4) denotes ensembling predictions of multiple teacher models.

TABLE 8  
Main Ablation Study on Sintel Training Datasets

Training Dataset	Occlusion Handling	Edge-Aware Smoothness	Data Distillation	Model Distillation	Sintel Clean					Sintel Final				
					EPE-all	EPE-noc	EPE-occ	Fl-all	Fl-noc	EPE-all	EPE-noc	EPE-occ	Fl-all	Fl-noc
Sintel Raw	×	×	×	×	4.17	1.85	33.95	10.21%	5.11%	5.36	2.86	37.30	14.46%	9.42%
	✓	×	×	×	3.58	1.65	28.67	8.96%	4.17%	4.67	2.61	31.08	13.39%	8.49%
	✓	✓	×	×	3.29	1.54	25.70	8.25%	3.76%	4.41	2.50	28.75	12.89%	8.18%
	✓	✓	✓	×	3.25	1.52	25.36	8.12%	3.63%	4.36	2.46	28.22	12.45%	7.92%
	✓	✓	✓	✓	3.04	1.42	23.72	7.56%	4.58%	3.98	2.26	25.81	11.14%	6.92%
	✓	✓	✓	✓	<b>2.98</b>	<b>1.39</b>	<b>23.43</b>	<b>7.45%</b>	<b>3.49%</b>	<b>3.90</b>	<b>2.21</b>	<b>25.52</b>	<b>10.98%</b>	<b>6.76%</b>
Sintel Train	×	×	×	×	(3.93)	(1.56)	(34.23)	(9.61%)	(4.45%)	(5.19)	(2.66)	(37.54)	(13.64%)	(8.54%)
	✓	×	×	×	(3.22)	(1.34)	(27.24)	(8.43%)	(3.51%)	(4.40)	(2.36)	(30.49)	(12.72%)	(7.44%)
	✓	✓	×	×	(2.93)	(1.24)	(24.66)	(7.63%)	(3.15%)	(4.17)	(2.32)	(27.83)	(12.31%)	(7.62%)
	✓	✓	✓	×	(2.89)	(1.22)	(24.21)	(7.46%)	(3.08%)	(4.12)	(2.29)	(27.24)	(12.12%)	(7.51%)
	✓	✓	✓	✓	(2.66)	(1.16)	(21.89)	(7.03%)	(3.14%)	(3.76)	(2.10)	(24.92)	(10.70%)	(6.55%)
	✓	✓	✓	✓	<b>(2.61)</b>	<b>(1.12)</b>	<b>(21.63)</b>	<b>(6.87%)</b>	<b>(2.99%)</b>	<b>(3.70)</b>	<b>(2.07)</b>	<b>(24.60)</b>	<b>(10.61%)</b>	<b>(6.45%)</b>

In this experiment, we employ census transform when computing photometric loss. Note that when data distillation is not employed, model distillation (row 4 and row 10) denotes ensembling predictions of multiple teacher models.

‘v2’. Both ‘v1’ and ‘v2’ are from the occlusion view. As shown in Table 9, all of these four kinds of knowledge distillation strategies can greatly improve the performance, especially for occluded pixels. Comparing ‘v1’ and ‘v2’, we show that superpixel occlusion hallucination can handle occlusion in a wider range of cases. compared with ‘v2’, we add more challenging transformations for ‘v3’, such as geometric transformations and color transformations. As a result, we achieve slight performance improvements. However, most performance gains come from occlusion hallucination techniques. Comparing ‘v3’ and ‘v4’, we show that it does not make much difference to distinguish occluded or non-occluded pixels during the knowledge distillation stage. This is because forward-backward consistency only predicts confident or less confident flow predictions, but not occluded or non-occluded pixels.

*Photometric Losses.* When computing photometric loss, certain transformations are usually applied to the images to make them more robust for illumination changes. As a result, different papers employ different strategies, e.g., raw pixel intensity [22], [23], SSIM [41] and census transform [21]. To analyze the effect of different photometric loss, we make a comparison in Table 10, where SSIM is better than the raw pixel intensity and census transform achieves the best performance. This is because census transform is specially designed to handle the change of illumination. However, we believe census transform is not the optimal transformation for optical flow estimation. Exploring more robust transformations when computing the photometric difference is a potential direction for future research.

TABLE 9  
Ablation Study of Different Knowledge Distillation Strategies on KITTI and Sintel Datasets

Knowledge Distillation	KITTI 2012			KITTI 2015			Sintel Clean			Sintel Final		
	EPE-all	EPE-noc	EPE-occ	EPE-all	EPE-noc	EPE-occ	EPE-all	EPE-noc	EPE-occ	EPE-all	EPE-noc	EPE-occ
no	2.92	0.93	16.06	6.45	2.59	30.90	(2.93)	(1.24)	(24.66)	(4.17)	(2.32)	(27.83)
v1	1.62	0.87	6.21	3.88	2.19	15.24	(2.76)	(1.16)	(22.98)	(3.94)	(2.16)	(25.72)
v2	1.54	0.87	5.77	3.57	2.10	12.88	(2.71)	(1.18)	(22.51)	(3.87)	(2.19)	(25.38)
v3	1.41	0.85	5.12	3.12	2.01	9.48	(2.63)	<b>(1.12)</b>	(21.72)	(3.74)	(2.09)	(24.81)
v4	<b>1.38</b>	<b>0.83</b>	<b>4.98</b>	<b>2.93</b>	<b>1.96</b>	<b>9.04</b>	<b>(2.61)</b>	<b>(1.12)</b>	<b>(21.63)</b>	<b>(3.70)</b>	<b>(2.07)</b>	<b>(24.60)</b>

For ‘v1’, ‘v2’ and ‘v3’, we use knowledge distillation variant 1 (from occlusion view and split pixels in occluded and non-occluded), while for ‘v2’ we use variant 2 (from confidence view). ‘v1’ denotes distillation used in DDFlow [24], ‘v2’ denotes distillation used in SelFlow [25], ‘v3’ and ‘v4’ denotes distillation with more challenging transformations as in Flow2Stereo [26]. Default experimental settings: census transform (yes), occlusion handling (yes), edge-aware smoothness loss (yes).

TABLE 10  
Ablation Study of Photometric Losses

Pixel	SSIM	Census Transform	KITTI 2012			KITTI 2015			Sintel Clean			Sintel Final		
			EPE-all	EPE-noc	EPE-occ	EPE-all	EPE-noc	EPE-occ	EPE-all	EPE-noc	EPE-occ	EPE-all	EPE-noc	EPE-occ
Brightness														
✓	×	×	6.58	2.03	36.69	10.63	3.99	52.62	(4.44)	(2.12)	(34.16)	(6.86)	(4.42)	(38.05)
✓	✓	×	5.75	1.09	36.62	9.85	3.14	52.32	(4.15)	(1.98)	(32.00)	(6.22)	(3.71)	(38.29)
×	×	✓	<b>3.22</b>	<b>0.98</b>	<b>18.07</b>	<b>6.57</b>	<b>2.88</b>	<b>29.87</b>	<b>(3.22)</b>	<b>(1.34)</b>	<b>(27.24)</b>	<b>(4.40)</b>	<b>(2.36)</b>	<b>(30.49)</b>

Default experiment settings: edge-aware smoothness loss (no), occlusion handling (yes), distillation (no).

## 5 CONCLUSION

We have presented DistillFlow, a knowledge distillation approach to effectively learning optical flow in a self-supervised manner. To this end, we train multiple teacher models and a student model, where teacher models are used to generate confident flow predictions, which will then be employed to guide the learning of the student model. To make the knowledge distillation effective, we create three types of challenging transformations: occlusion hallucination-based transformations, geometric transformations and color transformations. We show that the key factors for performance gains are generating hallucinated occlusions as well as less confident predictions. With knowledge distillation, DistillFlow achieves the best performance on both KITTI and Sintel datasets and outperforms previous unsupervised methods by a large margin, especially for occluded pixels. More importantly, our self-supervised pre-trained model provides an excellent initialization for supervised fine-tuning. We show that it is possible to completely remove the reliance of pre-training on synthetic labeled datasets, and achieve superior performance by self-supervised pre-training on unlabeled data. Furthermore, we demonstrate the generalization capability of DistillFlow in three aspects: framework generalization, correspondence generalization and cross-dataset generalization. Going forward, our results suggest that our knowledge distillation technique may be a promising direction for advancing other vision tasks like depth estimation or semantic segmentation.

## ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China (No. 2018AA0100204) and Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 14210717 of the General Research Fund). Pengpeng Liu did mainly the work during an internship at Huya AI.

## REFERENCES

- [1] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3061–3070.
- [2] A. K. Chauhan and P. Krishan, "Moving object tracking using gaussian mixture model and optical flow," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 4, pp. 243–246, 2013.
- [3] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
- [4] N. Bonneel, J. Tompkin, K. Sunkavalli, D. Sun, S. Paris, and H. Pfister, "Blind video temporal consistency," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–9, 2015.
- [5] D. Chen, J. Liao, L. Yuan, N. Yu, and G. Hua, "Coherent online video style transfer," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 1105–1114.
- [6] M. S. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-recurrent video super-resolution," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6626–6634.
- [7] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1/3, pp. 185–203, 1981.
- [8] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 25–36.
- [9] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.
- [10] J. Xu, R. Ranftl, and V. Koltun, "Accurate optical flow via direct cost volume processing," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1289–1297.
- [11] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2462–2470.
- [12] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8934–8943.
- [13] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A lightweight convolutional neural network for optical flow estimation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8981–8989.
- [14] A. Dosovitskiy et al., "FlowNet: Learning optical flow with convolutional networks," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 2758–2766.
- [15] N. Mayer et al., "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4040–4048.
- [16] "Iterative residual refinement for joint optical flow and occlusion estimation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5754–5763.
- [17] A. Bar-Haim and L. Wolf, "ScopeFlow: Dynamic scene scoping for optical flow," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7998–8007.
- [18] S. Zhao, Y. Sheng, Y. Dong, E. I.-C. Chang, and Y. Xu, "MaskFlowNet: Asymmetric feature matching with learnable occlusion mask," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6278–6287.
- [19] J. Y. Jason, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 3–10.
- [20] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation," in *Proc. Assoc. Advancement Artif. Intell.*, 2017, pp. 1495–1501.
- [21] S. Meister, J. Hur, and S. Roth, "UnFlow: Unsupervised learning of optical flow with a bidirectional census loss," in *Proc. Assoc. Advancement Artif. Intell.*, 2018, pp. 7251–7259.
- [22] Y. Wang, Y. Yang, Z. Yang, L. Zhao, and W. Xu, "Occlusion aware unsupervised learning of optical flow," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4884–4893.
- [23] J. Janai, F. Güney, A. Ranjan, M. J. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 690–706.
- [24] P. Liu, I. King, M. R. Lyu, and J. Xu, "DDFlow: Learning optical flow with unlabeled data distillation," in *Proc. Assoc. Advancement Artif. Intell.*, 2019, pp. 8770–8777.
- [25] P. Liu, I. King, M. R. Lyu, and J. Xu, "SelfFlow: Self-supervised learning optical flow," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4571–4580.
- [26] P. Liu, I. King, M. R. Lyu, and J. Xu, "Flow2Stereo: Effective self-supervised learning of optical flow and stereo matching," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6648–6657.

- [27] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2432–2439.
- [28] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proc. Int. Conf. Comput. Vis.*, 2013, pp. 1385–1392.
- [29] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1164–1172.
- [30] J. Janai, F. Güney, J. Wulff, M. J. Black, and A. Geiger, "Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3597–3607.
- [31] R. Kennedy and C. J. Taylor, "Optical flow with geometric occlusion estimation and fusion of multiple frames," in *Proc. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 364–377.
- [32] D. Sun, E. B. Sudderth, and M. J. Black, "Layered image motion with explicit occlusions, temporal consistency, and depth ordering," in *Proc. Conf. Neural Inf. Process. Syst.*, 2010, pp. 2226–2234.
- [33] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer, "Modeling temporal coherence for optical flow," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1116–1123.
- [34] M. J. Black and P. Anandan, "Robust dynamic motion estimation over time," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 1991, pp. 296–303.
- [35] C. Bailer, K. Varanasi, and D. Stricker, "CNN-based patch matching for optical flow with thresholded hinge embedding loss," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3250–3259.
- [36] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4161–4170.
- [37] G. Yang and D. Ramanan, "Volumetric correspondence networks for optical flow," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 1–12.
- [38] Y. Zhong, P. Ji, J. Wang, Y. Dai, and H. Li, "Unsupervised deep epipolar flow for stationary or dynamic scenes," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12095–12104.
- [39] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1851–1858.
- [40] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 36–53.
- [41] Z. Yin and J. Shi, "GeoNet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1983–1992.
- [42] A. Ranjan *et al.*, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12240–12249.
- [43] L. Liu, G. Zhai, W. Ye, and Y. Liu, "Unsupervised learning of scene flow estimation fusing with local rigidity," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 876–882.
- [44] H.-Y. Lai, Y.-H. Tsai, and W.-C. Chiu, "Bridging stereo matching and optical flow via spatiotemporal correspondence," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1890–1899.
- [45] Y. Wang, P. Wang, Z. Yang, C. Luo, Y. Yang, and W. Xu, "UnOS: Unified unsupervised optical-flow and stereo-depth estimation by watching videos," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8071–8081.
- [46] R. Garg, B. V. Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 740–756.
- [47] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 270–279.
- [48] C. Zhou, H. Zhang, X. Shen, and J. Jia, "Unsupervised learning of stereo matching," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 1567–1575.
- [49] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang, "Learning monocular depth by distilling cross-domain stereo networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 484–500.
- [50] G. Yang, H. Zhao, J. Shi, Z. Deng, and J. Jia, "SegStereo: Exploiting semantic information for disparity estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 636–651.
- [51] A. Li and Z. Yuan, "Occlusion aware stereo matching via cooperative unsupervised learning," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 197–213.
- [52] A. Tonioni, M. Poggi, S. Mattoccia, and L. Di Stefano, "Unsupervised adaptation for deep stereo," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 1605–1613.
- [53] Y. Zhong, H. Li, and Y. Dai, "Open-world stereo video matching with deep RNN," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 101–116.
- [54] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. D. Stefano, "Real-time self-adaptive deep stereo," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 195–204.
- [55] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 4, 2020, doi: [10.1109/TPAMI.2020.2992393](https://doi.org/10.1109/TPAMI.2020.2992393).
- [56] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.
- [57] G. Larsson, M. Maire, and G. Shakhnarovich, "Colorization as a proxy task for visual understanding," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6874–6883.
- [58] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 69–84.
- [59] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan, "Learning features by watching objects move," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2701–2710.
- [60] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2051–2060.
- [61] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *Stat.*, vol. 1050 p. 9, 2015.
- [62] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He, "Data distillation: Towards omni-supervised learning," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4119–4128.
- [63] N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by GPU-accelerated large displacement optical flow," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 438–451.
- [64] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
- [65] T.-W. Hui, X. Tang, and C. C. Loy, "A lightweight optical flow CNN—revisiting data fidelity and regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 28, 2020, doi: [10.1109/TPAMI.2020.2976928](https://doi.org/10.1109/TPAMI.2020.2976928).
- [66] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Models matter, so does training: An empirical study of CNNs for optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1408–1423, Jun. 2020.
- [67] M. Neoral, J. Šochman, and J. Matas, "Continual occlusion and optical flow estimation," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 159–174.
- [68] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6044–6053.
- [69] Z. Ren, O. Gallo, D. Sun, M.-H. Yang, E. B. Sudderth, and J. Kautz, "A fusion approach for multi-frame optical flow estimation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 2077–2086.
- [70] H. Jiang, D. Sun, V. Jampani, Z. Lv, E. Learned-Miller, and J. Kautz, "SENSE: A shared encoder network for scene-flow estimation," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 3195–3204.
- [71] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
- [72] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 611–625.
- [73] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proc. Eur. Conf. Comput. Vis.*, 1994, pp. 151–158.
- [74] D. Hafner, O. Demetz, and J. Weickert, "Why is the census transform good for robust optic flow computation?," in *Proc. Int. Conf. Scale Space Variational Methods Comput. Vis.*, 2013, pp. 210–221.
- [75] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, Sep. 2011.
- [76] S. Joung, S. Kim, K. Park, and K. Sohn, "Unsupervised stereo matching using confidential correspondence consistency," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 2190–2203, May 2019.



**Pengpeng Liu** received the BS degree in electronic engineering from Tsinghua University, Beijing, China, in 2016. He is currently working toward the PhD degree at the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include computer vision, deep learning, and machine learning. His SelfFlow paper was a finalist in the CVPR 2019 Best Paper selection.



**Michael R. Lyu** (Fellow, IEEE) received the BS degree in electrical engineering from National Taiwan University, Taipei, Taiwan; the MS degree in computer science from the University of California, Santa Barbara, and the PhD degree in computer science from the University of California, Los Angeles. He is currently Choh-Ming Li professor of Computer Science and Engineering, Chinese University of Hong Kong. His research interests include software engineering, software reliability, machine learning, multimedia information retrieval,

and distributed systems. He has published more than 600 refereed journal and conference papers in his research areas. His Google Scholar citation is more than 40,000, with an h-index of 99. He initiated the first International Symposium on Software Reliability Engineering (ISSRE) in 1990. He was an associate editor of the *IEEE Transactions on Reliability*, the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Transactions on Services Computing*, and the *Journal of Information Science and Engineering*. He is currently on the editorial board of *IEEE Access*, *Wiley Software Testing, Verification and Reliability Journal (STVR)*, and the *ACM Transactions on Software Engineering Methodology (TOSEM)*. He was elected to AAAS fellow (2007), ACM fellow (2015), and named IEEE Reliability Society Engineer of the Year (2010). He was granted with China Computer Federation (CCF) Overseas Outstanding Contributions Award in 2018, and the 13th Guanghua Engineering Science and Technology Award in 2020. He was also named in The AI 2000 Most Influential Scholars Annual List with three appearances in 2020.



**Irwin King** (Fellow, IEEE) received the BS degree in engineering and applied science from the California Institute of Technology (Caltech), Pasadena, and the MS and PhD degrees in computer science from the University of Southern California (USC), Los Angeles. He is the chair and professor of Computer Science and Engineering, The Chinese University of Hong Kong. His research interests include machine learning, social computing, AI, web intelligence, data mining, and multimedia information processing. In these research areas, he has more than 350 technical publications in journals and conferences. He is an associate editor of the *Journal of Neural Networks (NN)*. He is an ACM distinguished member, and a fellow of the Hong Kong Institute of Engineers (HKIE). He has served as the President of the International Neural Network Society (INNS), General Co-chair of The WebConf 2020, ICONIP 2020, WSDM 2011, RecSys 2013, ACML 2015, and in various capacities in a number of top conferences and societies such as WWW, NIPS, ICML, IJCAI, AAAI, APNNS, etc. He is the recipient of the ACM CIKM2019 Test of Time Award, the ACM SIGIR 2020 Test of Time Award, and 2020 APNNS Outstanding Achievement Award for his contributions made in social computing with machine learning. In early 2010 while on leave with AT&T Labs Research, San Francisco, he taught classes as a visiting professor with UC Berkeley.



**Jia Xu** (Fellow, IEEE) received the BS degree in computer science and technology from Nanjing University, Nanjing, Jiangsu, China, and the PhD degree in computer science from the University of Wisconsin-Madison, Madison, WI. His research interests include computer vision and deep learning. He worked as a senior research scientist with the Intel Visual Computing Lab, Santa Clara, CA, and a principal researcher and manager with Tencent AI Lab, Shenzhen, Guangdong, China. He is currently the head of Huya AI and General Manager at Huya Inc, Guangzhou, Guangdong, China.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/csdl](http://www.computer.org/csdl).