Optical Flow as Spatial-Temporal Attention Learners

Yawen Lu[1†], Cheng Han[2†], Qifan Wang[3], Heng Fan[4], Zhaodan Kong[5], Dongfang Liu[2], and Yingjie Chen[1]

[1]Purdue University, USA
[2]Rochester Institute of Technology, USA
[3]Meta AI, USA
[4]University of North Texas, USA
[5]University of California, Davis, USA

Abstract—Optical flow is an indispensable building block for various important computer vision tasks, including motion estimation, object tracking, and disparity measurement. To date, the dominant methods are CNN-based, leaving plenty of room for improvement. In this work, we propose TransFlow, a transformer architecture for optical flow estimation. Compared to dominant CNN-based methods, TransFlow demonstrates three advantages. First, it provides more accurate correlation and trustworthy matching in flow estimation by utilizing spatial self-attention and cross-attention mechanisms between adjacent frames to effectively capture global dependencies; Second, it recovers more compromised information (*e.g.*, occlusion and motion blur) in flow estimation through long-range temporal association in dynamic scenes; Third, it introduces a concise self-learning paradigm, eliminating the need for complex and laborious multi-stage pre-training procedures. The versatility and superiority of TransFlow extend seamlessly to 3D scene motion, yielding competitive outcomes in 3D scene flow estimation. Our approach attains state-of-the-art results on benchmark datasets such as Sintel and KITTI-15, while also exhibiting exceptional performance on downstream tasks, including video object detection using the ImageNet VID dataset, video frame interpolation using the GoPro dataset, and video stabilization using the DeepStab dataset. We believe that the effectiveness of TransFlow positions it as a flexible baseline for both optical flow and scene flow estimation, offering promising avenues for future research and development.

Index Terms—Self-learning Paradigm, Optical Flow Estimation, Scene Flow Estimation.

1 Introduction

With resurgence of connectionism, significant advancements have been achieved in the field of optical flow. Up until now, the majority of cutting-edge flow learning approaches have been built upon Convolutional Neural Networks (CNNs) [2]–[7]. These CNN-based methods, despite their diverse model architectures and impressive outcomes, typically rely on spatial locality and compute displacements by examining correlation volumes for flow prediction (Fig. 1(a)). Very recently, the vast success of Transformer [8]–[10] stimulates the emergence of attention-based paradigms for various tasks in language, vision, speech, and more. It appears to form an unanimous endeavor in the deep learning community to develop unified methodologies for solving problems in different areas. Towards unifying methodologies, less inductive biases [8] are introduced for a specific problem, which urges the models to learn useful knowledge purely from input data.

Manuscript created August, 2023; This work was developed by the IEEE Publication Technology Department. This work is distributed under the LaTeX Project Public License (LPPL) (http://www.latex-project.org/) version 1.3. A copy of the LPPL, version 1.3, is included in the base LaTeX documentation of all distributions of LaTeX released 2003/12/01 or later. The opinions expressed here are entirely that of the author. No warranty is expressed or implied. User assumes all risk.
A preliminary version of this work [1] has appeared in CVPR 2023 and selected as Highlight.
† *indicates equally contribution.*

Jumping on the bandwagon of unifying architecture, we study applying Vision Transformer [11] to the task of optical flow. The following question naturally arises: *What are the major limitations of existing CNN-based approaches?* Tackling this question can provide insights into the model design of optical flow, and motivate us to rethink the task from an attention-driven view. First, the concurrent CNN-based methods demonstrate inefficiency in modeling *global spatial dependencies* due to the intrinsic locality of the convolution operation. It usually requires a large number of CNN layers to capture the correlations between two pixels that are spatially far away. Second, CNN-based flow learners typically focus solely on modeling the flow between two consecutive frames, thereby overlooking the exploration of *temporal associations* within neighboring contexts. Consequently, these methods often yield weak predictions when confronted with significant photometric and geometric changes. Third, the existing training strategy usually requires a *tedious pipeline*. Performance guarantees heavily rely on excessive pre-training on extra datasets (*e.g.*, FlyingChairs [12], FlyingThings [13], etc). Without adequate pre-training procedures, the model converges with large errors.

In order to craft a Transformer architecture for scene flow that pursues performance guarantees, the question becomes more fundamental: *How to address these limitations using Transformer?* As responses to this question, we articulate the technical contributions to address above limitations:

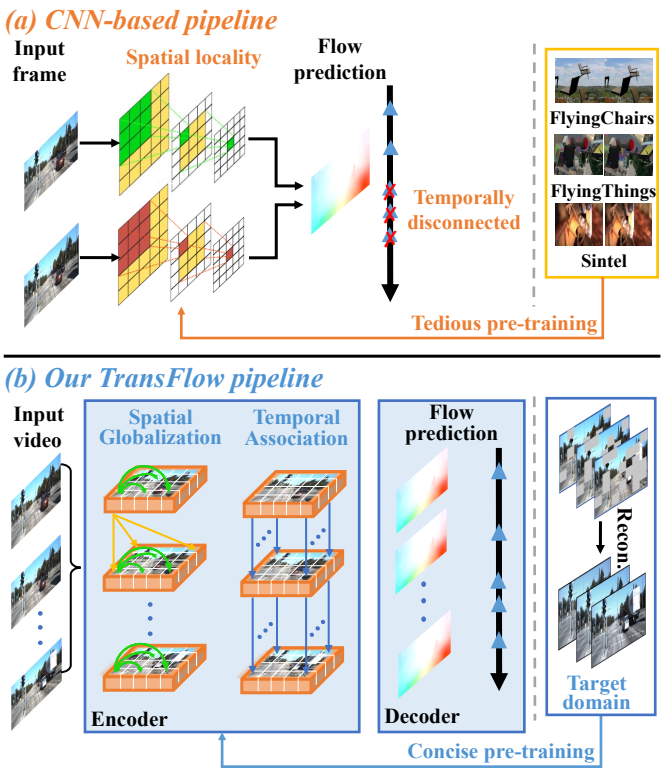


Fig. 1: **Conceptual comparison of flow estimation methods.** Existing CNN-based methods regress flow via local spatial convolutions, while TransFlow relies on *Transformer* to perform *global matching* (both *spatial* and *temporal*), and can easily be upgraded to 3D scene flow estimation.

- We introduce *spatial attention* in our approach to effectively capture global dependencies, ensuring precise correlation and reliable matching for flow estimation. Essentially, the spatial attention in Transformer facilitates the propagation of contextual cues from coherent regions to the surrounding areas with heavy-tailed noise, motion blurs, and large displacements. This mechanism significantly mitigates performance degradation in flow estimation by enabling effective information recovery.
- We explicitly model *temporal association* in dynamic scenes using multi-frame features extracted in the designed Transformer encoder. The correspondences among different frames are learned to generate the final estimated flow. One advantage is that when a specific region or object of a frame is occluded or blurred, neighboring frames can still effectively recover the missing information, leveraging the learned temporal association.
- To streamline the training process, we propose a concise *self-supervised pre-training* module that eliminates the need for complex and laborious multi-stage pre-training procedures. In particular, extended from MAE [14], we develop a masking strategy during the training to mask out visual tokens and learn strong pixel representations by reconstructing clean signals from corrupted inputs. We demonstrate the efficacy of this simple architecture by achieving superior performance compared to state-of-the-art (SOTA) baselines [4], [15]–[18].
- As a generic and flexible training algorithm, our framework can benefit object motion for *both 2D optical flow estimation and 3D scene flow estimation* tasks. In particular, for scene flow estimation, in addition to forcing truthful matching within the image plane, we further output the scene depth to learn geometric and contextual information that is helpful for matching corresponding 3D points, which can prevent matching ambiguity in the depth field and simultaneously achieve perfect matching of object positions and scales.

To summarize, we propose a pure transformer architecture that reformulates the typical optical flow and scene flow estimation. The proposed pipeline leverages a transformer-based framework, as illustrated in Fig. 1(b), to factorize pixel-wise flow learners while incorporating both spatial dependencies and temporal associations. By integrating these elements into the architecture, our method aims to enhance performance guarantees and improve the overall quality of 2D optical flow and 3D scene flow estimation.

This work serves as a substantial extension of our conference paper [1], building upon its foundations and introducing several significant enhancements in various aspects. First, we expand our algorithm to encompass more generic scenarios, allowing it to be applicable to both optical flow in 2D space and scene flow in 3D space (§3.5). Second, to showcase the the high flexibility of our algorithm, we provide additional experiments on the KITTI Scene Flow datasets (see Table 2 and Figure 7). Third, to thoroughly investigate the performance in downstream tasks, we provide further empirical results by applying our algorithm to the ImageNet VID, GoPro, and DeepStab datasets. These evaluations shed light on the capabilities of our algorithm in tasks such as video object detection, video frame interpolation, and video stabilization. Finally, in §5, we provide an in-depth discussion of the limitations and potential impacts of our method, which is expected to contribute to the research community and foster advancements in the field of object and scene motion analysis.

Remaining of the work is organized as follows: §2 a comprehensive literature review of the existing methods for optical flow estimation and scene flow estimation. It presents an overview of the concurrent approaches and discusses their strengths and limitations. §3 describes the model architecture of TransFlow. It explains the key components and mechanisms incorporated into the framework for optical flow estimation. In §4, we elaborate on the experimental setup and configuration settings used in our study. Concretely, §4.1 shows that our method achieves impressive results in popular flow estimation datasets (*e.g.*, Sintel [19] and KITTI 2015 [20]), and demonstrates its superiority over recent state-of-the-art approaches on 3D scene flow estimation; In §4.2, with a set of diagnostic experiment, our extensive experimental settings verify the effectiveness of our method and different optimization settings. In §4.3, we demonstrate the transferability and generalizability of TransFlow in modeling object motion for various downstream tasks. (*ie*, video object detection, video interpolation, and video stabilization), which can benefit from our method

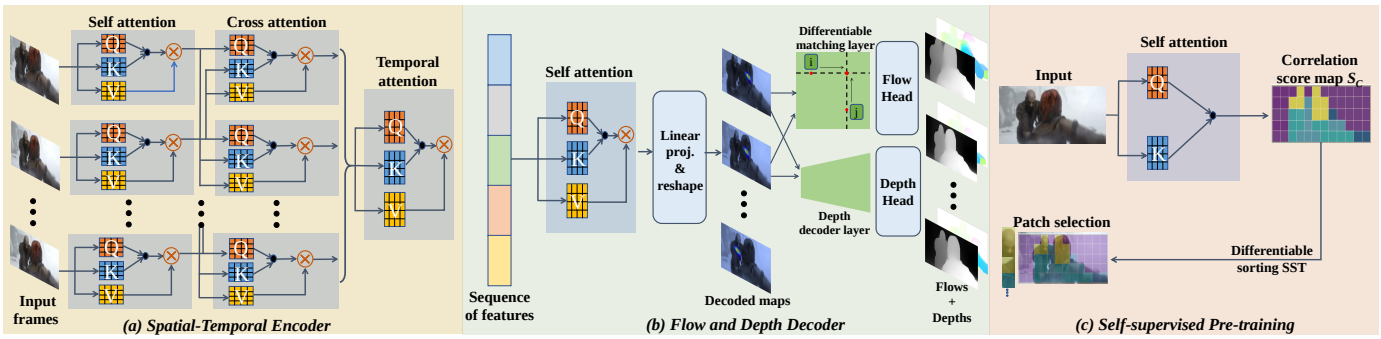


Fig. 2: **Overall model architecture.** It consists of three major components, a spatial-temporal encoder, a flow and depth decoder, and a self-supervised pre-training module. The spatial-temporal encoder jointly performs spatial globalization and temporal association among patch tokens. The flow and depth decoders decode the feature maps for multiple frames and generates the final optical flow and depth map. The pre-training module is designed to learn effective image representation in a self-supervised manner.

without bells and whistles. In the end, we make conclusions in §6 to highlight that, through the organization of these sections, this work is expected to show its potential for broader applications and pave the way for future research.

2 Important Knowns and Gaps

Optical Flow Learners. Traditionally, optical flow was formulated as an energy optimization problem for maximizing similarity between image pairs [21]–[23]. More recently, visual similarity is computed via the computationally expensive correlation of high-dimensional features encoded by convolutional neural networks [5], [12], [15], [16], [24], [25]. FlowNet [12] was the first end-to-end CNN-based network, which uses a coarse and refined branch for optical flow estimation. Its successive work, FlowNet2.0 [24], adopted a stacked architecture with warping operation, resulting in further performance enhancements. Following this, a series of works employed coarse-to-fine strategy and performed the estimation iteratively. For example, PWC-Net [5] developed a framework composed of stacked image pyramids, image warping and cost volumes; Hofinger et al., [25] replaced the image warping with a sampling-based strategy to improve the cost volume construction; Teed and Deng et al., [16] proposed to build a 4D cost volume for matching between all pairs of pixels and added a recurrent decoder for propagation. More recently, UpFlow [4] designed a self-guided upsample module to tackle the interpolation blur problem between pyramid levels and FlowFormer [15] proposed to integrate transformer encoder and encoded the cost tokens into the cost volume. However, the feature maps generated by these methods often suffer from limited receptive fields and are highly susceptible to outliers, rendering them less effective in capturing global motion clues. In contrast, our method adopts a different conceptual approach. We design a transformer-based structure that leverages both self-attention and cross-attention mechanisms, as well as temporal association, to enable effective global matching. Moreover, we demonstrate the possibility of achieving competitive results without the need for costly training pipelines by employing the introduced self-supervised learning paradigm.

Attention in Optical Flow. While becoming the standard for natural language processing tasks [26], [27], a flurry of research has successfully introduced Transformers to computer vision. Inspired by successes in image classification [11], [28], multiple recent architectures have been trying to combine CNN-based architectures with self-attention, including detection [29], [30], image restoration [31], video inpainting [32] and flow estimation [33]. Recently, there have been several attempts to apply Transformer structures to boost the performance of optical flow. Generally for these works, attention is applied in tandem with CNNs to compensate for the absence of image-specific inductive bias [32], [34]–[36]. A stack of Transformer blocks are added between CNN encoder and decoder for preventing blurry edges [32] and a combination of light-weight self-attention and convolutions are unitized to improve the inconsistent segmentation output [34]. Among these works, a closely related study to ours is FlowFormer [15], which utilizes a transformer-based structure to embed the 4D cost volume into a cost embedding and subsequently decodes it with a convolutional recurrent network. However, there are notable distinctions between their work and ours. First, they can only model two frames but ignore long-range temporal correlations. Second, we enable efficient and effective pre-training in optical flow, which fully explores the potential of the transformer model to rely on the target datasets only. More comparisons in 2D optical flow estimation, 3D scene flow estimation and downstream tasks of these two approaches are presented in the experimental sections.

3D Scene Flow Estimation. Scene flow, as introduced by Vedula et al. [37], aims to estimate not only the motion field in the image plane, but also a dense 3D motion field for each point in the scene, thereby providing a more comprehensive understanding of the scene structure. The most common approach is to simultaneously estimate the 3D scene structure and the 3D motion of each point using stereo images [38]–[40]. Early techniques relied heavily on variational formulations and energy reduction, resulting in limited accuracy and long running times [41], [42]. Recently, CNN-based models have been proposed for both supervised [13], [43] and unsupervised/self-supervised techniques. While supervised techniques rely on large synthetic datasets and

in-domain data to achieve state-of-the-art performance with real-time efficiency, unsupervised/self-supervised learning algorithms [44], [45] address the challenge of obtaining sufficient ground truth data, albeit with lower accuracy. Besides image-based approaches, scene flow estimation based on RGB-D [46], [47] or 3D point clouds [48], [49] has also been explored, utilizing the available 3D sparse points as input.

As a unified framework, our algorithm can be seamlessly crafted into any 3D scene flow learner. With a unified and sophisticated self-learning strategy and occlusion-aware loss, we address both tedious multi-stage synthetic training and the inefficiency of occlusion reasoning across successive frames. Our algorithm significantly improves the performance of 3D scene flow estimation, which is a more flexible and practical, but much more challenging, setup that jointly reasons 2D motion and 3D structures.

3 Framework Overview

The task of scene flow estimation is to estimate a series of dense 3D displacement fields from a sequence of consecutive frames. The overview of the proposed model architecture is illustrated in Fig. 2. Our approach is a transformer model that consists of three major components, a *spatial-temporal encoder*, a *flow and depth decoder* and a *self-supervised pre-training module*. The spatial-temporal encoder jointly performs spatial globalization and temporal association to effectively capture the correlations among frames and propagate global flow features. The flow and depth decoder decodes the feature maps for multiple frames which are then used to generate the estimated optical flow and scene depth. The pre-training module is designed to learn the effective image pixel representation in a self-supervised manner, which eliminates the complex and laborious multi-stage pre-training procedures widely used in previous approaches. The inclusion of an estimated depth branch enables the incorporation of geometric consistency to enhance flow estimation and elevate the 2D optical flow to 3D scene flow estimates.

3.1 Problem Definition

The input is a sequence of frames $X \in R^{T\times H\times W\times C}$, which consists of T frames and C channels with (H, W) as the resolution. Following the work in ViT [11], we split each frame into N fixed-size non-overlapping patches $\mathbf{x}_p$, where $p \in \{1, 2, ..., N\}$, $h \times w$ is patch size, and $N = \frac{H}{h} \times \frac{W}{w}$ ensures the N patches span the entire frame. The purpose of TransFlow is to output a sequence of feature map for each frame, which is then used to generate the per-pixel displacement field f between any source and target frames.

3.2 Spatial-Temporal Encoder

3.2.1 Spatial Globalization

The existing CNN-based flow learners demonstrate inefficiency in modeling global spatial dependencies due to the intrinsic locality of the convolution operation. However, the global spatial correlation is important information which enables effective contextual cue propagation from coherent regions to the surroundings with heavy-tailed noise, motion blurs, and large displacements, preventing performance degradation in estimating the optical flow.

In this work, we apply a spatial attention mechanism between two consecutive frames to capture the global spatial dependencies among the pixels. In particular, similar to ViT [11], each patch $\mathbf{x}_p$ (Table 6b for patch size) is first converted into a d-dimensional embedding vector $\mathbf{e}_p \in R^d$ with a projection matrix $\mathbf{W}_e$. The final input sequence of patch embeddings is denoted as:

$$\begin{aligned} \mathbf{z}^0 &= [\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N]; \\ \mathbf{e}_p &= \mathbf{W}_e \cdot \mathbf{x}_p + \mathbf{p}_p, \end{aligned} \tag{1}$$

where $\mathbf{p}_p$ is a set of learnable position embeddings (Table 6c) to retain the positional information, which is significant to motion clues. The patch tokens are passed through a series of Multi-head Self-Attention (MSA), Multi-head Cross-Attention (MCA) and MLP layers:

$$\begin{aligned} \mathbf{y}^{\ell} &= \mathrm{MLP}\left(\mathrm{MSA}\left(\mathbf{z}^{\ell-1}\right)\right) + \mathbf{z}^{\ell-1}; \\ \mathbf{y}'^{\ell} &= \mathrm{MLP}\left(\mathrm{MSA}\left(\mathbf{z}'^{\ell-1}\right)\right) + \mathbf{z}'^{\ell-1}; \\ \mathbf{z}^{\ell} &= \mathrm{MLP}\left(\mathrm{MCA}\left(\mathbf{y}^{\ell}, \mathbf{y}'^{\ell}\right)\right) + \mathbf{y}^{\ell}. \end{aligned} \tag{2}$$

Essentially, the self-attention is used to capture the global pixel dependencies within the same frame, while the cross-attention is designed to communicate the information between two adjacent frames. The self-attention and cross-attention are defined as:

$$\begin{aligned} \mathrm{MSA}\left(\mathbf{z}\right) &= softmax\left(\mathbf{Q}\cdot\mathbf{K}^T/\sqrt{d}\right)\cdot\mathbf{V}; \\ \mathrm{MCA}\left(\mathbf{y}, \mathbf{y}'\right) &= softmax\left(\mathbf{Q}\cdot\mathbf{K}'^T/\sqrt{d}\right)\cdot\mathbf{V}', \end{aligned} \tag{3}$$

where $\mathbf{Q} = \mathbf{z}\cdot\mathbf{W}^Q$, $\mathbf{K} = \mathbf{z}\cdot\mathbf{W}^K$ and $\mathbf{V} = \mathbf{z}\cdot\mathbf{W}^V$ are the query, key and value embedding matrices of the local frame in MSA. $\mathbf{K}' = \mathbf{z}'\cdot\mathbf{W}'^K$ and $\mathbf{V}' = \mathbf{z}'\cdot\mathbf{W}'^V$ are the key and value embedding matrices from adjacent frame in MCA.

The output from the attention layers is the refined correlation features, and we interleave the self-attention and cross-attention layers by L times. Through the joint aggregation, we benefit from the feature aggregation via local frame in self-attention, which is further facilitated via adjacent perspectives in cross-attention, as depicted in Fig. 2 (a).

3.2.2 Temporal Association

Previous approaches for estimating optical flow from each pair of adjacent frames are less effective as they ignore the inherent nature of long-range temporal associations. The motion estimation in discontinuous and occluded regions cannot be well modelled under modern architectures. To better capture high-level temporal information in the flow tokens, we learn the token embeddings by jointly modeling the temporal association with the spatial attention described above. As a result, each transformer layer can measure long-range interaction between input embeddings. Specifically, given a sequence of attentioned features from video clips consisting of T frames (see related experiments in Table 6d), we iteratively choose one as query and the rest as key features to compute the temporal attention using Softmax, which is similar as Eq. 3. The resulted d-dimensional

embedded feature volume is then passed to the following transformer decoding block. By learning temporal features in this manner, we allow for the accumulation of temporal information into each frame, effectively capturing temporal associations across multiple frames, which is further illustrated in Fig. 2 (a).

3.3 Flow Decoder

Different from the traditional Transformer decoder, our decoder is designed to decode the feature maps of each frame (Fig. 2 (b)). These decoded features are then utilized in obtaining the final flows. Therefore, our decoder aims to generate multiple feature maps at the same time, instead of autoregressive decoding. There are two major advantages in such a design. First, simultaneous decoding allows us to remove the encoder-decoder cross-attention in the traditional Transformer decoders. Second, beam search is no longer needed, which makes our decoding process much more efficient. Therefore, in this work, we adopt a structurally symmetric design with the Transformer encoder. In other words, our decoder has the same self-attention architecture as our encoder except that the input to the decoder is the latent cost embedding from the encoder.

Given the decoded feature maps between two consecutive frames, we compare the feature similarity by computing the correlation following [50]. To enable the end-to-end training, we apply the differentiable matching layer [51] to identify the correspondence from the adjacent frames. The final flow f can then be generated from the correspondences. During training, we further conduct an additional occlusion detection [52] by performing a forward consistency checking and considering pixels to be occluded if the mismatching in both frames is too large. Consequently, the occlusion areas M_{occ} is computed as $M_{occ} = f_D(I_s - I_t(x + f))$, where f_D can be any function that measures the photometric distance. f is the estimated forward optical flow. I_s and I_t are the source and target images/frames. The overall objective can be formulated as:

$$L = \sum_{i=1}^{R} (1 - M_{occ}) \gamma^{(i-R)} \left\| f_{gt} - f \right\|_1 , \tag{4}$$

where $|| \cdot ||_1$ denotes the L_1 norm, R is the total number of the training iterations. γ is a hyperparameter that controls the weight of the loss among different iterations. f_{gt} stands for the ground-truth flow map.

3.4 Self-supervised Pre-training

The performance of the existing flow learners heavily relies on excessive pre-training on extra synthetic datasets, followed by fine-tuning on the target domain. Insufficient pre-training on large-scale data often leads to models that converge with substantial errors. Hence, it becomes crucial to develop an efficient and effective pre-training strategy that enhances the performance of the subsequent optical flow task.

Drawing inspiration from the recent Masked Autoencoder (MAE) [14], we introduce a masking strategy in self-supervised pre-training that adaptively masks out patch tokens and learns pixel representations by reconstructing clean signals from corrupted inputs. Specifically, we learn a score map for patch selection to choose the most informative patches as masked tokens under a determined ratio, as opposed to randomly masking in [14] or uniformly masking in [53]. In our diagnostic experiments (Table 6e and 6f), we will demonstrate that the capability of our self-learning paradigm in recovering crucial regions can be enhanced. More specifically, we adopt multiple layers of self-attention blocks taking all the patch token embeddings as input. The attention map is then calculated as the correlation between the query embedding from the image token Q and all key embeddings across all patches K. The correlations are then followed by a Softmax activation to generate the correlation score map S_c, as depicted in Eq. 5. The correlation score map output from the final layer of the attention blocks will be utilized to guide our strategic masking learning:

$$S_c = softmax(Q \cdot K^T) \tag{5}$$

The obtained correlation score map S_c is then modeled as a ranking problem to be sorted in an ascending order to select the most informative tokens for masking, as in Fig. 2 (c). In order to prevent the discrete property of the $argsort$ operation, we instead utilize the soft sort operation in [54] denoting as $SST(\cdot)$:

$$SST(\cdot) = softmax(\frac{|sort(S_c)\mathbf{1}^T - \mathbf{1}(S_c)^T|}{\tau}) \tag{6}$$

where $|\cdot|$ calculates element-wise absolute and τ is the temperature constant that is set to 0.1 to control the degree of approximation. With the differentiable sorting, we are able to identify and retain the most significant token candidates and learn the score map as network weights in conjunction with our primary flow estimation task.

3.5 Extension for 3D Scene Flow Estimation

Our training algorithm is also applicable for 3D scene flow estimation models, which can be formulated to estimate the 3D coordinates $P_i = (x, y, z)$ and the flow motion vector $F_i = (\Delta x, \Delta y, \Delta z)$ for every pixel $p_i = (u, v) \in I_i$. Unlike optical flow estimation, which primarily captures pixel displacement in the image plane, our approach incorporates depth estimation by fine-tuning a pre-trained depth network [64], with a geometric consistency loss so that each object and scene achieve perfect matching of positions and scales simultaneously (see Fig. 3).

The depth network takes two adjacent frames (I_t, $It - 1$) as input to build a cost volume to capture the geometric compatibility at different depth values between the pixels from the current and nearby frames. The ResNet-18 based depth encoder and decoder then process the computed cost volume to produce a depth image at the current frame I_t. Pretrained on KITTI dataset, and constrained with the pixel reconstruction loss L_{pixel} between the current frame I_t and synthesized frame $I_{t-1->t}$, depth consistency loss across multi-frame output $L_{consistency}$ to ensure scale consistency, and a smoothness loss L_{smooth} to eliminate depth discontinuities, the depth estimation can achieve reliable and consistent estimation, which is necessary to recover the geometric information in the 3D scene flow.

For each adjacent frame pair $(i, j) \in I$, we target at retrieving the 3D scene flow F_i which indicates the 3D

Training Data	Method	Sintel (train) clean	Sintel (train) final	KITTI-15 (train) F1-epe	KITTI-15 (train) F1-all	Sintel (test) clean	Sintel (test) final	KITTI-15 (test) F1-all
C+T	PWC-Net [CVPR18] [5]	2.55	3.93	10.35	33.7	-	-	-
	HD3 [CVPR19] [55]	3.84	8.77	13.17	24.0	-	-	-
	LiteFlowNet [TPAMI20] [56]	2.24	3.78	8.97	25.9	-	-	-
	RAFT [ECCV20] [16]	1.43	2.71	5.04	17.4	-	-	-
	FM-RAFT [ECCV21] [57]	1.29	2.95	6.80	19.3	-	-	-
	GMA [ICCV21] [58]	1.30	2.74	4.69	17.1	-	-	-
	Separable Flow [ICCV21] [59]	1.30	2.59	4.60	15.9	-	-	-
	Flow1D [ICCV21] [60]	1.98	3.27	5.59	22.95	-	-	-
	AGFlow [AAAI22] [61]	1.31	2.69	4.82	17.0	-	-	-
	KPA-Flow [CVPR22] [62]	1.28	2.68	4.46	15.9	-	-	-
	Flowformer [ECCV22] [15]	1.01	2.40	4.09	14.72	-	-	-
	TransFlow [CVPR23] [1]	0.93	2.33	3.98	14.40	-	-	-
	Ours	**0.91**	**2.32**	**3.96**	**14.35**	-	-	-
C+T+S+K (+H)	PWC-Net [CVPR18] [5]	-	-	-	-	4.39	5.04	9.60
	HD3 [CVPR19] [55]	1.87	1.17	1.31	4.1	4.79	4.67	6.55
	LiteFlowNet [TPAMI20] [56]	1.35	1.78	1.62	5.58	4.54	5.38	9.38
	RAFT [ECCV20] [16]	0.77	1.20	0.64	1.5	2.08	3.41	5.27
	FM-RAFT [ECCV21] [57]	0.86	1.75	0.75	2.1	1.77	3.88	6.17
	Separable Flow [ICCV21] [59]	0.71	1.14	0.68	1.57	1.99	3.27	4.89
	Flow1D [ICCV21] [60]	(0.84)	(1.25)	-	(1.6)	(2.24)	(3.81)	(6.27)
	KPA-Flow [CVPR22] [62]	(0.60)	(1.02)	(0.52)	(1.10)	(1.35)	(2.36)	(4.60)
	Flowformer [ECCV22] [15]	(0.48)	(0.74)	(0.53)	(1.11)	(1.16)	(2.09)	(4.68)
	TransFlow [CVPR23] [1]	(0.42)	(0.69)	(**0.49**)	(1.05)	(1.06)	(2.08)	(4.32)
	Ours	(**0.40**)	(**0.66**)	(**0.49**)	(**1.03**)	(**1.02**)	(**2.05**)	(**4.31**)

TABLE 1: **Quantitative comparisons with state-of-the-arts.** We follow existing works to compare the results on two standard benchmarks Sintel and KITTI-15. "C+T" denotes training only on FlyingChairs and FlyingThings datasets and testing on others for the generalization ability. "C+T+S+K(+H)" denotes training on mixed datasets and testing on Sintel and KITTI-15 for evaluation. Recent works [15], [60], [62] including HD1K [63] dataset for training are marked with brackets in results. Our self-learning paradigm helps to get superior results by avoiding tedious pre-training stages on "C/T" and simplifying the training pipeline. The best and second best results are highlighted in bold and underlined. See §4.1 for details.

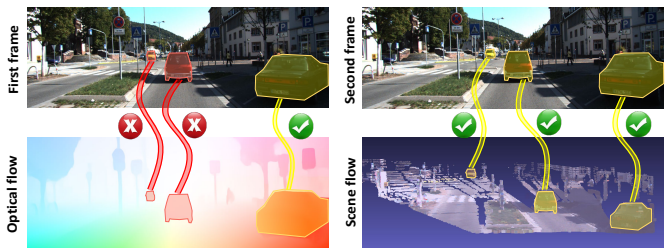


Fig. 3: **Comparison of 2D optical flow and 3D scene flow.** While optical flow can estimate the motion field on the frame plane, but often encounters challenges when the objects are in different sizes while moving. Scene flow uses scene depth information to overcome the limitations of optical flow.

motion field between a pair of video frames [65]. To achieve that, we add a separate depth net pretrained on KITTI to estimate the scene depth D_i at the current frame. Therefore, 3D scene point $P_i(x)$ can be expressed using the camera intrinsic K and the depth D_i by:

$$P_i(x) = D_i(x)K_i^{-1}x \tag{7}$$

the scene motion from the i's camera coordinate to the j's camera coordinate can be expressed as:

$$F_i(x) = P_{i\to j}(x) = R_j^{\top}\left(R_i c_i(x) + t_{i\to j}\right) \tag{8}$$

where c_i is the 3D coordinate from camera i. R_i and R_j are the rotation matrices for camera i and j, respectively. $t_{i->j}$ is the translation vector from camera i to camera j. By applying the rotation matrix R_i and the translation vector $t_{i->j}$ to the 3D point c_i (x), the 3D point is mapped from the coordinate system of i to a common world coordinate. Then, by multiplying the result by R_j^T (the transpose of R_j), the world coordinate is transformed into the local coordinate system of camera j.

Then, the optical flow can be generated by projecting the scene motion back to the pixel location in frame I_j as:

$$p_{i\to j}(x) = \pi\left(K_j P_{i\to j}(x)\right) \tag{9}$$

Optical Flow. To accommodate both the optical flow estimation and the depth estimation, and to coerce them with geometric consistency to produce more consistent depth and flow, we extend the flow estimation objective in Eq. 4 by adding an extra consistency term between the displacements on the estimated flow and the projected depth points:

$$L_{flo} = \sum_{i=1}^{R}(1 - M_{occ})\gamma^{(i-R)}\left\|f_{gt} - f\right\|_1 + ||p_{i\to j}(x) - f_{i\to j}(x)||_1 \tag{10}$$

where the second term (consistency loss) penalizes the spatial difference between the pixel displacement of the flow and the projected depth. In this way, 3D motion ambiguity is addressed in our unified training system.

Scene Depth and 3D Scene Flow. We use the smooth $L1$ loss as L_{dep} to supervise the scene depth and the $L1$ loss as L_{scn} to supervise the scene flow in 3D space as additional constraints when the ground truths are available.

Joint Optimization. The supervised losses for joint optimization on optical flow, depth map, and 3D scene flow are defined as $L_{joint} = L_{dep} + L_{flo} + L_{scn}$. Note that the joint

losses are only imposed on synthetic datasets where ground truth annotations are available for all three components and correspond perfectly. In Sec. 4.1, we empirically investigate the performance of our training algorithm in scene flow estimation, and in Sec. 4.2, we specifically investigate the effects of applying joint optimization.

4 Experiments

Datasets. Existing flow estimation approaches require a tedious training pipeline which first pre-train the models on FlyingChairs ("C") [12] and FlyingThings ("T") [13], and then fine-tune the trained models on Sintel ("S") [19] and KITTI 2015 ("K") [20]. Without the progressive steps, the flow estimation performance will get a significant degradation. Simplifying the cumbersome procedures, we rely on training optical flow task on the target domain without excessive pre-training stages. MPI-Sintel [19] dataset is rendered based on animated movies and is split into *Clean* and *Final* pass. KITTI-15 [20] contains 200 training and 200 testing road scenes with sparse ground truth flow, where images are captured via stereo cameras. For datasets provide only pairwise flow (e.g., KITTI-15), we access raw data in the self-supervised pre-training.

To evaluate the 3D scene flow extension, we use multiple datasets for a thorough verification. We use the KITTI scene flow split [20] as a test set because it provides ground truth labels for disparity and scene flow for 200 images. The FlyingThings3D dataset [13] provides 3824 RGB-D image pairs rendered with multiple randomly moving objects from ShapeNet [66], which is also included to evaluate recent methods with different input modalities. The nuScenes dataset [67] is a large-scale autonomous driving dataset in urban environments with dynamic scenes. In the absence of official scene flow annotations, we have followed the data processing in [68] to create a pseudo-ground truth scene flow for verification in 150 test scenes. In addition, we incorporate the large-scale synthetic dataset Virtual KITTI 2 [69], which mimics the real KITTI scenes and provides corresponding dense depth, camera motion, optical flow and scene flow annotations, to evaluate the generalization ability when transferring from synthetic to real estimation.

Implementation Details. We stack 12 transformer blocks in the encoder to adaptively learn the feature encoding. To keep the resolution to be the same as the input, we adopt the convex upsampling technique in [16] to upsample the prediction. The model is first pre-trained in a self-learning paradigm with a learning rate of $1e$-4 and then the entire network is continuously trained on the target domain with a batch size of 6 and learning rate of $12.5e$-5 for 140K steps. For the hyperparameters, γ is set to 0.8 and the masking ratio is 50%. The detailed diagnostic experiments of these hyperparameters are provided in §4.2.

Evaluation Metrics. The main evaluation metrics for 2D optical flow, used by the Sintel datasets, is the average end-point error (*AEPE*), which denotes the average pixel-wise flow error. The KITTI dataset adopts *F1-epe* (%) and *F1-all* (%), which refers to the percentage of flow outliers over all pixels on foreground regions and entire image pixels. For 3D scene flow evaluation, we follow the standard evaluation metrics of the KITTI scene flow benchmark [20] to evaluate the accuracy of the disparity for the first frame ($D1$-all), the disparity for the second frame mapped to the first frame ($D2$-all), as well as the outlier rate within the estimated scene flow ($SF1$-all).

4.1 Comparison to the State-of-the-Art Methods

Quantitative Evaluations on Optical Flow. We compare our approach with existing supervised flow estimation methods on the most popular optical flow benchmarks ($i.e.$ Sintel and KITTI). Without tedious multi-stage flow estimation pre-training on synthetic benchmarks FlyingChairs and FlyingThings, our designated framework beats existing state-of-the-art methods, as demonstrated by the quantitative results in Table 1. As shown, for generalization ability, we train our TransFlow on the FlyingChairs and FlyingThings (C+T) and directly evaluate it on the Sintel and KITTI-15 without further fine-tuning. Our TransFlow depicts the best result with the smallest errors among all compared methods on both datasets. Specifically on the Sintel dataset, we achieves **0.91** and **2.32** *AEPE* on the clean and final pass, which is **0.52** and **0.39** lower than the widely used method RAFT [16]. On the KITTI-15 dataset, we reduce the *F1-all* error by **17.5%** of RAFT [16].

When following the introduced self-learning paradigm on the target datasets and evaluate on the Sintel test set, our method achieves a **1.02** and **2.05** *AEPE* on the Sintel clean and final pass, which is **51%** and **40%** lower than RAFT [16], respectively. Similarly on the KITTI-15 benchmark, our approach performs a **4.31** *F1-all* score in errors, which is **0.96** and **0.37** lower than recent RAFT [16] and FlowFormer [15], respectively. However, RAFT [16] and FlowFormer [15] both require a multi-stage flow estimation pre-trainings before training on formal C+T+S+K or C+T+S+K+H datasets. In contrast to the prevailing methodologies, our proposed training pipeline delivers superior performance by employing streamlined and more effective procedures.

Qualitative Evaluations on Optical Flow. We sample test samples from Sintel `val` set and provide the corresponding optical flow estimation of the state-of-the-art FlowFormer [15] and our TransFlow in Fig. 4. From the visual comparison, it is evident that TransFlow exhibits superior capabilities in distinguishing occluded regions and producing clearer boundaries, especially for small and thin objects. This improved performance can be attributed to our spatial globalization and temporal association considerations, which are embedded in the design of TransFlow. These enhancements demonstrate the efficacy of our designed structures in spatial attentions, temporal associations, and novel strategic masking strategy in improving flow reasoning. To further validate the effectiveness of TransFlow, we also evaluate its flow estimation performance on the real-world driving dataset KITTI `val` set, as in Fig. 5. Similar to the observations on the Sintel dataset, our proposed flow estimation framework demonstrates enhanced stability in distinguishing occluded and small objects, such as persons, wheels, and traffic signs.

Compared with the existing algorithms, which require significant efforts in tedious pre-training pipeline and re-training / fine-tuning for the new target domain, an advantage of our TransFlow is that we break the limitations in

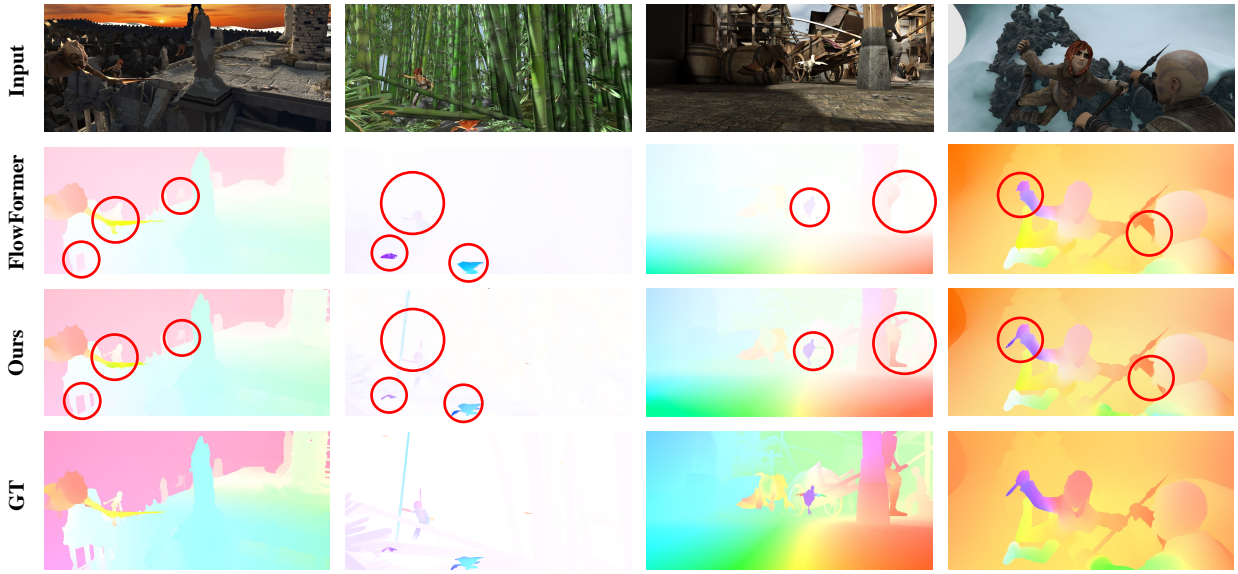


Fig. 4: **Qualitative results of optical flow** on Sintel `val` set. Given the target frame, we show the results of the state-of-the-art FlowFormer [15], our TransFlow results, and the provided ground truth flow. ○ highlights comparing details. See §4.1 for details.

Fig. 5: **Qualitative results of flow estimation** on KITTI `val`. Our TransFlow shows better performance on the boundaries of thin and fine objects (*e.g.*, wheels and traffic signs). The overall performance is close to the provided ground truth. ○ highlights comparing details. See §4.1 for details.

terms of being able to generalize well to unseen domains, which is shown on the DAVIS [70] dataset in Fig. 6. It can be observed that TransFlow performs well in challenging and complex scenarios with multiple occluded and blurred regions compared to the state-of-the-art FlowFormer [15].

Scene Flow Estimation Evaluations. We first evaluate the results of scene flow estimation by upgrading our optical flow estimation to 3D space, compared to the direct baseline scene flow methods on the KITTI Scene Flow 2015 benchmark, as shown in Table 2. The scene flow inferred by our method achieves state-of-the-art accuracy among un-/self-supervised methods. Our proposed approach yields a substantial improvement in accuracy compared to the second best method. Specifically, it achieves a remarkable 22.9% reduction in error for depth estimation and a notable 21.6% reduction in error for 3D scene flow estimation.

We also performed a performance comparison with several recent methods that utilize different modalities of input information on the FlyingThings3D dataset in Table 3. By incorporating both modalities of 2D from optical flow and 3D information from scene depth, our method significantly outperforms all other image-only and 3D point-only methods. The proposed pipeline also outperforms RAFT-3D [78], which takes dense RGB-D frames as input, demonstrating the superior performance of our trustworthy and consistent scene flow estimation.

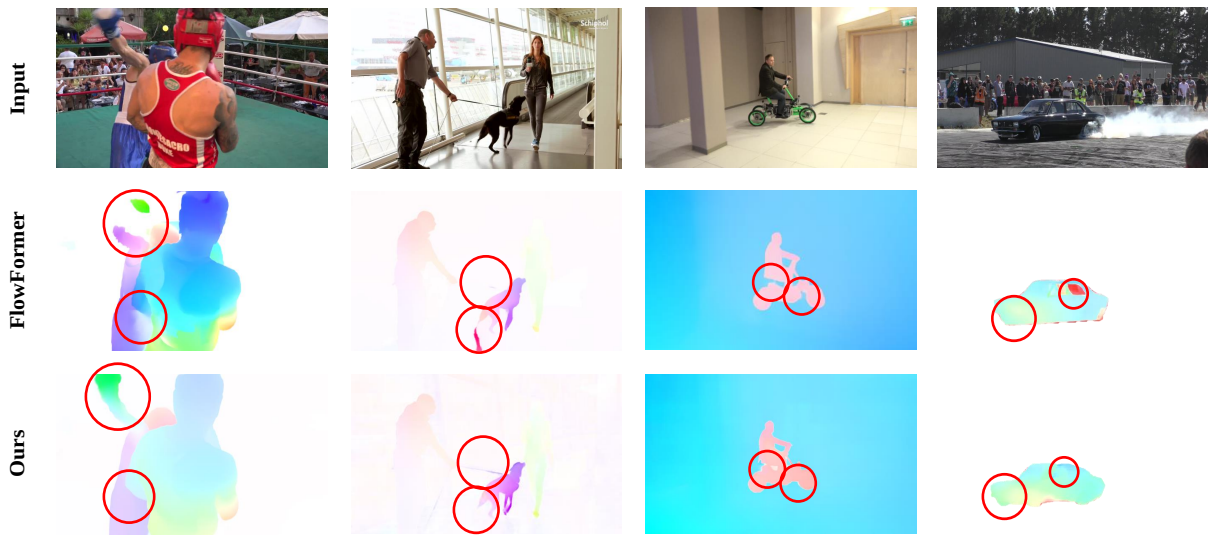


Fig. 6: **Qualitative results of flow estimation** on DAVIS. With our efficient and effective training pipeline, our TransFlow can also generalize well on unseen domains. ○ highlights comparing details. See §4.1 for details.

Method	D1-all	D2-all	SF1-all
DF-Net [71]	46.50	61.54	73.30
GeoNet [72]	49.54	58.17	71.32
EPC [73]	26.81	60.97	(>60.97)
EPC++ [74]	23.84	60.32	(>60.32)
Mono-sf [75]	31.25	34.86	47.05
Multi-sf [76]	27.33	30.44	39.82
Ours	**21.06**	**23.91**	**31.20**

TABLE 2: 3D scene flow evaluation on KITTI 2015 Scene Flow training split. Our upgraded 3D scene flow estimation significantly outperforms both multi-task CNN methods and recently published scene flow methods on all three metrics.

Method	Information	$EPE_{2D}\downarrow$	$EPE_{3D}\downarrow$
FlowNet2.0 [24]	2D	5.05	-
PWC-Net [5]		6.55	-
RAFT [16]		3.12	-
FlowNet3D [48]	3D	-	0.151
FLOT [77]		-	0.170
RAFT-3D [78]	2D + 3D	2.46	0.062
Ours		**2.35**	**0.054**

TABLE 3: **Performance comparison on the val split of the FlyingThings3D dataset.** We can see from the 2D and 3D EPE that our method outperforms other recent approaches that use either 2D information, 3D information, or both.

We perform the evaluation of scene flow estimation on more synthetic and real datasets in Table 4. We train our and compared methods on the Virtual KITTI 2 (VKITTI 2) and nuScenes scene flow datasets and evaluate the performance on the test scenes. As shown in Table 4, our method achieves superior performance on both synthetic VKITTI 2 and real nuScenes, improving the strong 3D point-based baseline FlowNet3D [48] by 39.0% and 53.5% in EPE_{3D}, respectively. The proposed method also improved the SOTA method [76] which incorporates both 2D and 3D information, by 8.8% and 12.0% in EPE_{3D}, which is consistent with the conclusion in the FlyingThings3D and KITTI datasets.

Method	Virtual KITTI 2		nuScenes	
	$EPE_{3D}\downarrow$	$Acc_{5\%}\uparrow$	$EPE_{3D}\downarrow$	$Acc_{5\%}\uparrow$
FlowNet3D [48]	0.136	22.37	0.505	2.12
Mono-sf [75]	0.098	43.32	0.339	36.79
Multi-sf [76]	0.091	51.04	0.267	45.13
Ours	**0.083**	**58.90**	**0.235**	**49.57**

TABLE 4: **Scene flow performance of our method across more datasets and metrics.** Here EPE denotes the endpoint error which is smaller values are better, $Acc_{5\%}$ denotes strict accuracy which means larger values are better.

Method	Virtual KITTI → KITTI		Virtual KITTI → nuScenes	
	$EPE_{3D}\downarrow$	$Acc_{5\%}\uparrow$	$EPE_{3D}\downarrow$	$Acc_{5\%}\uparrow$
FlowNet3D [48]	0.229	10.02	0.810	2.03
Mono-sf [75]	0.101	47.39	0.513	27.95
Multi-sf [76]	0.093	51.26	0.439	32.79
Ours	**0.087**	**53.10**	**0.414**	**34.18**

TABLE 5: **Scene flow performance comparisons on two Synthetic-to-Real dataset transfer.** Methods are trained on synthetic domain and tested on real-world scenarios to verify the generalization ability. EPE_{3D} and $Acc_{5\%}$ are scene flow evaluation metrics where ↓ and ↑ indicate negative and positive polarity, respectively.

In Table 5, a comparison of domain adaptation for synthetic-to-real scene flow estimation is performed to verify the generalization ability of the proposed method when all methods are directly transferred from the synthetic pre-training on the Virtual KITTI dataset without fine-tuning on real scenes such as KITTI and nuScenes. It can be seen that the proposed method achieves the best generalization in both conditions, surpassing the second best approach of over 6.5% and 5.7% in EPE_{3D}, respectively.

In Fig. 7, we present a visualization of the successive input frames along with the corresponding optical flow estimate, depth output, and scene flow estimates. This vi-

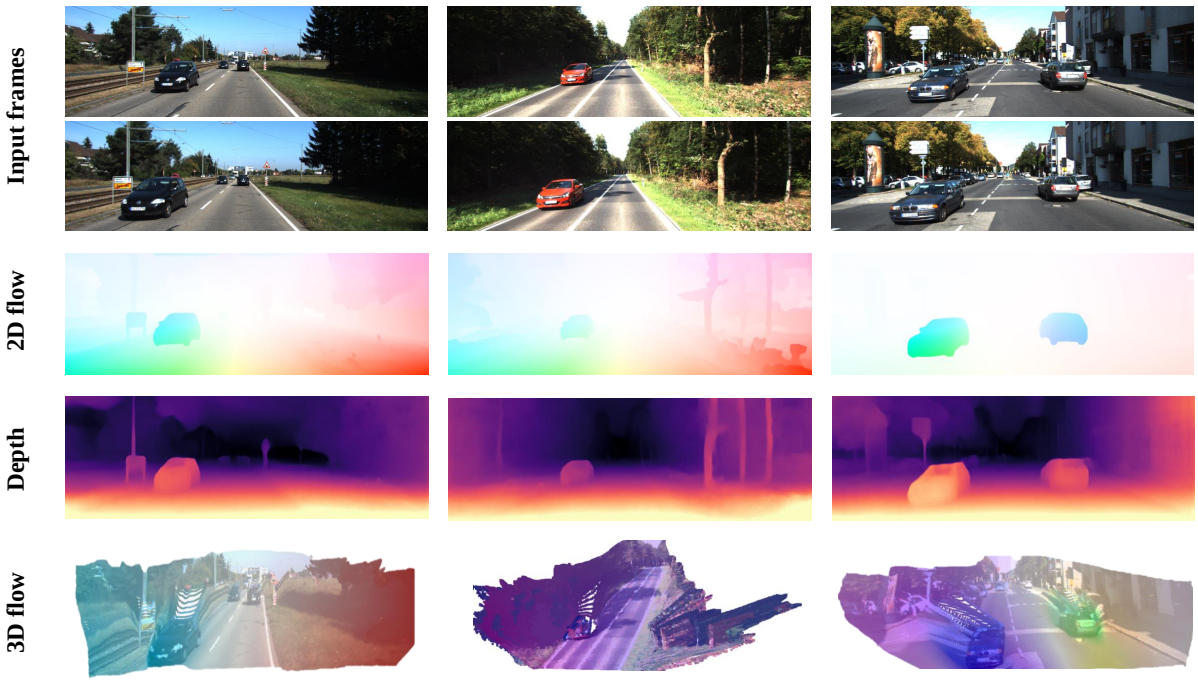


Fig. 7: **Qualitative results of 3D Scene Flow estimation** on the KITTI 2015 Scene Flow dataset. Given consecutive frames, we show the results from left to right: input images; depth outputs; optical flow estimation; scene flow estimation, where x-z coordinates are colored using the standard optical flow color coding for easier 3D visualization. See §4.1 for details.

sualization provides insight into how scene flow evolves over time at the same pixel location. It is evident that the developed model produces visually consistent scene flow motion over time, especially for foreground objects, indicating that the model effectively captures and predicts the motion of objects in the scene. By upgrading the 2D motion to 3D motion, the method extends its local understanding and estimation to entire scenes.

4.2 Diagnostic Experiments

Core Components. First, we study the efficacy of the core components of our algorithm in Table 6a to analyze their contributions to the final results. As shown, It can be seen that solely self-attention can yield limited performance as a baseline of our framework, while a combination of both self- and cross- attentions boost the performance thanks to the effective local feature aggregation between two views. The design of temporal association among multiple frames successfully alleviates the potential ambiguities in un-smooth and occluded regions, therefore bringing further improvement. The strategic masking reduces the reliance on multi-stage pre-training and benefits our self-learning paradigm. The occlusion consistency on the flow loss has a similar effect with the temporal association and adds additional performance gains to the results. From Table 6a, each component contributes to the improvement of performance, clearly depicting the effectiveness of our proposed components.

Patch Settings. We empirically evaluate the end-point error by adjusting the patch size (§3.2.1) in our TransFlow. As shown in Table 6b, when patch size are increasingly set from 4×4 to 8×8, the performance are slightly improved (**0.44** and **0.71** $\rightarrow$ **0.42** and **0.69**) in *EPE* and **1.11** $\rightarrow$ **1.05** in *F1-all* on Sintel and KITTI datasets, respectively. Nevertheless, when further enlarging the patch size to 16×16, the performance drops while the cost of computing continues to drop. The reason is that larger patch sizes lead to bigger kernel regions, resulting in the loss of global and long-range context information which is crucial for flow propagation.

Positional Embedding. The efficacy of different positional embeddings (§3.2.1) is rarely discussed in previous works. Consequently, we compare the performance of flow estimation under different positional embeddings (*e.g.*, Abs/Rel, Learnable/Fixed). As depicted in Table 6c, we observe that the learnable Abs Pos. achieves a slightly better result than the fixed Abs Pos. and a recent Positional Encoding Generator (PEG) [79], while showing a larger improvement than relative Pos. We suppose that the fixed sin-cos Abs Pos. can encode the flow features almost as well as the learnable Abs Pos. and PEG Pos., and positional embedding is indispensable in our setting. In addition, we believe the degradation from relative Pos. is due to the fact that object motion requires more absolute position encoding in order to locate and learn the motion, and global information is also more important than local relative information in this task, which is consistent with our claim.

Temporal Length. Table 6d shows that as we increase the number of frames (§3.2.2) fed into our temporal module, the error gets decrease since the network is able to incorporate longer temporal context and to avoid temporal artifacts and discontinuous estimation in the flow. However, Table 6d also demonstrates that the accuracy will become saturated once the number of temporal length is sufficient enough to

(a) Contribution of each core component (§3).

Config					Sintel (val)		KITTI (val)
Self	Cross	Temporal	Masking	Occ	Clean	Final	F1-all
✓					0.58	0.81	1.22
✓	✓				0.55	0.78	1.18
✓	✓	✓			0.49	0.73	1.18
✓	✓	✓	✓		0.44	0.70	1.07
✓	✓	✓	✓	✓	**0.42**	**0.69**	**1.05**

(b) Adjust Patch Size settings in the applied transformer architecture (§3.2.1).

Patch size	Sintel (val)		KITTI (val)
	Clean	Final	F1-all
4 × 4	0.44	0.71	1.11
8 × 8	**0.42**	**0.69**	**1.05**
14 × 14	0.46	0.77	1.14
16 × 16	0.59	0.85	1.21

(c) Different Positional Embedding methods (§3.2.1).

Pos. Embed	Sintel (val)		KITTI (val)
	Clean	Final	F1-all
Fixed Abs Pos.	0.43	0.71	1.06
Learnable Abs Pos.	**0.42**	**0.69**	**1.05**
PEG Pos.	**0.42**	0.70	1.07
Learnable Rel Pos.	0.47	0.76	1.10

(d) Variations in Temporal Length (§3.2.2).

Temporal length	Sintel (val)	
	Clean	Final
2 frame	0.47	0.75
3 frame	0.44	0.73
5 frame	**0.42**	0.69
7 frame	0.43	**0.68**

(e) Multiple Mask Sampling strategies (§3.4).

Sampling	Sintel (val)		KITTI (val)
	Clean	Final	F1-all
Block	0.48	0.74	1.10
Random	0.50	0.76	1.13
Uniform	0.45	0.72	1.09
Strategic	**0.42**	**0.69**	**1.05**

(f) Varying Masking Ratios in self-learning (§3.4).

Masking ratio	Sintel (val)		KITTI (val)
	Clean	Final	F1-all
30%	0.45	0.71	1.08
50%	**0.42**	**0.69**	**1.05**
70%	0.46	0.71	1.09
90%	0.49	0.77	1.16

TABLE 6: **A set of diagnostic experiments.** The adopted algorithm designs and settings are marked in red. See §4.2 for details.

cover visible motion. Considering that when increasing the temporal length from 5 to 7, there is no discernible difference in performance while the computational cost will increase correspondingly. Therefore, we choose 5-frame length as input.

Sampling Strategy. Table 6e shows the effect of various sampling strategies for masking (§3.4). We compare our strategic masking with block-wise masking [80], random masking [14] and uniform masking [53]. Under the same masking ratio, it can be seen the compared samplings have different levels of degradation compared to ours. The naive block-wise masking and random masking may destroy the tokens of vital regions of the original image that are required for object motion, whereas uniform masking may disregard the significance and relationship between tokens. On the contrary, our sampling has the ability to learn pixel representations effectively, which validates our claim.

Masking Ratio. Table 6f illustrates the effect of varying masking ratios (§3.4). It depicts that a suitable masking ratio (50% for ours) outperforms other settings with notable advantages. Such an empirical advantage can be explained by that the higher masking ratio may discard too much necessary information for self-learning paradigm via reconstruction to learn an effective image representation, whereas a low masking ratio may not be sufficient to increase the reconstruction difficulty and, consequently, the quality of the predicted flows.

Optimization settings. Table 7 provides more ablation analysis of different optimization settings. We ablate the effects of applying supervision only to 2D optical flow, to 2D optical flow with geometric consistency, to both optical flow and scene depth without and with geometric consistency, and finally to a joint optimization with supervision on 2D flow, scene depth, and 3D scene flow. It can be observed that the addition of scene depth supervision leads to a significantly lower EPE error and a higher strict accuracy in the estimation of scene flow, compared to using optical flow only. Geometric consistency between optical flow and scene depth has a small but consistent positive impact on performance for all settings. Joint optimization on optical flow, scene depth, and 3D scene flow leads to optimal performance, because scene flow can help to better estimate the motion generated by outliers or dynamic objects.

Optimization settings	Virtual KITTI 2	
	$EPE_{3D}\downarrow$	$Acc_{5\%}\uparrow$
Optical flow only	0.094	48.53
Optical flow **w/** geometric consistency	0.090	53.08
Optical flow + Scene depth	0.085	57.04
Optical flow + Scene depth **w/** geometric consistency	0.083	58.90
Joint optimization	0.078	61.72

TABLE 7: More ablation analysis of the optimization settings on the VKITTI 2 dataset. We ablate variations of optical flow optimization only, and add geometric consistency, depth supervision, or joint optimization of both 2D and 3D scene flow.

4.3 Downstream Tasks

High-quality flow estimation plays a crucial role in many video-based downstream tasks. We show here quantitatively that TransFlow generalizes well and can help further improve the state-of-the-art of various video-based tasks, including video object detection, interpolation, and stabilization.

Video Object Detection. We conduct our experiments on the ImageNet VID dataset [88] containing over 1M frames for training and more than 100k frames for validation. As shown in Table 8, adding TransFlow encoder feature in RDN [81], SELSA [82] and PTSEFormer [83] results in **3.7%**, **2.6%** and **1.7%** improvement in the mean average precision (*mAP*), respectively.

Video Frame Interpolation. To evaluate our model for 8× interpolation, we train SuperSloMo [84] and IFR-Net [85] on GoPro [89] training set with our TransFlow encoder features embedded, and test the trained model on GoPro testing set. As shown in Table 8, the updated model outperform original methods with 2 input frames in both *PSNR* and *SSIM* (*e.g.*

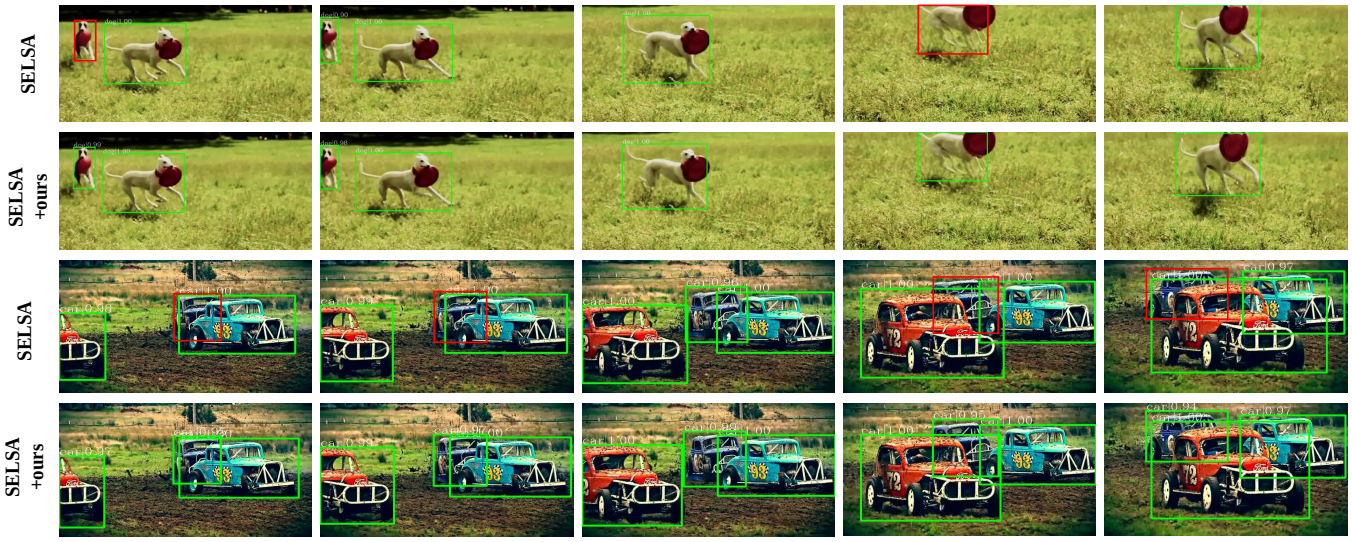


Fig. 8: **Qualitative results of video object detection on ImageNet VID [88] dataset.** Valid detections are marked with green boxes, while missing detections are marked with red boxes. With our approach, moving objects with blurring and occlusion can be better detected now. See §4.3 for details.

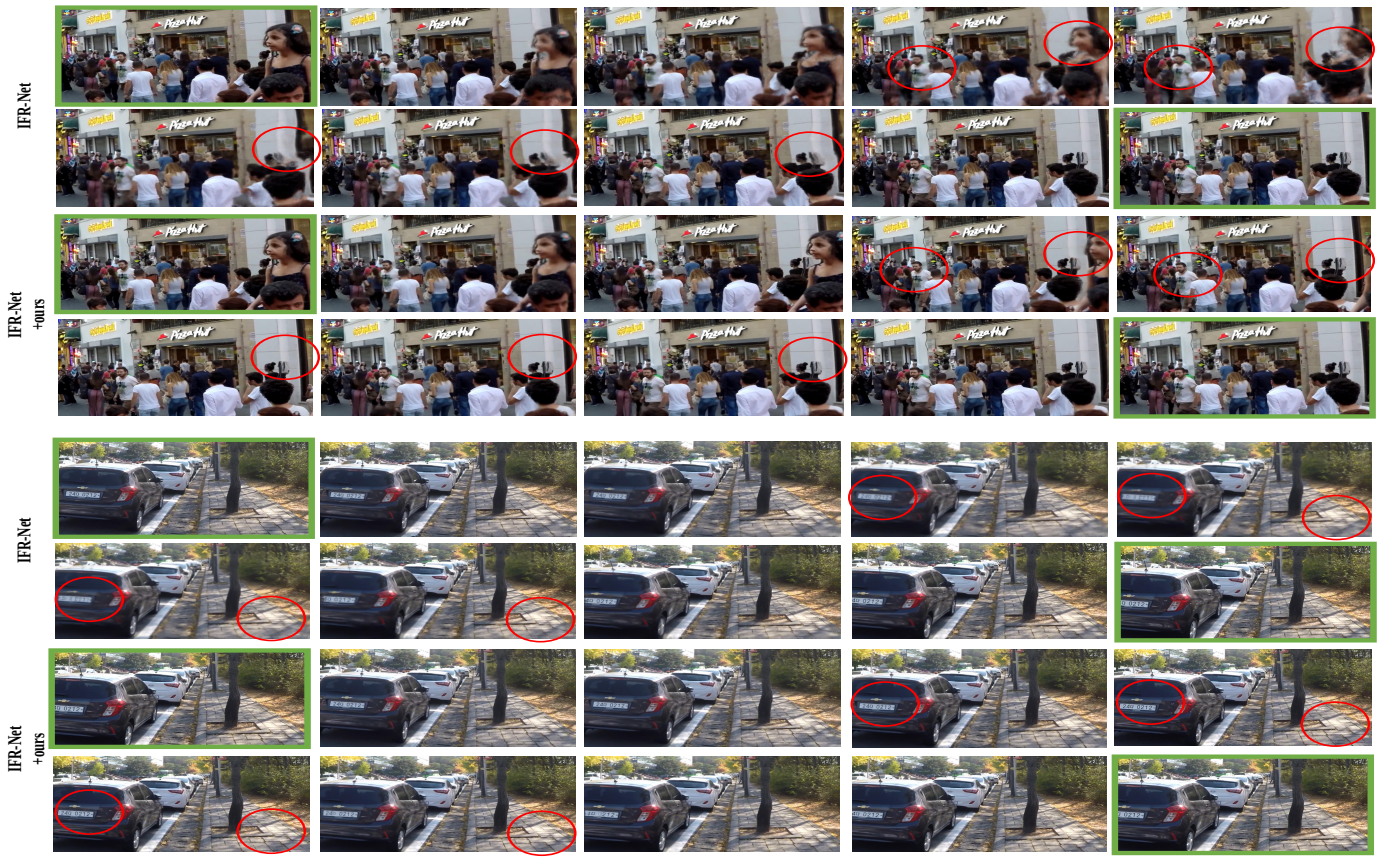


Fig. 9: **Qualitative results of video frame interpolation on GoPro [89] dataset**. With our approach, fewer ghosting and distortion on persons and details during the video interpolation are generated. Green boundary indicates inputs and ○ highlights comparing details. See §4.3 for details.

0.29 dB higher results than SuperSloMo and **0.18** dB higher than IFR-Net).

Video Stabilization. We follow the training configurations of StabNet [86] and PWStableNet [87] and aggregate the

Fig. 10: **Qualitative results of video stabilization on DeepStab [86] dataset**. With our approach, more stable and consistent frames during the video interpolation are generated. ○ highlights comparing details. See §4.3 for details.

	Method	Backbone	*mAP* (%)
Video Object Detection	RDN [81]	ResNet-50	76.7
	RDN+ours	ResNet-50	80.4 (3.7↑)
	SELSA [82]	ResNet-101	80.3
	SELSA+ours	ResNet-101	82.9 (2.6↑)
	PTSEFormer [83]	ResNet-101	87.4
	PTSEFormer+ours	ResNet-101	89.1 (1.7↑)
	Method	PSNR	SSIM
Video Interpolation	SuperSloMo [84]	28.52	0.891
	SuperSloMo+ours	28.81 (0.29↑)	0.905 (0.014↑)
	IFR-Net [85]	29.84	0.920
	IFR-Net+ours	30.02 (0.18↑)	0.932 (0.012↑)
	Method	Distortion	Stablity
Video Stabilization	StabNet [86]	0.83	0.75
	StabNet+ours	0.85 (0.02↑)	0.79 (0.04↑)
	PWStableNet [87]	0.79	0.80
	PWStableNet+ours	0.82 (0.03↑)	0.82 (0.02↑)

TABLE 8: **Quantitative comparison of downstream video task performance with our TransFlow.** See §4.3 for details.

learned features from the TransFlow encoder and the original encoder together for the later regressor. On the DeepStab [86] dataset which contains 61 pairs of stable and unstable videos, TransFlow feature-added method achieves a higher *Distortion Value (D)* and *Stability Score (S)* than the ones without it, as depicted in Table 8.

4.4 Failure cases

Several failure cases in our proposed flow estimation algorithm are presented in Figure 11. These failure cases exemplify scenarios in which the accuracy of the estimated flow is significantly compromised due to the presence of lightning and shadows. A detailed analysis and discussion regarding these instances will be provided in §5.

5 Discussion

Broader Impact. This research critically examines existing CNN-based methods for flow estimation and introduces a transformer architecture with the objective of achieving optimal and efficient performance in flow estimation. The effectiveness of our algorithm has been successfully demonstrated across prominent models using well-established benchmarks like Sintel and KITTI. Leveraging the succinct self-learning paradigm, our approach holds promise for diverse downstream tasks including video object detection, video interpolation, and video stabilization. Moreover, its

Fig. 11: Some failure cases and affected regions of our TransFlow under significant illumination changes and shadows. ○ highlights failure regions. See §4.4 for details.

applicability extends to a wide array of real-world domains, such as autonomous vehicles and medical robots. Given the potential safety implications of erroneous predictions in practical applications, we strongly recommend the implementation of rigorous security protocols to mitigate any adverse societal consequences.

Limitation Analysis. Based on the depicted erroneous predictions in Figure 11, it becomes evident that our model's effectiveness diminishes in scenarios characterized by substantial variations in illumination or the presence of obvious shadowed regions. This limitation can be attributed to the lack of explicit modeling and optimization of flow estimation in shadowed and shaded areas. Analogously, alterations in illumination conditions such as glare, low contrast, and reflectance can likewise contribute to inaccuracies in the predictions. We propose a hypothesis that incorporating a joint modeling approach encompassing scene illumination aspects—encompassing materials, shading, and illumination—could potentially enhance the accuracy in these challenging regions. This avenue for improvement presents an opportunity for the research community to further investigate and explore.

6 Conclusion

In this work, we propose a transformer architecture with spatio-temporal attention for optical flow estimation. It takes advantage of self- and cross-attention to aggregate full image features for reliable matching between adjacent views, and temporal association to establish long-range matching to further refine the prediction by avoiding occlusions and discontinuities. Considering that existing learning paradigms require tedious multi-stage pre-training, we enable a concise self-learning paradigm on the target domain only via our designed strategic masking. The 2D flow learners can be easily extended to 3D scene flow estimation to explore longer and more literal processing. Extensive empirical analysis shows that TransFlow sets new records for public benchmarks. We believe that this work has the potential to provide valuable insights into the applicability of Transformer to a wider range of motion modelling tasks.

Acknowledgement

This work is supported by the National Science Foundation under Award No. 2242243.

References

[1] Y. Lu, Q. Wang, S. Ma, T. Geng, Y. V. Chen, H. Chen, and D. Liu, "Transflow: Transformer as flow learner," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 063–18 073.

[2] Z. Cao, D. Liu, Q. Wang, and Y. Chen, "Towards unbiased label distribution learning for facial pose estimation using anisotropic spherical gaussian," in *ECCV*, 2022.

[3] M. Gehrig, M. Millhäusler, D. Gehrig, and D. Scaramuzza, "E-raft: Dense optical flow from event cameras," in *3DV*, 2021.

[4] K. Luo, C. Wang, S. Liu, H. Fan, J. Wang, and J. Sun, "Upflow: Upsampling pyramid for unsupervised optical flow learning," in *CVPR*, 2021.

[5] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *CVPR*, 2018.

[6] J. Wang, Y. Zhong, Y. Dai, K. Zhang, P. Ji, and H. Li, "Displacement-invariant matching cost learning for accurate optical flow estimation," in *NIPS*, 2020.

[7] Z. Cheng, J. Liang, H. Choi, G. Tao, Z. Cao, D. Liu, and X. Zhang, "Physical attack on monocular depth estimation with optimal adversarial patches," in *ECCV*, 2022.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[9] W. Wang, J. Liang, and D. Liu, "Learning equivariant segmentation with instance-unique querying," *NeurIPS*, 2022.

[10] W. Wang, C. Han, T. Zhou, and D. Liu, "Visual recognition with deep nearest centroids," *NeurIPS*, 2022.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[12] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *ICCV*, 2015.

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

[14] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *CVPR*, 2022.

[15] Z. Huang, X. Shi, C. Zhang, Q. Wang, K. C. Cheung, H. Qin, J. Dai, and H. Li, "Flowformer: A transformer architecture for optical flow," in *ECCV*, 2022.

[16] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," in *ECCV*, 2020.

[17] S. Zhao, Y. Sheng, Y. Dong, E. I. Chang, Y. Xu *et al.*, "Maskflownet: Asymmetric feature matching with learnable occlusion mask," in *CVPR*, 2020.

[18] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. Hartley, "Learning to estimate hidden motions with global motion aggregation," in *ICCV*, 2021.

[19] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *ECCV*, 2012.

[20] M. Menze, C. Heipke, and A. Geiger, "Joint 3d estimation of vehicles and scene flow," *ISPRS Annals*, vol. 2, p. 427, 2015.

[21] M. J. Black and P. Anandan, "A framework for the robust estimation of optical flow," in *ICCV*, 1993.

[22] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *IJCV*, vol. 61, no. 3, pp. 211–231, 2005.

[23] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *IJCV*, vol. 106, no. 2, pp. 115–137, 2014.

[24] 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 *CVPR*, 2017.

[25] M. Hofinger, S. R. Bulò, L. Porzi, A. Knapitsch, T. Pock, and P. Kontschieder, "Improving optical flow on a pyramid level," in *ECCV*, 2020.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[27] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *ACL*, 2019.

[28] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *ICCV*, 2021.

[29] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.

[30] Y. Cui, L. Yan, Z. Cao, and D. Liu, "Tf-blender: Temporal feature blender for video object detection," in *ICCV*, 2021.

[31] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general u-shaped transformer for image restoration," in *CVPR*, 2022.

[32] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, "Fuseformer: Fusing fine-grained information in transformers for video inpainting," in *ICCV*, 2021.

[33] X. Sui, S. Li, X. Geng, Y. Wu, X. Xu, Y. Liu, R. Goh, and H. Zhu, "Craft: Cross-attentional flow transformer for robust optical flow," in *CVPR*, 2022.

[34] C. Yang, Y. Wang, J. Zhang, H. Zhang, Z. Wei, Z. Lin, and A. Yuille, "Lite vision transformer with enhanced self-attention," in *CVPR*, 2022.

[35] F. Wu, A. Fan, A. Baevski, Y. N. Dauphin, and M. Auli, "Pay less attention with lightweight and dynamic convolutions," in *ICLR*, 2019.

[36] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *ICCV*, 2021.

[37] S. Vedula, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 3, pp. 475–480, 2005.

[38] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, and C. Theobalt, "Joint estimation of motion, structure and geometry from stereo sequences." in *ECCV (4)*, 2010, pp. 568–581.

[39] C. Vogel, K. Schindler, and S. Roth, "3d scene flow estimation with a piecewise rigid scene model," *International Journal of Computer Vision*, vol. 115, pp. 1–28, 2015.

[40] W.-C. Ma, S. Wang, R. Hu, Y. Xiong, and R. Urtasun, "Deep rigid instance scene flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3614–3622.

[41] C. Vogel, K. Schindler, and S. Roth, "3d scene flow estimation with a rigid motion prior," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1291–1298.

[42] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3d motion understanding," *International Journal of Computer Vision*, vol. 95, pp. 29–51, 2011.

[43] R. Saxena, R. Schuster, O. Wasenmuller, and D. Stricker, "Pwoc-3d: Deep occlusion-aware end-to-end scene flow estimation," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 324–331.

[44] L. Liu, G. Zhai, W. Ye, and Y. Liu, "Unsupervised learning of scene flow estimation fusing with local rigidity." in *IJCAI*, 2019, pp. 876–882.

[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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8071–8081.

[46] J. Quiroga, T. Brox, F. Devernay, and J. Crowley, "Dense semi-rigid scene flow estimation from rgbd images," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*. Springer, 2014, pp. 567–582.

[47] Y.-L. Qiao, L. Gao, Y. Lai, F.-L. Zhang, M.-Z. Yuan, and S. Xia, "Sf-net: Learning scene flow from rgb-d images with cnns," 2018.

[48] X. Liu, C. R. Qi, and L. J. Guibas, "Flownet3d: Learning scene flow in 3d point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 529–537.

[49] Y. Lu, Y. Zhu, and G. Lu, "3d sceneflownet: Self-supervised 3d scene flow estimation based on graph cnn," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 3647–3651.

[50] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, "Learning feature descriptors using camera pose supervision," in *ECCV*, 2020.

[51] H. Xu and J. Zhang, "Aanet: Adaptive aggregation network for efficient stereo matching," in *CVPR*, 2020.

[52] Z. Feng, L. Yang, L. Jing, H. Wang, Y. Tian, and B. Li, "Disentangling object motion and occlusion for unsupervised multi-frame monocular depth," in *ECCV*, 2022.

[53] X. Li, W. Wang, L. Yang, and J. Yang, "Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality," *arXiv preprint arXiv:2205.10063*, 2022.

[54] S. Prillo and J. Eisenschlos, "Softsort: A continuous relaxation for the argsort operator," in *ICML*, 2020.

[55] Z. Yin, T. Darrell, and F. Yu, "Hierarchical discrete distribution decomposition for match density estimation," in *CVPR*, 2019.

[56] T.-W. Hui, X. Tang, and C. C. Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," in *CVPR*, 2018.

[57] S. Jiang, Y. Lu, H. Li, and R. Hartley, "Learning optical flow from a few matches," in *CVPR*, 2021.

[58] S. Jiang, D. Campbell, Y. Lu, H. Li, and R. Hartley, "Learning to estimate hidden motions with global motion aggregation," in *ICCV*, 2021.

[59] F. Zhang, O. J. Woodford, V. A. Prisacariu, and P. H. Torr, "Separable flow: Learning motion cost volumes for optical flow estimation," in *ICCV*, 2021.

[60] H. Xu, J. Yang, J. Cai, J. Zhang, and X. Tong, "High-resolution optical flow from 1d attention and correlation," in *ICCV*, 2021.

[61] A. Luo, F. Yang, K. Luo, X. Li, H. Fan, and S. Liu, "Learning optical flow with adaptive graph reasoning," in *AAAI*, 2022.

[62] A. Luo, F. Yang, X. Li, and S. Liu, "Learning optical flow with kernel patch attention," in *CVPR*, 2022.

[63] D. Kondermann, R. Nair, K. Honauer, K. Krispin, J. Andrulis, A. Brock, B. Gussefeld, M. Rahimimoghaddam, S. Hofmann, C. Brenner *et al.*, "The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving," in *CVPRW*, 2016.

[64] J. Watson, O. Mac Aodha, V. Prisacariu, G. Brostow, and M. Firman, "The temporal opportunist: Self-supervised multi-frame monocular depth," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1164–1174.

[65] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2. IEEE, 1999, pp. 722–729.

[66] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[67] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[68] X. Li, J. Kaesemodel Pontes, and S. Lucey, "Neural scene flow prior," *Advances in Neural Information Processing Systems*, vol. 34, pp. 7838–7851, 2021.

[69] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," *arXiv preprint arXiv:2001.10773*, 2020.

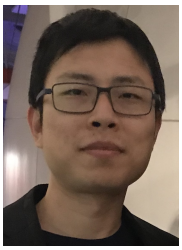
[70] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.

[71] Y. Zou, Z. Luo, and J.-B. Huang, "Df-net: Unsupervised joint learning of depth and flow using cross-task consistency," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 36–53.

[72] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1983–1992.
[73] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0–0.
[74] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, and A. Yuille, "Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding," *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 10, pp. 2624–2641, 2019.
[75] J. Hur and S. Roth, "Self-supervised monocular scene flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7396–7405.
[76] J. H and S. R, "Self-supervised multi-frame monocular scene flow," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2684–2694.
[77] G. Puy, A. Boulch, and R. Marlet, "Flot: Scene flow on point clouds guided by optimal transport," in *European conference on computer vision*. Springer, 2020, pp. 527–544.
[78] Z. Teed and J. Deng, "Raft-3d: Scene flow using rigid-motion embeddings," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 8375–8384.
[79] X. Chu, Z. Tian, B. Zhang, X. Wang, X. Wei, H. Xia, and C. Shen, "Conditional positional encodings for vision transformers," *Arxiv preprint 2102.10882*, 2021. [Online]. Available: https://arxiv.org/pdf/2102.10882.pdf
[80] H. Bao, L. Dong, and F. Wei, "Beit: Bert pre-training of image transformers," in *ICLR*, 2022.
[81] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Relation distillation networks for video object detection," in *ICCV*, 2019.
[82] H. Wu, Y. Chen, N. Wang, and Z. Zhang, "Sequence level semantics aggregation for video object detection," in *ICCV*, 2019.
[83] H. Wang, J. Tang, X. Liu, S. Guan, R. Xie, and L. Song, "Ptseformer: Progressive temporal-spatial enhanced transformer towards video object detection," in *ECCV*, 2022.
[84] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *CVPR*, 2018.
[85] L. Kong, B. Jiang, D. Luo, W. Chu, X. Huang, Y. Tai, C. Wang, and J. Yang, "Ifrnet: Intermediate feature refine network for efficient frame interpolation," in *CVPR*, 2022.
[86] M. Wang, G.-Y. Yang, J.-K. Lin, S.-H. Zhang, A. Shamir, S.-P. Lu, and S.-M. Hu, "Deep online video stabilization with multi-grid warping transformation learning," *TIP*, vol. 28, no. 5, pp. 2283–2292, 2018.
[87] M. Zhao and Q. Ling, "Pwstablenet: Learning pixel-wise warping maps for video stabilization," *TIP*, vol. 29, pp. 3582–3595, 2020.
[88] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.
[89] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *CVPR*, 2017.

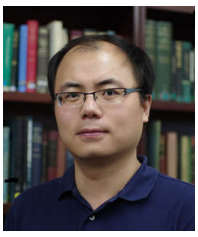
Yawen Lu received the B.S. degree from Zhengzhou University, China and M.S. from Nanyang Technology University, Singapore. He is pursuing a Ph.D. degree in computer graphics at Purdue University. He has served as reviewers and program committees for esteemed journals and conferences such as Pattern Recognition, TIP, TCSVT, CVIU, CVPR, ECCV, AAAI, ACMMM, ISMAR, VIS, IROS, ICRA, RA-L, etc.

Cheng Han received his bachelor degree in Automation from Tianjin University (TJU) in 2019 and his master degree from the Pennsylvania State University (PSU) in 2021. He is currently a Ph.D. candidate in Imaging Science at Rochester Institute of Technology (RIT). His research interests include computer vision, network explainability, representation learning and efficient learning. His publications include flagship conferences in the AI and machine learning such as NeurIPS, ICCV, ICLR, CVPR, and IJCAI.

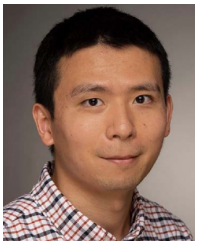


Qifan Wang is the Research Scientist at Meta AI, leading a team building innovative Deep Learning and Natural Language Processing models for Recommendation System. Before joining Meta, He worked at Google Research, and Intel Labs before joining Meta. He received his PhD from Purdue University. Prior to that, he obtained both my MS and BS degrees from Tsinghua University. His research interests include deep learning, natural language processing, information retrieval, data mining, and computer vision. He has co-authored over 50 publications in top-tier conferences and journals, including NeurIPS, SIGKDD, WWW, SIGIR, AAAI, IJCAI, ACL, EMNLP, WSDM, CIKM, ECCV, TPAMI, TKDE and TOIS. He also serve as area chairs, program committee members, editorial board members, and reviewers for academic conferences and journals.

Heng Fan received the B.S. degree from Huazhong Agricultural University, Wuhan, China, in 2013, and the Ph.D. degree from Stony Brook University, Stony Brook, NY, USA, in 2021. He is currently an Assistant Professor with the Department of Computer Science and Engineering, University of North Texas, Denton, TX, USA. His research interests include computer vision, machine learning, and robotic vision. Dr. Fan has served as an Area Chair for WACV in 2022, 2023, and 2024.



Zhaodan Kong received the bachelor's and master's degrees in astronautics and mechanics from the Harbin Institute of Technology, China, in 2004 and 2006, respectively, and the Ph.D. degree in aerospace engineering with a minor in cognitive science from the University of Minnesota, Twin Cities, MN, USA, in 2011. Before joining the University of California (UC) at Davis, CA, USA, he was a Postdoctoral Researcher with the Laboratory for Intelligent Mechatronic Systems and the Hybrid and Networked Systems Lab, Boston University, Boston, MA, USA. He is currently an Associate Professor in mechanical and aerospace engineering with the UC Davis. His current research interests include control theory, machine learning, formal methods, and their applications to human–machine systems, cyber-physical systems, and neural engineering.



Dongfang Liu received the Ph.D. degree from Purdue University. In his research, he is focusing on the development of artificial intelligence-based solutions for interdisciplinary research to address challenges that are of societal importance. He is currently an Assistant Professor with the Department of Computer Engineering, Rochester Institute of Technology, USA. His current research interests include artificial intelligence (AI), machine learning (ML), deep learning (DL), computer vision (CV), human–computer interaction (HCI), and medical imaging. His publication portfolio includes papers from major conferences in the artificial intelligence and robotics fields, such as CVPR, ICCV, AAAI, IJCAI, WWW, WACV, and IROS. He serves on the Program Committee for the American Association for Artificial Intelligence (AAAI) and an Associate Editor for the IET Computer Vision.

Yingjie Victor Chen received his B.S. degree from Tsinghua University, and M.S. and Ph.D. degree at Simon Fraser University (SFU). He is currently a Full Professor and Associate Department Head of Computer Graphics Technology at Purdue University. He has co-authored over 150 articles in journals and conferences. His research topics covers interdisciplinary domains of Information Visualization, Visual Analytics, and Human Computer Interaction.