MemFlow: Optical Flow Estimation and Prediction with Memory

Qiaole Dong and Yanwei Fu✉
School of Data Science, Fudan University
{qldong18, yanweifu}@fudan.edu.cn

Abstract

Optical flow is a classical task that is important to the vision community. Classical optical flow estimation uses two frames as input, whilst some recent methods consider multiple frames to explicitly model long-range information. The former ones limit their ability to fully leverage temporal coherence along the video sequence; and the latter ones incur heavy computational overhead, typically not possible for real-time flow estimation. Some multi-frame-based approaches even necessitate unseen future frames for current estimation, compromising real-time applicability in safety-critical scenarios. To this end, we present MemFlow, a real-time method for optical flow estimation and prediction with memory. Our method enables memory read-out and update modules for aggregating historical motion information in real-time. Furthermore, we integrate resolution-adaptive re-scaling to accommodate diverse video resolutions. Besides, our approach seamlessly extends to the future prediction of optical flow based on past observations. Leveraging effective historical motion aggregation, our method outperforms VideoFlow with fewer parameters and faster inference speed on Sintel and KITTI-15 datasets in terms of generalization performance. At the time of submission, MemFlow also leads in performance on the 1080p Spring dataset. Codes and models will be available at: https://dqiaole.github.io/MemFlow/.

1. Introduction

Optical flow, a critical area in computer vision, plays a key role in various real-world applications like video inpainting [21], action recognition [58], and video prediction [23, 67]. In essence, it captures the displacement vector field for each pixel between successive video frames. Recent advances in optical flow estimation, as highlighted by works such as FlowNet [28], PWC-Net [57], RAFT [60], SKFlow [59], FlowFormer [26], and a rethinking training approach by MatchFlow[17], have been successful. This success is attributed to advancements in model architectures [26, 57, 60] and dedicated datasets [17, 19, 42].

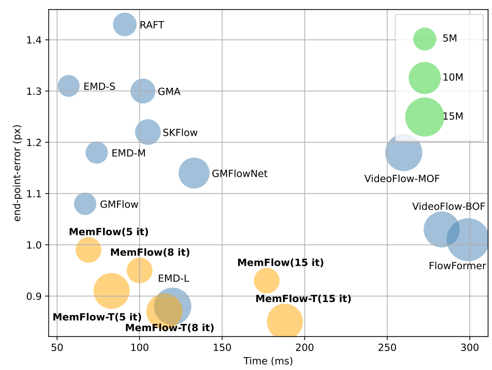

Figure 1. End-point-error on Sintel (clean) vs. inference time (ms) and model size (M). All models are trained on FlyingChairs and FlyingThings3D, and tested with one NVIDIA A100 GPU. MemFlow(-T) (x it) indicates running our network with only x iterations of GRU. Our MemFlow(-T) achieves significant reductions in computational overhead as well as substantial performance boosts over the state-of-the-art methods.

The classical optical flow works use two frames as input, potentially limiting their ability to fully leverage temporal coherence along a video sequence. This limitation results in a bottleneck, prompting an increasing reliance on computationally intensive vision transformer encoders [13] for improved performance, as noted in [26].

Conversely, some recent approaches [40, 49, 53] explore the use of multi-frame videos as input. Typically, these methods either employ simple fusion modules with modest improvements or explicitly model long-range information, incurring heavy computational overhead. For instance, PWC-Fusion [49] straightforwardly fuses backward warped past flow with current flow, resulting in a modest improvement of 0.65% over the baseline PWC-Net [57]. TransFlow [40] and VideoFlow [53] explicitly model long-range motion within a 5-frame context, leading to a significant computational overhead. Importantly, these methods [40, 53] operate in an offline mode, demanding access to unseen future frames in advance for current estimation. Additionally, VideoFlow runs considerably slower than its 2-frame baseline, SKFlow [59], as depicted in Fig. 1. The substantial number of parameters (13.5M) also poses a significant burden on model deployment, causing out-of-memory issues when tested on the 1080p Spring dataset [43] with a single NVIDIA A100 80 GB GPU.

To this end, we present **MemFlow**, an innovative architecture by proposing the memory module [9, 10, 25, 35, 36, 45, 46, 63, 66, 69] for effective optical flow estimation. It operates in real-time (online mode) efficiently with the following notable strengths: (1) *Strong Cross-dataset Generalization Performance*. On both clean and final pass of the Sintel [6] dataset, our model achieves an end-point-error (EPE) of 0.93 and 2.08 pixels. This represents a substantial 23.8% and 15.4% error reduction compared to our 2-frame baseline, SKFlow [59] (1.22 and 2.46 pixels). When evaluated on the KITTI dataset [22], our method demonstrates an error rate of 13.7%, showcasing an 11.6% improvement over SKFlow (15.5%). (2) *High Inference Efficiency*. Our MemFlow achieves an impressive inference speed of 5.6 frames per second (fps) on A100 GPU for processing 1024x436 videos. Even a faster variant of our model, with 5 iterations of GRU, can run at 14.5 fps. Importantly, this accelerated version maintains the near-best generalization performance, as illustrated in Fig. 1.

Technically, our MemFlow is a real-time approach designed for optical flow estimation and prediction, incorporating a memory component. Specifically, MemFlow maintains a memory buffer that stores both historical motion information and context features from the input video stream. As new frame pairs are inputted, the memory buffer is continually updated with the latest context and motion features. We employ an attention mechanism to query the memory buffer using the context feature, extracting useful motion information as the aggregated motion feature. By combining this aggregated motion feature with the current motion and context features, we can regress the residual flow.

Additionally, we introduce a resolution-adaptive rescaling for similarity computation within the attention mechanism to enhance cross-resolution generalization during inference. Furthermore, we provide the option to replace our feature encoder with a more robust vision transformer [13], referred to as **MemFlow-T**, resulting in improved outcomes. As depicted in Fig. 1, our MemFlow (-T) demonstrates significant reductions in computational overhead and substantial performance enhancements compared to state-of-the-art methods. Notably, even with only 5 iterations, our MemFlow surpasses the performance of heavyweight state-of-the-art approaches such as VideoFlow [53] and FlowFormer [26], while running faster than RAFT [60].

In addition to estimating optical flow, we're investigating future flow prediction using our versatile memory module. This capability is crucial for intelligent systems like autonomous vehicles and robots, enabling effective planning and response in dynamic environments. Our adapted network for flow prediction is named **MemFlow-P**. In MemFlow-P, we maintain a memory buffer for decoding residual flow from context and aggregated motion features, eliminating the need for 2D motion features between the current and next frame. Upon the arrival of the next frame, we update the memory module with a new motion feature calculated between the current and next frames. To showcase our flow prediction quality, we combine MemFlow-P with Softmax Splatting [44] and image inpainting [7, 8, 16] for video prediction, involving the synthesis of future video frames based on past ones. Despite not being specifically trained for video prediction, our method achieves comparable results with two competitive flow-based methods [23, 67] in SSIM [64] and LPIPS [71].

In summary, we make four significant contributions: 1) *Innovative Real-Time Optical Flow Estimation*: We introduce a novel architecture that effectively employs a memory module, allowing for real-time optical flow estimation. 2) *Enhanced Generalization with Resolution-Adaptive Re-scaling*: We propose the use of resolution-adaptive re-scaling in attention computation, enhancing cross-resolution generalization performance. 3) *Superior Optical Flow Estimation*: Our MemFlow(-T) achieves state-of-the-art or near-SOTA performance on various standard optical flow estimation benchmarks, demonstrating exceptional performance with minimal computational overhead. 4) *Future Prediction Capability without Explicit Training*: Repurposing MemFlow for optical flow future prediction, we achieve competitive results in video prediction without the need for specific training for this downstream task, highlighting the adaptability of our approach.

2. Related Work

Optical Flow Estimation by Two Frames. Traditionally, it is solved by optimizing energy function [2–5, 24, 50, 70] to maximize visual similarity between images. Recent efforts resort to deep neural networks for directly regressing [15, 19, 27, 28, 48, 57] or generating [18, 52] optical flow from two frames. Specifically, FlowNet [19] first proposed optical flow estimation with end-to-end trainable CNN. PWC-Net [57] and Lite-FlowNet [27] modified the network following the strategy of coarse-to-fine. RAFT [60] further developed a convolutional GRU block upon multi-scale 4D correlation volume for iteratively updating. The following works [17, 26, 32, 40, 53, 59, 73] continually improve the performance of optical flow estimation based on this recurrent architecture. Our MemFlow is also built upon and benefited from the most recently developed GRU-based network, while we employ memory to maintain past motion information and attention for gathering temporal cues for optical flow estimation.

Optical Flow Estimation by Multiple Frames. Traditional works [12, 20] employed Kalman filter for opti-

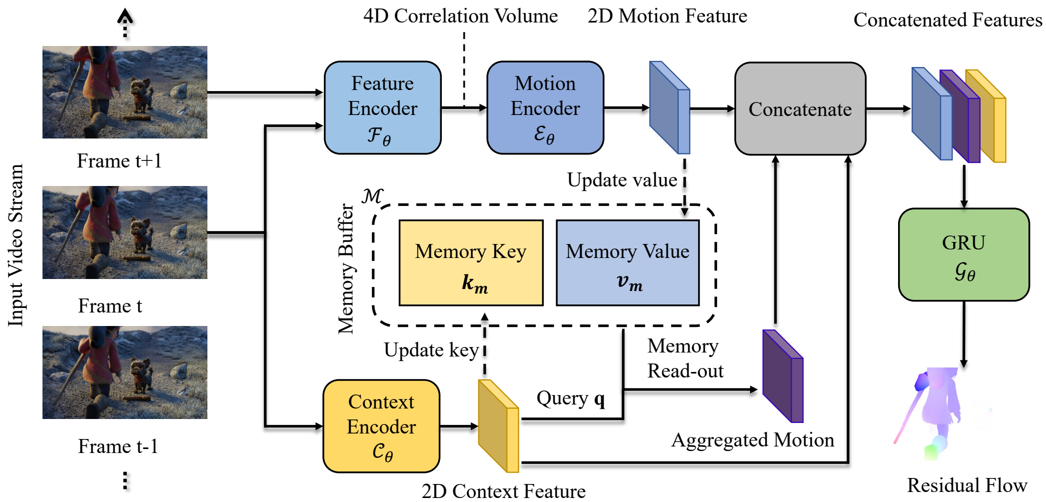

Figure 2. Overview of our MemFlow. MemFlow maintains a memory buffer to store historical motion states of video, together with an efficient update and read-out process that retrieves useful motion information for the current frame's optical flow estimation. It has three key components: 1) *Feature Extractors*. Feature and motion encoder extract and construct the motion feature for the current frame. Another context encoder produces the context feature. 2) *Memory buffer*. Memory buffer stores historical context and motion features and read-out the aggregated motion feature. 3) *Update Modules*. GRU updates the optical flow with a series of residual flows. And the Memory buffer is kept updating when a new frame comes.

cal flow estimation with temporal coherence. Some recent un/self-supervised deep models [31, 37, 38] take three frames to estimate the optical flow of the current frame. On the other hand, there are also several supervised efforts. Particularly, PWC-Fusion [49] fused the backward warped past flow with current flow through a fusion module. RAFT [60] implicitly utilized the historical frames as the initialization to "warm-start" the optical flow estimation of current frame. TransFlow [40] and VideoFlow [53] take five frames (including both past and future frames) to better model the long-range temporal information, while this demands prohibitive computational cost and memory footprint to store and process these frames. To make a balance of performance and cost, our MemFlow integrates a memory module using past frames for optical flow, and thus is capable of being running in an online mode. Moreover, MemFlow outperforms these competitors with better generalization performance while being much more computationally friendly.

Future Prediction by Flows. It aims to predict the optical flow into the future based on past frames [33, 41], motion history [14] or even single image [1, 61, 62]. Luo *et al*. [41] firstly proposed to predict future 3D flow through a convolutional LSTM architecture. And OFNet [14] employed a UNet and ConvLSTM to predict the optical flow autoregressively based on past flows. However, Walker *et al*. [61, 62] predicted the optical flow from a single image and utilized variational autoencoders to model the uncertainty. In contrast, we repurpose our MemFlow for one time step ahead of optical flow prediction with minimal changes and achieve better prediction performance compared to recent OFNet [14] and several strongly competitive baselines.

3. MemFlow

3.1. Definition and Overview

Problem Setup. Optical flow is a per-pixel displacement vector field: $\mathbf{f}_{t\to t+1} = (\mathbf{f}^1, \mathbf{f}^2)$, mapping the location (u, v) in current frame $\mathbf{I}_t$ to next frame $\mathbf{I}_{t+1}$ as $(u + f^1(u), v + f^2(v))$. In the setting of our online multi-frame optical flow estimation, we have access to memory of past history and current frame pair: $\mathbf{I}_t, \mathbf{I}_{t+1}$. We aim to output an optical flow $\mathbf{f}_{t\to t+1}$ and update the memory accordingly. Note that the same MemFlow framework can be directly utilized to estimate optical flow for future prediction. Thus such a task is also evaluated here: we only get video frame $\mathbf{I}_t$ and memory of past, while predicting optical flow $\mathbf{f}_{t\to t+1}$ into future.

Overview. We present an overview of our Memory module for optical Flow estimation (MemFlow) as in Fig. 2. Specifically, our MemFlow consists of three key components: 1) *Feature Extractors*. We have a feature encoder $\mathcal{F}_\theta$ extracting features from the frames to construct the 4D correlation volume subsequently. By correlation lookup operation [60], the following motion encoder $\mathcal{E}_\theta$ can produce the motion feature. To provide context features of the current frame, we also employ the context encoder $\mathcal{C}_\theta$. 2) *Memory buffer*. We store historical context and motion features in the buffer $\mathcal{M}$, while only the aggregated motion feature can be read-out through an attention mechanism. 3) *Update Modules*. We iterate the modules of GRU and Memory for flow and feature updating, respectively. Typically, GRU $\mathcal{G}_\theta$ outputs a series of residual flows. And we update the memory buffer with the final optical flow.

In the next sections, we'll elaborate on our proposed memory module for optical flow estimation (MemFlow(-T)) in Secs. 3.2 and 3.3. Subsequently, we'll demonstrate its application in future prediction through minimal modifications, resulting in our optical flow prediction model, MemFlow-P, as outlined in Sec. 3.4.

3.2. Memory Read-out

We will first introduce the necessary feature extraction module, then present our novel memory read-out and resolution-adaptive re-scaling for the aggregated motion feature here.

Feature Extraction. Given current input image pairs: $\mathbf{I}_t, \mathbf{I}_{t+1}$, we first extract the feature of images at 1/8 resolution by feature encoder $\mathcal{F}_\theta : \mathcal{F}_\theta(\mathbf{I}_t), \mathcal{F}_\theta(\mathbf{I}_{t+1}) \in \mathbb{R}^{H\times W\times D}$, where D is the number of channel; H, W indicate the 1/8 height and width of original images. We then construct the 4D correlation volume C through the dot product between all pairs of features:

$$C = \mathcal{F}_\theta(\mathbf{I}_t) \times \mathcal{F}_\theta(\mathbf{I}_{t+1})^T \in \mathbb{R}^{H\times W\times H\times W}. \quad (1)$$

With the current estimation of optical flow f_i, which is initialized as an all zeros tensor, we can lookup correlation values from C as in [60]. Combined with the current flow, we can get the motion feature $f_m = \mathcal{E}_\theta(f_i, \mathrm{LookUp}(C, f_i))$. Finally, we extract the context feature f_c from $\mathbf{I}_t$ with our context encoder $\mathcal{C}_\theta$, which is trained with the same network architecture of feature encoder $\mathcal{F}_\theta$. Please refer to the implementation and supplementary for the network details.

Memory Read-out. We present a novel module for memory read-out. Our memory buffer $\mathcal{M} = \{k_m \in \mathbb{R}^{L\times D_k}, v_m \in \mathbb{R}^{L\times D_v}\}$, initialized from an empty set, consists of memory keys and values, where $L = l \times H \times W$ is the number of keys and values. D_k, D_v are the feature dimension. We further define l as the length of memory buffers. With current context feature f_c and motion feature f_m, we read-out the aggregated motion feature through an attention mechanism. Specifically, we first linear project f_c, f_m and get the corresponding query, key, and value by concatenation with memory buffer,

$$q = f_c W_q, \quad k = [f_c W_k; k_m], \quad v = [f_m W_v; v_m], \quad (2)$$

where W_q, W_k, W_v are the learnable projection parameters, and $[;]$ is the concatenation operation along first dimension. The aggregated motion feature can be read-out by

$$f_{am} = f_m + \alpha \cdot \mathrm{Softmax}(1/\sqrt{D_k} \times q \times k^T) \times v, \quad (3)$$

where α is a learnable scalar initialized from 0. And we omit the necessary reshape operation here for simplicity. Note that, GMA [32] utilizes attention to aggregate spatial information, while we employ the attention for gathering additional temporal information, as illustrated in Eq. (2). Furthermore, we enhance the attention for resolution adapting through a re-scaling technique, as explained later.

Resolution-adaptive Re-scaling. We also introduce a novel strategy for adapting resolution here. Specifically, if the model is trained using sequences up to length N, it struggles to generalize attention effectively to sequences longer than N. Pioneer work [11] found that the dilution of similarity score accounts for this. So Chiang and Cholak [11] proposed to fix this problem by scaling similarity with $\log n$, where n is the sequence length. In contrast to them, we further update the scaling with average training sequence length n_{avg} as the logarithmic base, and use the length of key k as the sequence length in our cross-attention. So the softmax function in Eq. (3) is updated as

$$\mathrm{Softmax}(\frac{\log_{n_{avg}}(L + H * W)}{\sqrt{D_k}} \times q \times k^T). \quad (4)$$

After incorporating this novel scaling coefficient into memory read-out, it can work for various resolutions and even generalize well to 1080p video as verified in the experiment.

3.3. Memory Update and Flow Estimation

In this section, we introduce a novel memory update strategy and flow estimation with our new memory module.

Particularly, with the context, motion, and aggregated motion features, we can now output a residual flow through a GRU unit: $\Delta f_i = \mathrm{GRU}(f_c, f_m, f_{am})$. After N iterations of GRU, we can get the final optical flow and corresponding motion feature f_m. We then update the memory buffer by inserting the transformed context and motion feature into the key and value tensors of memory,

$$k_m = [f_c W_k; k_m], \quad v_m = [f_m W_v; v_m]. \quad (5)$$

When the memory buffer length l exceeds a pre-defined maximum of l_{max}, we simply discard the obsolete features. Though we try to distill these obsolete features into long-term memory and model the long-range motion information, we find it has no effect on the final performance.

Loss Functions. Our loss functions are inherited from the classical works - RAFT [60]. Generally, we supervise our network with l_1 distance between our partially summed residual flow $\{f_1, \ldots, f_N\}$ and groundtruth f_{gt} with exponentially increasing weights,

$$\mathcal{L} = \sum_{i=1}^{N} 0.85^{N-i} ||f_{gt} - f_i||_1, \quad N = 12. \quad (6)$$

3.4. Beyond Flow Estimation: Future Prediction

The framework in Fig. 2 can be directly utilized for future prediction with minimal changes, and we present the modifications here. More details are in the supplementary.

As we do not have access to frame I_{t+1}, we are not able to calculate the correlation volume and encode the motion feature for the current frame. So we extract the context feature f_c from the current frame and read-out the aggregated motion feature f_{am} from the memory buffer as in Sec. 3.2. Then we predict the optical flow by a small convolutional network based on f_c, f_{am}, and past flow f_p: $f = \mathrm{Convs}(f_c, f_{am}, f_p)$. After the next frame comes, we can now calculate motion feature based on our predicted flow or flow estimated by MemFlow. Finally, we could update the memory buffer of MemFlow-P as in Sec. 3.3 and are ready for optical flow prediction of the next frame. We also use l_1 distance as our loss function.

We further utilize the flow prediction for video prediction. Generally, we forward warp the last video frame by Softmax Splatting [44] with monocular depth from DPT [47] and our predicted optical flow. And we will fill the holes due to splatting with image inpainting method [16].

4. Experiments

Dataset and Implementation. We adopt SKFlow [59] as the network architecture of MemFlow(-P). And we further replace the feature encoder of SKFlow with Twins-SVT [13] as MemFlow-T. The maximum length of memory buffer l_{max} is set to 1. The iteration number of GRU is set to 15 by default during inference. In order to learn better correlation and motion features, we first pre-train our network with 2-frame input on FlyingChair [19] and FlyingThings3D [42] following SKFlow. Subsequently, with 3-frame video as input, we train MemFlow(-T) and MemFlow-P on FlyingThings3D for generalization evaluation and then finetune for Sintel [6] submission with the combination of Sintel, KITTI [22], HD1K [34], FlyingThings3D. Finally, we finetune the model with KITTI and newly proposed Spring [43] for KITTI and Spring submission, respectively. Our network is trained with AdamW [39] optimizer with one-cycle [55] learning rate on two NVIDIA A100 GPUs. Further details are provided in the supplementary material.

Evaluation Metric. We utilize end-point-error (EPE) and Fl-all for Sintel and KITTI evaluation. EPE denotes the l_2 distance between estimated flow and groundtruth. And Fl-all refers to the percentage of outliers whose EPE is larger than 3 pixels and 5% of groundtruth flow magnitude. For the Spring benchmark, we also adopt 1-pixel outlier rate (1px) and WAUC [51], which is the weighted average of the inlier rates for a range of thresholds from 0 to 5 pixels. In the following tables, the best results are in bold, while the second-best ones are underlined.

4.1. Optical Flow Estimation

Generalization Performance. Following previous works, we first show the generalization performance as in Tab. 1. Our MemFlow(-T) achieves state-of-the-art zero-shot performance on both challenging datasets, even with fewer iterations and inference time as shown in Fig. 1. MemFlow with 5 iterations of GRU can even run in real-time while still keeping near-SOTA in terms of generalization. Particularly, though share the same model architecture, our MemFlow reduces EPE by 0.38 and 0.39 from SKFlow [59] on Sintel final pass and KITTI-15, respectively. Besides, we visualize the EPE evolution over iterations in Fig. 3. With only 2 iterations, our MemFlow can beat the previous SKFlow, showing superior efficiency. Fig. 4 further provides a qualitative comparison on Sintel final pass with SKFlow and VideoFlow-MOF [53]. Our MemFlow(-T) not only exhibits more accurate details on large motion regions of hands and head but also are good at fine detail of small object.

Besides, we find that recent multi-frame based VideoFlow is typically not good at estimating optical flow for the first frame of a video. Because VideoFlow needs strict 3-frame or 5-frame input, and output optical flow for the center frame only. So they need to copy the first frame and insert it into the video as a pseudo previous frame, which accounts for why it tends to output some zero flow for the first frame as shown in the first and fourth row of Fig. 4. Yet our MemFlow can work normally without regard to the position of the input frame within the video.

Table 1. Generalization performance of optical flow estimation on Sintel and KITTI-15 after trained on FlyingChairs and FlyingThings3D. *MF* indicates methods using multi frames for optical flow.

Model	Sintel		KITTI-15	
	Clean	Final	Fl-epe	Fl-all
RAFT [60]	1.43	2.71	5.04	17.4
GMA [32]	1.30	2.74	4.69	17.1
GMFlow [68]	1.08	2.48	7.77	23.4
GMFlowNet [72]	1.14	2.71	4.24	15.4
SKFlow [59]	1.22	2.46	4.27	15.5
MatchFlow [17]	1.03	2.45	4.08	15.6
FlowFormer++ [54]	0.90	2.30	3.93	14.1
EMD-L [15]	0.88	2.55	4.12	13.5
TransFlow$^{(MF)}$ [40]	0.93	2.33	3.98	14.4
VideoFlow-BOF$^{(MF)}$ [53]	1.03	2.19	3.96	15.3
VideoFlow-MOF$^{(MF)}$ [53]	1.18	2.56	3.89	14.2
MemFlow (Ours)$^{(MF)}$	0.93	2.08	3.88	13.7
MemFlow-T (Ours)$^{(MF)}$	**0.85**	**2.06**	**3.38**	**12.8**

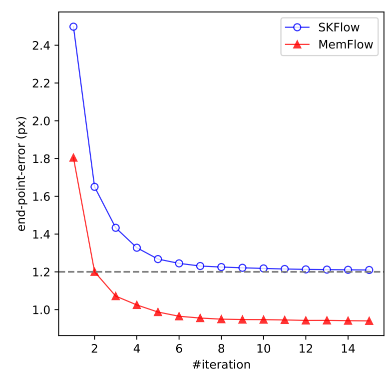

Figure 3. End-point-error of optical flow vs. number of iterations during inference. This figure provides the generalization performance on Sintel (clean) training set. Our method outperforms 15-iteration SKFlow's performance, after using only 2 iterations.

Table 2. Optical flow finetuning evaluation on the public benchmark. *MF* indicates methods using multi frames for optical flow. * uses RAFT's multi-frame "warm-start" strategy on Sintel.

Model	Sintel		KITTI-15
	Clean	Final	Fl-all
RAFT* [60]	1.61	2.86	5.10
GMA* [32]	1.39	2.47	5.15
GMFlow [68]	1.74	2.90	9.32
GMFlowNet [72]	1.39	2.65	4.79
SKFlow* [59]	1.28	2.23	4.84
MatchFlow* [17]	1.16	2.37	4.63
FlowFormer++ [54]	1.07	1.94	4.52
EMD-L [15]	1.32	2.51	4.51
PWC-Fusion$^{(MF)}$ [49]	3.43	4.57	7.17
TransFlow$^{(MF)}$ [40]	1.06	2.08	4.32
VideoFlow-BOF$^{(MF)}$ [53]	<u>1.01</u>	<u>1.71</u>	4.44
VideoFlow-MOF$^{(MF)}$ [53]	**0.99**	**1.65**	**3.65**
VideoFlow-MOF$^{(MF)}$ (online) [53]	-	-	4.08
MemFlow (Ours)$^{(MF)}$	1.05	1.91	4.10
MemFlow-T (Ours)$^{(MF)}$	1.08	1.84	<u>3.88</u>

Finetuning Evaluation. We further report the finetuning results on public benchmark datasets, Sintel and KITTI, in Tab. 2. Our MemFlow(-T) improves SKFlow by a large margin and outperforms most previous methods, *e.g.* SOTA 2-frame based methods FlowFormer++ [54] and multi-frame based TransFlow [40], except recent VideoFlow. However, the online version of 5-frame VideoFlow-MOF degrades Fl-all on the KITTI-15 test set from 3.65 to 4.08, which is worse than our 3.88. We suspect that using multi-frame in an offline mode explicitly can lead to better dataset-specific performance, while with limited cross-dataset generalization performance as in Tab. 1, where 5-frame VideoFlow-MOF performs much worse than 3-frame VideoFlow-BOF on Sintel. Finally, a qualitative result in Fig. 5 on the KITTI test set shows our MemFlow(-T) outperforms others in distinguishing between different vehicles and between the foreground and the sky.

Evaluation on Full-HD Spring Dataset. We also report the generalization and finetuning results on the newly proposed Full-HD (1080p) Spring benchmark in Tab. 3. We first test MemFlow trained on Sintel for evaluation of generalization. Tab. 3 shows that our MemFlow achieves the best EPE and Fl-all while being competitive with MS-RAFT+ [30] in terms of 1px within different regions. After finetuning on the Spring training set, MemFlow outperforms previous SOTA CroCo-Flow [65] by a large margin, though CroCo-Flow is pretrained with additional 5.3M real-world image pairs. Qualitative comparison in Fig. 6 also shows that MemFlow performs better on fine details, while CroCo-Flow employs the much slower tile-technique [29] for high-resolution testing and leads to block-like artifacts. Besides, we should point out that VideoFlow encounters out-of-memory when tested on the Spring dataset with a single NVIDIA A100 80 GB GPU. This further shows that our MemFlow is much more computationally friendly.

4.2. Future Prediction of Optical Flow

For future prediction of optical flow, we compare with following three baselines: (1) *MemFlow*, we use MemFlow to estimate $\mathbf{f}_{t-1\to t}$ with available frames and forward warp the flow to next time step as $\hat{\mathbf{f}}_{t\to t+1}$. (2) *Warped Oracle*, in contrast to (1), we forward warp the available optical flow groundtruth in dataset as a performance upper bound of the warping-based method. (3) *OFNet* [14], a learning-based method that trains a UNet and ConvLSTM with 6 past optical flows as input for future prediction. Note that all trainable models are trained on FlyingThings3D for comparison.

Flow Prediction Results. Left part of Tab. 4 reports the EPE on test split of FlyingThings3D, training set of Sintel and KITTI-15. MemFlow-P outperforms other competitors on all three datasets by a large margin, showing great dataset-specific and cross-dataset performance. More results can be found in supplementary material.

Downstream Task: Video Prediction. We show here quantitatively that our MemFlow-P generalizes well to video prediction. We compare our method with recent two flow-based video prediction models [23, 67] on KITTI. Though without training for video prediction specifically, our method can indeed achieve comparable or even better SSIM [64] and LPIPS [71] as in the right part of Tab. 4.

4.3. Ablation study

In this section, we provide ablation studies on MemFlow by evaluating the generalization performance. We also finetune SKFlow with the same data volume as ours, denoted as SKFlow* in Tab. 5. Note that a T-frame video has $T-1$ flow labels. So the training step varies for fair comparison.

2-Frame Pretraining. We first assess the effect of 2-frame pre-training. As in Tab. 5, pretraining substantially improves the generalization performance on Sintel and KITTI-15, except for the slightly worse EPE of Sintel final pass.

Inference Memory Length. In this experiment, MemFlow is trained using 5-frame video on FlyingThings3D, and the maximum length of memory buffer l_{max} is set to 2 at training. However, Tab. 5 shows that during inference, set l_{max} to 1 can achieve the best performance on three metrics. Compared to not using memory module during inference, enabling the memory module can improve the performance a lot, which shows the benefit of our designed memory module. But increasing l_{max} from 1 to 3 can degrade the results. We think that it's because motion state a few frames ago is less relevant to current motion.

Training Video Length. We are also interested in how long the video we need for training the memory module. Fortunately, Tab. 5 shows that video of 3-frame is good enough

Table 3. Optical flow generalization and finetuning results on Spring [43]. We provide the 1px outlier rate for low/high-detail, (un)matched, (non-)rigid, and (not) sky regions. We also show the EPE, Fl error [22], and WAUC [51]. Important metrics are highlighted in blue.

Dataset	Model	1px total	low-det.	high-det.	matched	unmat.	rigid	non-rig.	not sky	sky	s0-10	s10-40	s40+	EPE	Fl	WAUC
C+T+S+K+H	RAFT [60]	6.79	6.43	64.09	6.00	39.48	4.11	27.09	5.25	30.18	3.13	5.30	41.40	1.476	3.20	90.92
	GMA [32]	7.07	6.70	66.20	6.28	39.89	4.28	28.25	5.61	29.26	3.65	5.39	40.33	0.914	3.08	90.72
	GMFlow [68]	10.36	9.93	76.61	9.06	63.95	6.80	37.26	8.95	31.68	5.41	9.90	52.94	0.945	2.95	82.34
	FlowFormer [26]	6.51	6.14	64.22	5.77	37.29	3.53	29.08	5.50	<u>21.86</u>	3.38	5.53	35.34	0.723	2.38	91.68
	MS-RAFT+ [30]	**5.72**	**5.37**	**61.50**	**5.04**	<u>33.95</u>	**3.05**	<u>25.97</u>	<u>4.84</u>	**19.15**	**2.06**	<u>5.02</u>	<u>38.32</u>	<u>0.643</u>	<u>2.19</u>	**92.89**
	MemFlow	<u>5.76</u>	<u>5.39</u>	<u>63.35</u>	<u>5.11</u>	**32.76**	<u>3.29</u>	**24.42**	**4.49**	24.99	<u>2.92</u>	**4.82**	**32.07**	**0.627**	**2.11**	<u>92.25</u>
+Spring	CroCo-Flow [65]	4.57	4.21	**60.59**	3.85	**34.20**	**2.19**	22.50	4.48	**5.87**	**1.23**	**4.33**	33.13	0.498	1.51	93.66
	MemFlow	**4.48**	**4.12**	61.70	**3.74**	35.12	2.39	**20.31**	**3.93**	12.81	1.31	4.44	**31.18**	**0.471**	**1.42**	**93.86**

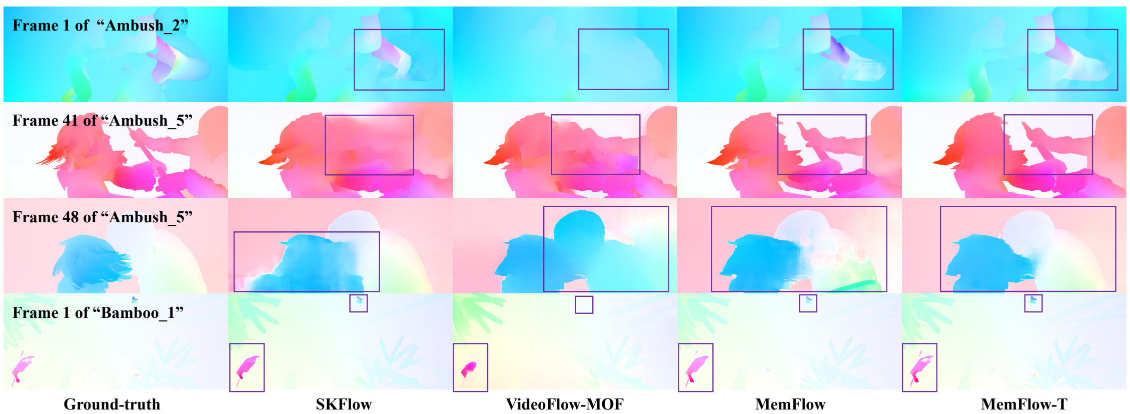

Figure 4. Qualitative comparison on the training set of Sintel final pass after pre-training on FlyingChair and FlyingThings3D. Notable areas are marked by a bounding box. Please zoom in for details.

Table 4. *Left*: End-point-error of flow prediction on FlyingThings3D (Final), Sintel (Final), and KITTI-15. *Right*: Comparison of next frame prediction on KITTI test set (256x832). Note that our method is not trained for video prediction specifically.

Method	Things	Sintel	KITTI
Warped Oracle	14.76	5.76	-
MemFlow	15.70	6.23	12.95
OFNet [14]	13.76	6.03	12.43
MemFlow-P	**7.56**	**5.38**	**8.82**

Method	SSIM↑	LPIPS↓
VPVFI [67]	**0.827**	**0.123**
VPCL [23]	0.820	0.172
Ours	<u>0.825</u>	<u>0.138</u>

Table 5. Ablation studies. Parameters used in our final model are underlined. * means finetuning SKFlow [59] with 1200k steps.

Experiment	Method	Sintel Clean	Sintel Final	KITTI-15 Fl-epe	KITTI-15 Fl-all
Baseline	SKFlow*	1.13	2.39	4.03	14.63
Reference Model, Training: 300k on FlyingThings, max Mem is 2					
2-Frame Pretraining	No	1.16	**2.18**	4.27	16.55
	<u>Yes</u>	**1.00**	2.19	**3.86**	**14.76**
Inference Mem Length	0	1.16	2.41	4.20	15.44
	<u>1</u>	**1.00**	**2.19**	**3.86**	14.76
	2	1.02	2.36	**3.86**	14.65
	3	1.07	2.25	3.87	**14.64**
Reference Model, Training: 600k on FlyingThings, max Mem is 2					
Training Video Length	<u>3</u>	**0.93**	**2.08**	3.88	**13.71**
	5	0.97	2.11	3.80	14.14
	8	0.95	**2.08**	**3.64**	14.65
Warm-start	With	1.02	2.14	3.92	**13.70**
	<u>Without</u>	**0.93**	**2.08**	**3.88**	13.71

to train MemFlow. Training on longer video only results in comparable results but with much more computational overhead, *e.g.* GPU memory and training time.

Warm-start. Warm-start was originally proposed by RAFT for better initialization with the previous flow. It's also compatible with MemFlow. However, equipped with our memory module, warm-start has little effect on the results as shown in Tab. 5. So we don't use warm-start by default.

Resolution-adaptive Re-scaling. We ablate the proposed resolution-adaptive re-scaling on the newly proposed 1080p Spring dataset, which has optical flow groundtruth at a resolution of 1920x1080 and therefore an ideal testbed for research of cross-resolution generalization. MemFlow is trained at a resolution of 368x768 by default, while we also train another model at a much higher resolution of 432x960 for comparison. As shown in Tab. 6, our proposed method can substantially improves the cross-resolution generalization performance. Compared to high-resolution finetuning, our method also performs better with minimal training cost.

Discussion: why set max memory length to 1? We've

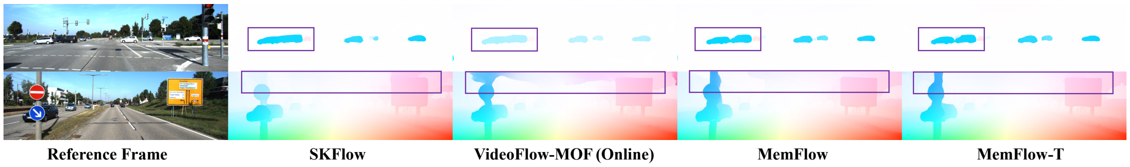

Figure 5. Qualitative comparison on test set of KITTI-15 after finetuning. Ours do much better at distinguishing between different vehicles (first row) and between the foreground and the sky (second row). Please zoom in for details.

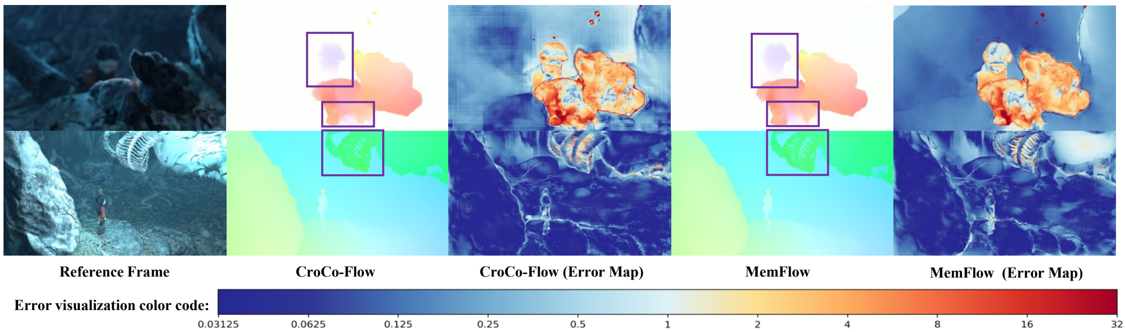

Figure 6. Qualitative comparison with CroCo-Flow [65] on Spring test set. MemFlow performs better on fine details. Notable areas are also marked by a bounding box. Please zoom in for details.

Table 6. Resolution-adaptive re-scaling substantially improves the generalization to Full-HD (*i.e.*, 1920x1080) Spring training set.

Method	1px				EPE
	total	s0-10	s10-40	s40+	
None	4.245	2.535	12.216	42.630	0.448
High-reso. ft	4.456	2.741	12.360	43.174	0.436
Ada. Re-scaling	**4.211**	**2.512**	**12.113**	**42.404**	**0.433**

demonstrated that optimizing the memory buffer's maximum length to 1 achieves the best performance. Essentially, this implies that our model operates in a mode similar to a 3-frame setup. The underlying intuition behind this is that a 3-frame video represents the minimal length needed for effective temporal modeling, ensuring the most consistent motion along the time axis for the same object. Moreover, a 3-frame configuration already encompasses all the matching information required for the center frame [31]. In simpler terms, if there's a pixel in the center frame, it's usually visible in at least one other frame. This is probably one reason why various methods [31, 37, 38, 49, 53] opt for a 3-frame approach in flow estimation. However, it's important to note that our MemFlow operates differently. Our memory module efficiently accumulates and updates the motion state without redundant computations over time and doesn't explicitly extract a 3-frame sequence for estimation.

Discussion: why not a longer range? We have taken an initial step in capturing long-range motion cues for flow estimation. Specifically, we train MemFlow on an 8-frame video and enhance motion features in the memory buffer by adding relative position encoding [56] along the time axis. Despite these efforts, setting the memory length to 1 continues to yield the best results, as indicated in Tab. 7. Furthermore, we have also experimented with distilling outdated memory features into long-term memory based on historical attention scores, following a similar approach to XMem [9]. However, introducing long-term memory doesn't significantly impact performance, as evidenced in Tab. 7. We believe a potential avenue for future research involves exploring long-range motion history for optical flow estimation while ensuring efficiency for real-time applications.

Table 7. More ablations about memory module. All models are trained on 8 frames video with relative position encoding here.

Experiment	Method	Sintel		KITTI-15	
		Clean	Final	Fl-epe	Fl-all
Inference Mem Length	1	1.05	**2.12**	**3.65**	**13.78**
	2	1.06	2.27	**3.65**	13.81
	3	**1.03**	2.18	3.67	13.83
Long Term Mem	With	1.05	2.12	3.65	**13.78**
	Without	**1.03**	**2.10**	**3.64**	13.80

5. Conclusion

We introduced MemFlow, a novel online approach for video-based optical flow estimation and prediction. What sets MemFlow apart is its use of a memory module to store historical motion states. Additionally, MemFlow incorporates resolution-adaptive re-scaling, enhancing cross-resolution performance at minimal training cost. Notably, MemFlow stands out with state-of-the-art cross-dataset generalization and high inference efficiency. Besides, with minimal adjustments, MemFlow can be repurposed for flow prediction, achieving top-notch prediction performance.

Acknowledgements: Yanwei Fu is the corresponding authour. Yanwei Fu is also with Shanghai Key Lab of Intelligent Information Processing, Fudan University, and Fudan ISTBI-ZJNU Algorithm Centre for Brain-inspired Intelligence, Zhejiang Normal University. The computations in this research were performed using the CFFF platform of Fudan University.

References

[1] Dawit Mureja Argaw, Junsik Kim, Francois Rameau, Jae Won Cho, and In So Kweon. Optical flow estimation from a single motion-blurred image. In *Proceedings of the AAAI conference on Artificial Intelligence*, pages 891–900, 2021. 3

[2] Michael J Black and Padmanabhan Anandan. A framework for the robust estimation of optical flow. In *1993 (4th) International Conference on Computer Vision*, pages 231–236. IEEE, 1993. 2

[3] Michael J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

[4] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004.

[5] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International journal of computer vision*, 61(3):211–231, 2005. 2

[6] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 2, 5

[7] Chenjie Cao, Qiaole Dong, and Yanwei Fu. Learning prior feature and attention enhanced image inpainting. In *European Conference on Computer Vision*, pages 306–322. Springer, 2022. 2

[8] Chenjie Cao, Qiaole Dong, and Yanwei Fu. Zits++: Image inpainting by improving the incremental transformer on structural priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[9] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *European Conference on Computer Vision*, pages 640–658. Springer, 2022. 2, 8

[10] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34:11781–11794, 2021. 2

[11] David Chiang and Peter Cholak. Overcoming a theoretical limitation of self-attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7654–7664, Dublin, Ireland, 2022. Association for Computational Linguistics. 4

[12] Toshio M Chin, William Clement Karl, and Alan S Willsky. Probabilistic and sequential computation of optical flow using temporal coherence. *IEEE Transactions on Image Processing*, 3(6):773–788, 1994. 2

[13] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *Advances in Neural Information Processing Systems*, pages 9355–9366. Curran Associates, Inc., 2021. 1, 2, 5

[14] Andrea Ciamarra, Federico Becattini, Lorenzo Seidenari, and Alberto Del Bimbo. Forecasting future instance segmentation with learned optical flow and warping. In *International Conference on Image Analysis and Processing*, pages 349–361. Springer, 2022. 3, 6, 7

[15] Changxing Deng, Ao Luo, Haibin Huang, Shaodan Ma, Jiangyu Liu, and Shuaicheng Liu. Explicit motion disentangling for efficient optical flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9521–9530, 2023. 2, 5, 6

[16] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Incremental transformer structure enhanced image inpainting with masking positional encoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11368, 2022. 2, 5

[17] Qiaole Dong, Chenjie Cao, and Yanwei Fu. Rethinking optical flow from geometric matching consistent perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1337–1347, 2023. 1, 2, 5, 6

[18] Qiaole Dong, Bo Zhao, and Yanwei Fu. Open-ddvm: A reproduction and extension of diffusion model for optical flow estimation. *arXiv preprint arXiv:2312.01746*, 2023. 2

[19] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 1, 2, 5

[20] Michael Elad and Arie Feuer. Recursive optical flow estimation-adaptive filtering approach. *Journal of Visual Communication and image representation*, 9(2):119–138, 1998. 2

[21] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *European Conference on Computer Vision*, pages 713–729. Springer, 2020. 1

[22] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 2, 5, 7

[23] Daniel Geng, Max Hamilton, and Andrew Owens. Comparing correspondences: Video prediction with correspondence-wise losses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3365–3376, 2022. 1, 2, 6, 7

[24] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 2

[25] Li Hu, Peng Zhang, Bang Zhang, Pan Pan, Yinghui Xu, and Rong Jin. Learning position and target consistency for memory-based video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4144–4154, 2021. 2

[26] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. *arXiv preprint arXiv:2203.16194*, 2022. 1, 2, 7

[27] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8981–8989, 2018. 2

[28] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 1, 2

[29] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021. 6

[30] Azin Jahedi, Maximilian Luz, Lukas Mehl, Marc Rivinius, and Andrés Bruhn. High resolution multi-scale raft (robust vision challenge 2022). *arXiv preprint arXiv:2210.16900*, 2022. 6, 7

[31] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proceedings of the European conference on computer vision (ECCV)*, pages 690–706, 2018. 3, 8

[32] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9772–9781, 2021. 2, 4, 5, 6, 7

[33] Xiaojie Jin, Huaxin Xiao, Xiaohui Shen, Jimei Yang, Zhe Lin, Yunpeng Chen, Zequn Jie, Jiashi Feng, and Shuicheng Yan. Predicting scene parsing and motion dynamics in the future. *Advances in neural information processing systems*, 30, 2017. 3

[34] Daniel Kondermann, Rahul Nair, Katrin Honauer, Karsten Krispin, Jonas Andrulis, Alexander Brock, Burkhard Gussefeld, Mohsen Rahimimoghaddam, Sabine Hofmann, Claus Brenner, et al. The hci benchmark suite: Stereo and flow ground truth with uncertainties for urban autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 19–28, 2016. 5

[35] Sangmin Lee, Hak Gu Kim, Dae Hwi Choi, Hyung-Il Kim, and Yong Man Ro. Video prediction recalling long-term motion context via memory alignment learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3054–3063, 2021. 2

[36] Mingxing Li, Li Hu, Zhiwei Xiong, Bang Zhang, Pan Pan, and Dong Liu. Recurrent dynamic embedding for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1332–1341, 2022. 2

[37] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6489–6498, 2020. 3, 8

[38] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4571–4580, 2019. 3, 8

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[40] Yawen Lu, Qifan Wang, Siqi Ma, Tong Geng, Yingjie Victor Chen, Huaijin Chen, and Dongfang Liu. Transflow: Transformer as flow learner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18063–18073, 2023. 1, 2, 3, 5, 6

[41] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. Unsupervised learning of long-term motion dynamics for videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2203–2212, 2017. 3

[42] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 1, 5

[43] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4981–4991, 2023. 2, 5, 7

[44] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 5

[45] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019. 2

[46] Matthieu Paul, Martin Danelljan, Luc Van Gool, and Radu Timofte. Local memory attention for fast video semantic segmentation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1102–1109. IEEE, 2021. 2

[47] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 5

[48] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017. 2

[49] Zhile Ren, Orazio Gallo, Deqing Sun, Ming-Hsuan Yang, Erik B Sudderth, and Jan Kautz. A fusion approach for multi-frame optical flow estimation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2077–2086. IEEE, 2019. 1, 3, 6, 8

[50] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation

of correspondences for optical flow. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1164–1172, 2015. 2

[51] Stephan R Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2213–2222, 2017. 5, 7

[52] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[53] Xiaoyu Shi, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Videoflow: Exploiting temporal cues for multi-frame optical flow estimation. *arXiv preprint arXiv:2303.08340*, 2023. 1, 2, 3, 5, 6, 8

[54] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1599–1610, 2023. 5, 6

[55] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, pages 369–386. SPIE, 2019. 5

[56] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 8

[57] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 1, 2

[58] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1390–1399, 2018. 1

[59] Shangkun Sun, Yuanqi Chen, Yu Zhu, Guodong Guo, and Ge Li. Skflow: Learning optical flow with super kernels. *arXiv preprint arXiv:2205.14623*, 2022. 1, 2, 5, 6, 7

[60] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 1, 2, 3, 4, 5, 6, 7

[61] Jacob Walker, Abhinav Gupta, and Martial Hebert. Dense optical flow prediction from a static image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2443–2451, 2015. 3

[62] Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pages 835–851. Springer, 2016. 3

[63] Hao Wang, Weining Wang, and Jing Liu. Temporal memory attention for video semantic segmentation. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2254–2258. IEEE, 2021. 2

[64] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2, 6

[65] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. Croco v2: Improved cross-view completion pretraining for stereo matching and optical flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17969–17980, 2023. 6, 7, 8

[66] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597, 2022. 2

[67] Yue Wu, Qiang Wen, and Qifeng Chen. Optimizing video prediction via video frame interpolation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17814–17823, 2022. 1, 2, 6, 7

[68] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8121–8130, 2022. 5, 6, 7

[69] Jiyang Yu, Jingen Liu, Liefeng Bo, and Tao Mei. Memory-augmented non-local attention for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17834–17843, 2022. 2

[70] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007. 2

[71] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2, 6

[72] Shiyu Zhao, Long Zhao, Zhixing Zhang, Enyu Zhou, and Dimitris Metaxas. Global matching with overlapping attention for optical flow estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17592–17601, 2022. 5, 6

[73] Zihua Zheng, Ni Nie, Zhi Ling, Pengfei Xiong, Jiangyu Liu, Hao Wang, and Jiankun Li. Dip: Deep inverse patchmatch for high-resolution optical flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8925–8934, 2022. 2