# M2Flow: A Motion Information Fusion Framework for Enhanced Unsupervised Optical Flow Estimation in Autonomous Driving

**Xunpei Sun**[1], **Gang Chen**[1*], **Zuoxun Hou**[2]

[1]School of Computer Science, Sun Yat-sen University, China, 510006
[2]Beijing Institute of Space Mechanics and Electricity, China, 100094
sunxp7@mail2.sysu.edu.cn, cheng83@mail.sysu.edu.cn, houzx_bisme@spacechina.com

## Abstract

Estimating optical flow in occluded regions is a crucial challenge in unsupervised settings. In this work, we introduce M2Flow, a novel framework for unsupervised optical flow estimation that integrates motion information from multiple frames to address occlusions. By modeling inter-frame motion information and employing Motion Information Propagation (MIP) module, M2Flow effectively propagates and integrates motion information across frames, while concurrently estimating bidirectional optical flows for multiple frames. In addition, to handle occlusions across multiple frames, we provide two augmentation modules specifically designed for our multi-frame model to further refine optical flow. The experiments on KITTI and Sintel datasets demonstrate that M2Flow outperforms other state-of-the-art unsupervised approaches, especially in solving occlusions. Code is available at https://github.com/sunzunyi/M2FLOW.

## Introduction

Optical flow is a crucial domain in computer vision that provides a description of motion in images. It is widely applied in tasks such as autonomous driving, object tracking, and video editing. In recent years, the field of optical flow has seen rapid advancements driven by deep learning. Deep learning-based optical flow estimation can be categorized into two types: supervised (Teed and Deng 2021) and unsupervised. Unlike supervised optical flow networks, unsupervised optical flow networks do not require the costly ground-truth labels for training (Yuan et al. 2022) and can be directly trained in the target domain without pre-training on synthetic datasets (Sun et al. 2021; Huang et al. 2023).

Unsupervised optical flow networks define their loss function based on the assumption of constant brightness (Meister, Hur, and Roth 2018), which posits that corresponding points between frames maintain a similar local appearance. However, this assumption will fail in occluded or significant lighting changed regions (Marsal et al. 2023). Occlusions are prevalent in images containing motion, making the handling of these regions crucial for unsupervised methods.

Recently, numerous studies have shown that incorporating multi-frame data can mitigate occlusion issues (Janai et al. 2018; Liu et al. 2019b). Specifically, information extracted from historical frames is utilized for current frame optical flow estimation. Nevertheless, existing multi-frame unsupervised methods suffer from two primary issues. Firstly, in model construction, previous multi-frame unsupervised optical flow methods mostly rely on the constant velocity assumption (Janai et al. 2018; Liu et al. 2019b, 2020)(as illustrated in Fig. 1). However, this assumption often fails in some scenarios involving moving objects, particularly in autonomous driving contexts, leading to erroneous estimation results. Secondly, in training, most methods lack specifically designed data augmentation techniques for multi-frame scenarios, particularly the lack of occlusions with continuous motion and the inability to generate large displacement occlusions. Thus, they fail to effectively guide networks in utilizing multi-frame information for occlusion handling.

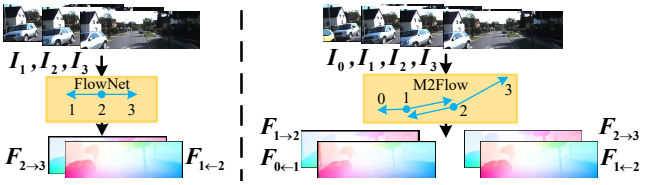



Figure 1: Comparison between previous multi-frame optical flow estimation methods and our approach. (Left) Previous methods assume that the forward and backward flows are equal in magnitude but opposite in direction. (Right) We introduce M2Flow, a novel framework that abandons the constant velocity assumption and integrates multi-frame motion information to concurrently estimate bidirectional optical flows for multiple frames.

In this paper, we introduce M2Flow, as shown in Fig. 1, a novel unsupervised optical flow estimation framework that integrates multi-frame motion information to enhance the accuracy of optical flow estimation in occluded regions. Regarding the first issue, rather than using uniform motion assumption between frames, M2Flow uses a dedicated network to provide a more accurate motion modeling for the occluded objects using four-frame inputs. It propagates non-constant motion information between different frames, estimating the motion trends of the scene more precisely. Subsequently, it integrates the motion information from other

*Corresponding author.

frames with that of the reference frame to address occlusion.

To address the second issue, we further propose two data augmentation methods for multi-frame scenarios, namely consecutive frames augmentation (CFA) and chain-based large displacement augmentation (CLDA), respectively. The CFA creates occlusions across multiple frames, aiming to improve the model's capacity to extract optical flow cues, especially in occluded regions. The CLDA utilizes optical flow from images with small intervals to self-supervise optical flow in images with large intervals, thereby enhancing the model's ability to model motion information.

Experiments show that M2Flow achieves leading accuracy compared to state-of-the-art unsupervised methods on both KITTI and Sintel benchmarks. In particular, we achieve EPE=1.2 on KITTI-2012 benchmark and F1-all=7.37% on KITTI-2015 benchmark. In contrast to state-of-the-art approaches such as SemARFlow (Yuan et al. 2023) and UnSAMFlow (Yuan et al. 2024), which require additional semantic segmentation information as input in the inference phase that depends on large complex models such as Segment Anything Model (SAM) (Kirillov et al. 2023), M2Flow only utilizes four raw images during its inference phase. Even so, M2Flow still outperforms SemARFlow and UnSAMFlow. When compared to state-of-the-art approaches that only use raw images, M2Flow outperforms UPFlow (Luo et al. 2021) (EPE=1.4, F1-all=9.38%) by a large margin. In summary, our contributions are as follows:

- We propose a novel multi-frame unsupervised optical flow estimation framework, termed M2Flow, which enhances the accuracy of optical flow estimation in occluded regions.
- We introduce a CFA technique that enhances the model's ability to extract optical flow cues from other frames, particularly in occluded regions.
- We propose a CLDA method to strengthen the model's capability to model motion information.

## Related works

**Two-Frame Optical Flow** FlowNet (Dosovitskiy et al. 2015) pioneered end-to-end trainable CNNs for optical flow estimation. Subsequently, several supervised learning CNN-based flow estimators have been introduced (Sun et al. 2018; Hui, Tang, and Loy 2018). RAFT (Teed and Deng 2021) develops a recurrent optimizer on a multi-scale 4D correlation volume to estimate flow, yielding superior performance. Several studies (Sun et al. 2022; Shi et al. 2023b) enhance this architecture to further improve performance.

Meanwhile, to avoid the requirement for ground-truth labels in supervised training, some works delve into unsupervised methods. Early unsupervised optical flow estimation used photometric loss and smoothness loss for training supervision (Ren et al. 2017; Luo et al. 2021). However, photometric loss is not suitable for occluded pixels. To address the issue of occlusion, some works exclude occluded regions from photometric loss, including forward-backward consistency (Meister, Hur, and Roth 2018) and distance map occlusion checking (Wang et al. 2018). OIFlow (Liu et al. 2021) addresses occluded regions by inpainting. Several studies (Liu et al. 2019a,b, 2020) utilize knowledge distillation techniques to enhance the performance of self-supervised learning on occluded pixel flows. There are additional methods to further constrain the problem by introducing supplementary information, such as pose, depth (Zou, Luo, and Huang 2018; Ranjan et al. 2019), and semantic segmentation cues (Yuan et al. 2023, 2024). UnSAMFlow (Yuan et al. 2024) integrates Segment Anything Model (SAM) (Kirillov et al. 2023) at the feature level, achieving SOTA performance across multiple benchmarks. Our method focuses on unsupervised optical flow estimation and explores solving occlusion problems through multi-frame information.

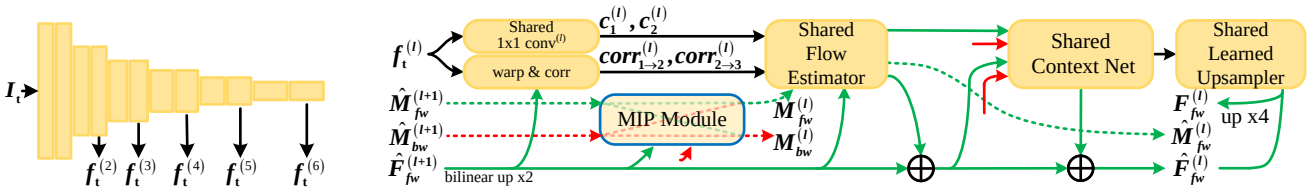
**Multi-Frame Optical Flow** In supervised learning, PWC-Fusion (Ren et al. 2019) improves upon PWC-Net (Sun et al. 2018) by introducing backward warping of past flow and fusing it with the current flow. TransFlow (Lu et al. 2023) and VideoFlow (Shi et al. 2023a) use five frames to model long-range temporal information. However, they rely on prior access to future frames. MemFlow (Dong and Fu 2024) utilizes attention mechanisms to read and aggregate motion features. Due to well-designed models and robust supervision from ground-truth, these methods have achieved notable effectiveness.

In unsupervised learning, multi-frame information has been shown to effectively address occlusion issues (Stone et al. 2021). PCLNet (Guan, Li, and Zheng 2019) used convolutional long short-term memory (ConvLSTM) to merge information from previous frames. Back2Future (Janai et al. 2018) directly encodes temporal relationships by leveraging the constant velocity assumption across three frames. SelFlow (Liu et al. 2019b) employs three frames to estimate the optical flow of the current frame, using initial backward flow and backward cost volume information during forward flow estimation. ARFlow (Liu et al. 2020) extends the modified PWC-Net (Sun et al. 2018) model to multiple frames by repeating the warping and correlation to the backward features. Compared to the aforementioned unsupervised methods, our approach does not rely on the assumption of uniform motion. Instead, it leverages multi-frame estimation to capture the trend of motion changes for a more accurate optical flow estimation. Furthermore, to enhance the efficiency of network training, we have designed data augmentation techniques specifically for training multi-frame models.

## Method

Our goal is to estimate dense optical flow fields using an unsupervised optical flow estimation given four consecutive RGB images $\mathbf{I}_0, \mathbf{I}_1, \mathbf{I}_2, \mathbf{I}_3 \in \mathbb{R}^{H \times W \times 3}$. This includes estimating the forward flow $\mathbf{F}_{fw} = \{\mathbf{F}_{1\to2}, \mathbf{F}_{2\to3}\}$ and the backward flow $\mathbf{F}_{bw} = \{\mathbf{F}_{0\leftarrow1}, \mathbf{F}_{1\leftarrow2}\}$.

**Motivation** Previous unsupervised multi-frame optical flow methods typically use three frames and assume constant velocity. This assumption states that the backward optical flow from the current frame to the previous frame has the same magnitude but opposite direction compared to the forward optical flow from the current frame to the next frame. Nevertheless, this assumption is fragile in dynamic scenes, especially in autonomous driving scenarios, where both the ego vehicle and other vehicles can accelerate and decelerate.




(a) shared pyramidal encoder (b) one forward flow iteration of the iterative decoder at $l$-th level

Figure 2: Network structure ($t \in \{0,1,2,3\}, l \in \{6,5,4,3,2\}$). The green solid arrows and dashed arrows represent forward flow and forward motion information, respectively, while the red solid arrows and dashed arrows denote backward flow and backward motion information, respectively. The red solid arrows from the backward flow iteration are depicted simply.

Moreover, due to the perspective effect, the velocity of objects within an image varies according to their distance from the camera. Our method abandons the unreliable constant velocity assumption and instead utilizes multi-frame motion information to estimate the trend of object motion changes.

## Network Structure

Our M2Flow is based on ARFlow (Liu et al. 2020) network backbone. The network structure is shown in Fig. 2.

**Encoder** We employ a fully convolutional encoder (Fig. 2a) to extract the pyramid features $\{\mathbf{f}_t^{(l)} \mid 2 \le l \le 6\}$ for each input image $\mathbf{I}_t (t \in \{0,1,2,3\})$.

**Decoder** The iterative residual refinement decoder estimates $\hat{\mathbf{F}}_{fw}^{(l)} = \{\hat{\mathbf{F}}_{1\to2}^{(l)}, \hat{\mathbf{F}}_{2\to3}^{(l)}\}$ and $\hat{\mathbf{F}}_{bw}^{(l)} = \{\hat{\mathbf{F}}_{0\leftarrow1}^{(l)}, \hat{\mathbf{F}}_{1\leftarrow2}^{(l)}\}$ from zero. Fig. 2b illustrates a single forward flow iteration from the estimated $\hat{\mathbf{F}}_{fw}^{(l+1)}$ to $\hat{\mathbf{F}}_{fw}^{(l)}$. Specifically, $\hat{\mathbf{F}}_{fw}^{(l+1)}$ is upsampled to match the resolution and used to warp $\mathbf{f}_2^{(l)}$ and $\mathbf{f}_3^{(l)}$. The correlation volumes $\mathbf{corr}_{1\to2}^{(l)}$ and $\mathbf{corr}_{2\to3}^{(l)}$ between two warped features and their respective warp targets $\mathbf{f}_1^{(l)}$ and $\mathbf{f}_2^{(l)}$ are computed. A one-by-one convolution layer $\mathrm{conv}^{(l)}$ compresses the feature channels of $\mathbf{f}_1^{(l)}$ and $\mathbf{f}_2^{(l)}$ to a fixed number, resulting in $\mathbf{c}_1^{(l)}$, $\mathbf{c}_2^{(l)}$. The flow estimator network predicts the residual flow to be added to the current flow, and a context network then further refines the flow. In the final iteration, the learned upsampler network (Teed and Deng 2021) upsamples $\hat{\mathbf{F}}_{fw}^{(2)}$ to obtain the final output $\mathbf{F}_{fw}^{(2)} = \{\mathbf{F}_{1\to2}^{(2)}, \mathbf{F}_{2\to3}^{(2)}\}$ on the original resolution $(H, W)$.

In Fig. 2b, the Motion Information Propagation (MIP) module is designed for four-frame input to propagate inter-frame forward motion information $\hat{\mathbf{M}}_{fw}^{(l)} = \{\hat{\mathbf{M}}_{1\to2}^{(l)}, \hat{\mathbf{M}}_{2\to3}^{(l)}\}$ and backward motion information $\hat{\mathbf{M}}_{bw}^{(l)} = \{\hat{\mathbf{M}}_{0\leftarrow1}^{(l)}, \hat{\mathbf{M}}_{1\leftarrow2}^{(l)}\}$.

## Motion Information Propagation

**Motion Information Modeling** For representing motion information, similar to flow, higher-dimensional motion information is also output by the flow estimation network. We add an auxiliary branch to the baseline flow estimation network, which outputs motion information concurrently with flow estimation. During the first iteration, the motion information is randomly initialized using learnable parameters and updated in subsequent iterations.

**MIP for Inter-Frame Information Propagation** As shown in Fig. 2b, the MIP module takes $\hat{\mathbf{M}}_{fw}^{(l+1)}$ and $\hat{\mathbf{M}}_{bw}^{(l+1)}$ from the previous level as inputs, performs inter-frame information propagation, and outputs $\mathbf{M}_{fw}^{(l)}$ and $\mathbf{M}_{bw}^{(l)}$. The input $\hat{\mathbf{M}}_{fw}^{(l+1)}$ and $\hat{\mathbf{M}}_{bw}^{(l+1)}$ are first deconvolved to match the resolution, resulting in $\hat{\mathbf{M}}_{fw}^{(l)}$ and $\hat{\mathbf{M}}_{bw}^{(l)}$. The initial frames of the corresponding forward and backward motion information are the same, either $\mathbf{I}_1$ or $\mathbf{I}_2$. Therefore, the motion residuals are computed by directly adding the forward and backward motion information as follows:

$$\Delta\mathbf{M}^{(l)} = \hat{\mathbf{M}}_{fw}^{(l)} + \hat{\mathbf{M}}_{bw}^{(l)}, \tag{1}$$

where $\Delta\mathbf{M}^{(l)}$ comprises $\Delta\mathbf{M}_1^{(l)}$ and $\Delta\mathbf{M}_2^{(l)}$. $\Delta\mathbf{M}_1^{(l)}$ represents the variation in motion information between the frame pairs $\mathbf{I}_0$, $\mathbf{I}_1$ and $\mathbf{I}_1$, $\mathbf{I}_2$, while $\Delta\mathbf{M}_2^{(l)}$ denotes the variation in motion information between the frame pairs $\mathbf{I}_1$, $\mathbf{I}_2$ and $\mathbf{I}_2$, $\mathbf{I}_3$.

To further propagate motion information across four frames, the motion residuals are warped using optical flow to adjacent frames:

$$\begin{cases} \Delta\tilde{\mathbf{M}}_1^{(l)} = \mathrm{Warp}(\Delta\mathbf{M}_2^{(l)}; \hat{\mathbf{F}}_{1\to2}^{(l)})(1 - \mathbf{O}_{1\to2}^{(l)}), \\ \Delta\tilde{\mathbf{M}}_2^{(l)} = \mathrm{Warp}(\Delta\mathbf{M}_1^{(l)}; \hat{\mathbf{F}}_{1\leftarrow2}^{(l)})(1 - \mathbf{O}_{1\leftarrow2}^{(l)}), \end{cases} \tag{2}$$

where $\mathbf{O}_{1\to2}^{(l)}$ and $\mathbf{O}_{1\leftarrow2}^{(l)}$ are occlusion maps(Meister, Hur, and Roth 2018). The new motion information is obtained from original motion information and the warped motion residuals:

$$\begin{cases} \bar{\mathbf{M}}_{fw}^{(l)} = \{\Delta\tilde{\mathbf{M}}_1^{(l)} - \hat{\mathbf{M}}_{0\leftarrow1}^{(l)}, \Delta\tilde{\mathbf{M}}_2^{(l)} - \hat{\mathbf{M}}_{1\leftarrow2}^{(l)}\}, \\ \bar{\mathbf{M}}_{bw}^{(l)} = \{\Delta\tilde{\mathbf{M}}_1^{(l)} - \hat{\mathbf{M}}_{1\to2}^{(l)}, \Delta\tilde{\mathbf{M}}_2^{(l)} - \hat{\mathbf{M}}_{2\to3}^{(l)}\}. \end{cases} \tag{3}$$

Eq. 3 represents the computation of motion information for the current frame by utilizing the motion variation trends of other frames. The new motion information are then concatenated with the corresponding original motion information to obtain $\mathbf{M}_{fw}^{(l)}$ and $\mathbf{M}_{bw}^{(l)}$, respectively. Thus, MIP completes the propagation and fusion of motion information across four frames. The final output are fed into the flow estimator network to estimate the optical flow.

## Consecutive Frames Augmentation

Our approach to consecutive frames augmentation (CFA) is motivated by the requirement for the network to further learn how to utilize multi-frame information for optical flow estimation. Moreover, supervision based on photometric loss exhibits reduced efficacy in the presence of occlusions. Therefore, we create occlusions within consecutive frames to achieve robust self-supervision.

**Overview** As shown in the top part of Fig. 3, after obtaining the estimated $\mathbf{F}_{1\to2}$, $\mathbf{F}_{2\to3}$, $\mathbf{F}_{0\leftarrow1}$, and $\mathbf{F}_{1\leftarrow2}$ from the input $\mathbf{I}_0$, $\mathbf{I}_1$, $\mathbf{I}_2$, and $\mathbf{I}_3$, some transformations are manually applied sequentially to $\mathbf{I}_0$, $\mathbf{I}_1$, $\mathbf{I}_2$, and $\mathbf{I}_3$ to obtain $\tilde{\mathbf{I}}_0$, $\tilde{\mathbf{I}}_1$, $\tilde{\mathbf{I}}_2$, and $\tilde{\mathbf{I}}_3$. These transformed frames are then input into the network to obtain $\tilde{\mathbf{F}}'_{1\to2}$, $\tilde{\mathbf{F}}'_{2\to3}$, $\tilde{\mathbf{F}}'_{0\leftarrow1}$, and $\tilde{\mathbf{F}}'_{1\leftarrow2}$. Since the previous transformations are

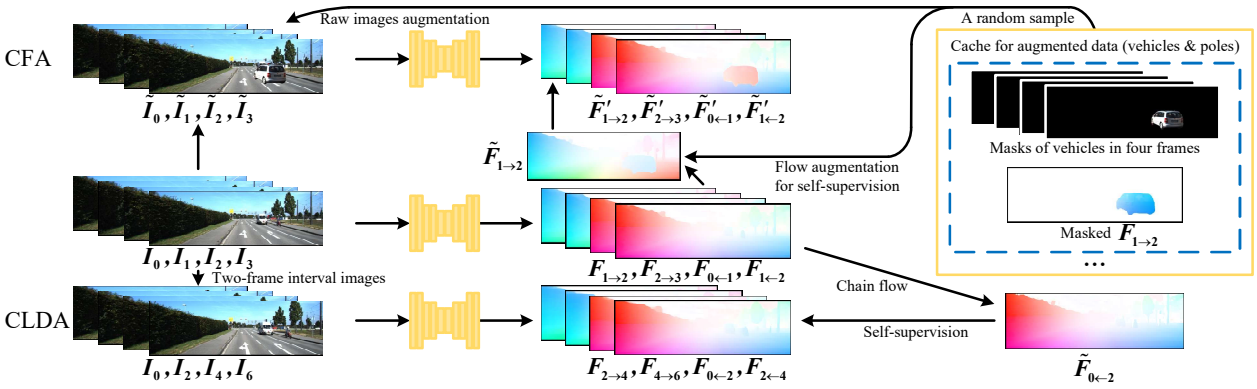



Figure 3: Illustration of CFA (top) and CLDA (bottom).

known, $\tilde{\mathbf{F}}_{1\to2}$ can be derived from $\mathbf{F}_{1\to2}$ to supervise $\tilde{\mathbf{F}}'_{1\to2}$. During the transformation process, we first adopt appearance augmentations (*e.g.*, color, contrast, random noise) and spatial transformations (*e.g.*, translation, random rotation, random rescaling) (Liu et al. 2020). Additionally, we incorporate CFA based on masks.

**CFA for Self-Supervision** This enhancement is inspired by SemARFlow (Yuan et al. 2023), but we extract object masks from a set of consecutive frames instead of a single frame and impose additional constraints to avoid improper mask selection.

Initially, after a single inference, we select specific objects (*e.g.*, cars and poles) in the images through semantic segmentation. Next, in our case, the object masks are extracted from four separate frames, requiring us to ensure that the same object is consistently extracted across consecutive frames. Furthermore, since the optical flow used for self-supervision is directly derived from the first inference, we need to assess its accuracy.

We first compute the Intersection over Union (IoU) of identical semantic objects between adjacent frames. Objects with the highest IoU between two frames are considered to be the same object, preliminarily filtering out identical object masks in consecutive frames. Then, we utilize the optical flow obtained from the initial inference to calculate the photometric loss of the four-frame masks, and set a corresponding loss threshold to further filter out non-corresponding and high optical flow error masks:

$$\begin{cases} \rho\left(\text{Warp}\left(\mathbf{I}_j\mathbb{M}_j;\mathbf{F}_{i\to j}\right),\mathbf{I}_i,\mathbf{O}_i^{\mathbb{M}}\right)\le\theta, \\ \mathbf{O}_i^{\mathbb{M}}=1-(1-\text{invalid}(\mathbf{F}_{i\to j}))\mathbb{M}_i, \end{cases} \tag{4}$$

where $(i,j)\in\{(1,0),(1,2),(2,1),(2,3)\}$. $\rho(\cdot)$ denotes the photometric error calculation method, employing census loss (Meister, Hur, and Roth 2018). $\mathbf{O}_i^{\mathbb{M}}$ represents the occlusion mask. $\mathbb{M}_i$ and $\mathbb{M}_j$ are the object masks corresponding to $\mathbf{I}_i$ and $\mathbf{I}_j$, respectively. $\text{invalid}(\cdot)$ is used to compute the mask for points in optical flow that are directed outside the image boundaries, and $\theta$ is the loss threshold. Only object masks with photometric errors of all four masks below this threshold will be retained. The qualified four-frame masks and the corresponding masked $\mathbf{F}_{1\to2}$ will be stored. In the next inference, some samples will be randomly loaded from the cache to augment the current sample.

## Chain-Based Large Displacement Augmentation

Chain-based large displacement augmentation (CLDA) is a method that utilizes optical flow between single-frame intervals to obtain optical flow for two-frame intervals, subsequently used for self-supervision. The term "chain-based" implies that the optical flow is linked together like a chain. As shown in the bottom part of Fig. 3, since we concurrently estimate the bidirectional optical flows of multiple frames, such as $\mathbf{I}_0$, $\mathbf{I}_1$, $\mathbf{I}_2$, and $\mathbf{I}_3$, we can link the optical flow between multiple frames, taking $\mathbf{F}_{0\leftarrow1}$ and $\mathbf{F}_{1\leftarrow2}$ as examples:

$$\tilde{\mathbf{F}}_{0\leftarrow2}=\mathbf{F}_{1\leftarrow2}+\text{Warp}\left(\mathbf{F}_{0\leftarrow1};\mathbf{F}_{1\leftarrow2}\right), \tag{5}$$

where $\tilde{\mathbf{F}}_{0\leftarrow2}$ represents the optical flow for two-frame intervals. Next, we input the two-frame interval images $\mathbf{I}_0$, $\mathbf{I}_2$, $\mathbf{I}_4$, and $\mathbf{I}_6$, into the network to obtain the $\mathbf{F}_{0\leftarrow2}$, using $\tilde{\mathbf{F}}_{0\leftarrow2}$ for self-supervision.

The concept of CLDA stems from two straightforward ideas: 1). Our method models the motion information of multiple frames, and increasing the data for large displacements helps enhance the network's ability to extract motion information. 2). Appearance changes between single-frame interval images are relatively insignificant, while those between two-frame interval images are relatively significant. Using single-frame interval images simplifies obtaining optical flow, which is used for self-supervised two-frame interval optical flow, thereby improving the network's robustness to large displacements.

## Loss Functions

In our method, the forward and backward optical flows share network parameters. Thus, supervising any single optical flow can train the entire network. Our loss function includes photometric loss and augmentation loss but excludes smoothness loss, as it conflicts with the learned upsampler (Yuan et al. 2023).

**Photometric Loss** During the first inference, we predict the forward and backward optical flows $\mathbf{F}_{fw}^{(l)}=\{\mathbf{F}_{1\to2}^{(l)},\mathbf{F}_{2\to3}^{(l)}\}$ and $\mathbf{F}_{bw}^{(l)}=\{\mathbf{F}_{0\leftarrow1}^{(l)},\mathbf{F}_{1\leftarrow2}^{(l)}\}$ at each level. We warp the corresponding frames with $\mathbf{F}_{1\to2}^{(l)}$ and $\mathbf{F}_{1\leftarrow2}^{(l)}$:

$$\mathbf{I}_i'^{(l)}=\text{Warp}(\mathbf{I}_j^{(l)};\mathbf{F}_{i\to j}^{(l)}),\quad(i,j\in\{1,2\}), \tag{6}$$

where $\mathbf{I}_i^{(l)}$ is the downsampled version of $\mathbf{I}_i$ at the $l$-th scale. Following ARFlow (Liu et al. 2020) , we compute the photometric differences between $\mathbf{I}_i^{(l)}$ and $\mathbf{I}_i'^{(l)}$ using a linear combination of the $L1$ distance, structural similarity (SSIM), and census loss (Meister, Hur, and Roth 2018). The final photometric loss $\ell_{\text{ph}}$ calculates both forward and backward optical flows at each level while occluded regions are masked out.

| | Method | Train | | Test | | | | | Param. |
|---|---|---|---|---|---|---|---|---|---|
| | | 2012 | 2015 | 2012 | | | 2015 | | |
| | | EPE | EPE | Fl-all | Fl-noc | EPE | Fl-all | Fl-noc | |
| Supervised | PWC-Net+ (Sun et al. 2019) | (0.99) | (1.47) | 6.72 | 3.36 | 1.4 | 7.72 | 4.91 | 8.75M |
| | PWC-Fusion (Ren et al. 2019)$^{MF}$ | – | – | – | – | – | 7.17 | 4.47 | 9.33M |
| | RAFT (Teed and Deng 2021) | – | (0.63) | – | – | – | 5.10 | 3.07 | 5.26M |
| | VideoFlow-MOF (Shi et al. 2023a)$^{MF}$ | – | (0.56) | – | – | – | **3.65** | – | 13.45M |
| | MemFlow-T (Dong and Fu 2024)$^{MF}$ | – | – | – | – | – | 3.88 | **2.45** | 12.71M |
| Unsupervised | Back2Future (Janai et al. 2018)$^{MF}$ | – | 6.59 | – | – | – | 22.94 | 13.85 | 12.21M |
| | SelFlow (Liu et al. 2019b)$^{MF}$ | 1.69 | 4.84 | 7.68 | 4.31 | 2.2 | 14.19 | 9.65 | 4.79M |
| | UFlow (Jonschkowski et al. 2020) | 1.68 | 2.71 | 7.91 | 4.26 | 1.9 | 11.13 | 8.41 | 4.36M |
| | ARFlow (Liu et al. 2020) | 1.44 | 2.85 | – | – | 1.8 | 11.80 | – | 2.24M |
| | ARFlow (Liu et al. 2020)$^{MF}$ | 1.26 | 3.46 | – | – | 1.5 | 11.79 | – | 2.37M |
| | UPFlow (Luo et al. 2021) | 1.27 | 2.45 | – | – | 1.4 | 9.38 | – | 3.49M |
| | SemARFlow (Yuan et al. 2023)$^{\dagger}$ | 1.28 | 2.18 | 7.35 | 3.90 | 1.5 | 8.38 | **5.43** | 2.65M |
| | UnSAMFlow (Yuan et al. 2024)$^{\dagger}$ | 1.26 | 2.01 | 7.05 | **3.79** | 1.4 | 7.83 | 5.67 | 2.63M |
| | M2Flow (Ours)$^{MF}$ | **1.09** | **1.95** | **6.24** | 3.95 | **1.2** | **7.37** | 5.73 | 2.65M |

Table 1: KITTI benchmark errors (EPE/px and Fl/%). Metrics evaluated at "all" (all pixels), "noc" (non-occlusions). *MF* indicates methods using multi frames for optical flow. † denotes models with semantic inputs. Missing entries (-) denote unreported results. Parentheses indicate that training and testing are conducted on the same dataset.

**Augmentation Loss** The augmentation loss consists of three components. First, similar to ARFlow (Liu et al. 2020), we calculate the $L1$ distance between the transformed $\mathbf{F}_{1\to2}$ and the optical flow obtained after the second inference. This distance, averaged over non-occluded regions, yields $\ell_{\text{ar}}$.

Secondly, for consecutive frame augmentation (third inference), we compute the $L1$ distance between $\tilde{\mathbf{F}}_{1\to2}$ and $\tilde{\mathbf{F}}'_{1\to2}$, averaging over non-occluded pixels $p$:

$$\ell_{\text{cfa}} = \frac{\sum_p (1 - \tilde{\mathbf{O}}_{1\to2}(p)) \|\tilde{\mathbf{F}}_{1\to2}(p) - \tilde{\mathbf{F}}'_{1\to2}(p)\|_1}{\sum_p (1 - \tilde{\mathbf{O}}_{1\to2}(p))}, \quad (7)$$

where $\tilde{\mathbf{O}}_{1\to2}$ is the augmentation occlusion mask. Given the initial occlusion region $\mathbf{O}_{1\to2}$, calculated using $\mathbf{F}_{1\to2}$ and $\mathbf{F}_{1\leftarrow2}$ through forward-backward consistency check and the object mask $\mathbb{M}$, $\tilde{\mathbf{O}}_{1\to2}$ is defined as:

$$\tilde{\mathbf{O}}_{1\to2} = 1 - \max(1 - \mathbf{O}_{1\to2}, \mathbb{M}). \quad (8)$$

Lastly, for chain-based large displacement augmentation (fourth inference), a similar loss is defined as follows:

$$\ell_{\text{clda}} = \frac{\sum_p (1 - \mathbf{O}_{1\leftarrow2}(p)) \|\tilde{\mathbf{F}}_{0\leftarrow2}(p) - \mathbf{F}_{0\leftarrow2}(p)\|_1}{\sum_p (1 - \mathbf{O}_{1\leftarrow2}(p))}. \quad (9)$$

**Final Loss** Our final loss is:

$$\ell = \ell_{\text{ph}} + \lambda(\ell_{\text{ar}} + \ell_{\text{cfa}} + \ell_{\text{clda}}). \quad (10)$$

# Experiments

## Datasets

We evaluated our method using the KITTI (Geiger et al. 2013; Menze and Geiger 2015), Sintel (Butler et al. 2012), and Cityscapes (Cordts et al. 2016) datasets. The training data schedules follow previous approaches (Liu et al. 2019b; Yuan et al. 2023, 2024).

## Implementation Details

Our model is implemented in PyTorch (Paszke et al. 2019) and trained using the Adam optimizer (Kingma and Ba 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.99$, with a batch size of 4. We first train for 100k iterations on raw data using a fixed learning rate of 0.0002 and then fine-tune on original dataset for another 100k iterations using the OneCycleLR scheduler (Smith and Topin 2019) with a maximum learning rate of 0.00025.

We commence appearance and spatial augmentation at 50k iterations (as described in ARFlow (Liu et al. 2020)). Subsequently, CFA and CLDA begin at 100k iterations. For CFA, on KITTI dataset, we utilize an off-the-shelf model (Zhu et al. 2019) to estimate all semantic segmentation of all images as the initial object masks. For vehicle objects and pole objects, the threshold $\theta$ in Eq. 4 are initially set as 0.35 and 0.25, respectively. After 150k iterations, these values are adjusted to 0.22 and 0.13, respectively. The augmentation loss weight $\lambda$ in Eq. 10 is set to 0.02. On Sintel dataset, we obtained the initial masks using SAM2 (Segment Anything Model 2) (Ravi et al. 2024), with the $\theta$ set to 0.35 initially, and then adjusted to 0.25.

Prior to entering the network, image inputs are resized to 256×832 for KITTI and 448×1024 for Sintel. Data augmentation strategies follow those of ARFlow (Liu et al. 2020), incorporating appearance transformations (brightness, contrast, saturation, hue, gaussian blur, *etc.*), random flipping, and random swapping.

## Benchmark Testing Results

We use standard optical flow metrics, including the average endpoint error (EPE) and the percentage of erroneous pixels (Fl). When calculating Fl, an estimation for each pixel is considered correct if the error is less than 3 pixels or 5% of the magnitude of the ground-truth flow (Menze and Geiger 2015). We compare our

| | Method | Train | | Test | | | | | | Param. |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Clean | Final | Clean | | | Final | | | |
| | | all | all | all | noc | occ | all | noc | occ | |
| Supervised | PWC-Net+ (Sun et al. 2019) | (1.71) | (2.34) | 3.45 | 1.41 | 20.12 | 4.60 | 2.25 | 23.70 | 8.75M |
| | PWC-Fusion (Ren et al. 2019)$^{MF}$ | – | – | 3.42 | 1.38 | 20.10 | 4.57 | 2.22 | 23.73 | 9.33M |
| | RAFT (Teed and Deng 2021) | (0.77) | (1.27) | 1.61 | 0.62 | 9.65 | 2.86 | 1.41 | 14.68 | 5.26M |
| | VideoFlow-MOF (Shi et al. 2023a)$^{MF}$ | (0.46) | (0.66) | **0.99** | **0.40** | **5.83** | **1.62** | **0.77** | **8.54** | 13.45M |
| | MemFlow-T (Dong and Fu 2024)$^{MF}$ | – | – | 1.08 | 0.43 | 6.38 | 1.84 | 0.87 | 9.71 | 12.71M |
| Unsupervised | Back2Future (Janai et al. 2018)$^{MF}$ | (3.89) | (5.52) | 7.23 | 3.60 | 36.78 | 8.81 | 5.03 | 39.65 | 12.21M |
| | SelFlow (Liu et al. 2019b)$^{MF}$ | (2.88) | (3.87) | 6.56 | 2.67 | 38.30 | 6.57 | 3.12 | 34.72 | 4.79M |
| | UFlow (Jonschkowski et al. 2020) | (2.50) | (3.39) | 5.21 | 2.04 | 31.06 | 6.50 | 3.08 | 34.40 | 4.36M |
| | ARFlow (Liu et al. 2020) | (2.79) | (3.73) | 4.78 | 1.91 | 28.26 | 5.89 | 2.73 | 31.60 | 2.24M |
| | ARFlow (Liu et al. 2020)$^{MF}$ | (2.73) | (3.69) | 4.49 | 1.89 | 25.80 | 5.67 | 2.76 | 29.43 | 2.37M |
| | UPFlow (Luo et al. 2021) | (2.33) | (2.67) | 4.68 | 1.71 | 28.95 | 5.32 | 2.42 | 28.93 | 3.49M |
| | UnSAMFlow (Yuan et al. 2024)$^{\dagger}$ | (2.21) | (3.07) | 3.93 | 1.67 | 22.34 | 5.20 | 2.56 | 26.75 | 2.63M |
| | M2Flow (Ours)$^{MF}$ | (2.01) | (3.12) | **3.38** | **1.32** | **20.21** | **5.01** | **2.52** | **25.29** | 2.65M |

Table 2: Sintel benchmark errors (EPE/px). Metrics evaluated at "all" (all pixels), "noc" (non-occlusions), and "occ" (occlusions). *MF* indicates methods using multi frames for optical flow. † denotes models with semantic inputs. Missing entries (-) denote unreported results. Parentheses indicate that training and testing are conducted on the same dataset.

method with both supervised and unsupervised methods on KITTI and Sintel benchmark.

As shown in Tab. 1, on KITTI-2012 and KITTI-2015 datasets, our method outperforms the SOTA unsupervised methods on evaluation metrics over all pixels (EPE and F1-all), and even surpasses some early supervised methods. On the training set, we achieve EPE=1.09 on KITTI-2012 and EPE=1.95 on KITTI-2015. On the test set, our model achieves EPE=1.2 on KITTI-2012 test set and F1-all=7.37% on KITTI-2015 test set, which is significantly better than UPFlow (Luo et al. 2021) (EPE=1.4, F1-all=9.38%, the current state-of-the-art) and ARFlow (Liu et al. 2020) (EPE=1.8, F1-all=11.80%, the backbone network we adapted).

Notably, our model exhibits a slightly higher error in non-occluded regions (F1-noc) compared to SemARFlow (Yuan et al. 2023) and UnSAMFlow (Yuan et al. 2024), which are also based on improvements to ARFlow (Liu et al. 2020).The primary reason is that they integrate semantic segmentation information into the network inputs, leading to better handling of motion boundaries and appearances under different lighting conditions. However, our model (EPE=1.2, F1-all=7.37%) outperforms SemARFlow (EPE=1.5, F1-all=8.38%) and UnSAMFlow (EPE=1.4, F1-all=7.83%) in terms of overall error, indicating that our error reduction primarily stems from occluded regions. This strongly demonstrates the significant advantage of our multi-frame model in handling occlusions.

As shown in Tab. 2, on Sintel dataset, we achieve significant advantages in both occluded and non-occluded regions.

## Qualitative Results

Some qualitative results on KITTI-2015 test set are shown in Fig. 4. Sample 9 shows occlusion from an object's movement (the vehicle is occluded by a traffic light pole), while Sample 17 illustrates occlusion resulting from the motion of the ego vehicle (depicted as dark regions in the error map). Compared to other methods, our approach yields more accurate optical flow in occluded regions.

## Ablation Study

We conducted ablation studies to analyze the effectiveness of each of our proposed modules. All models were trained with the same settings, except for the corresponding ablation variables.

**Network Modules** The ablation experiments on different network modules are presented in Tab. 3. As various modules are incorporated, the error rate gradually decreases. The ablation of MIP module demonstrates the effectiveness of our multi-frame model, particularly in significantly reducing the error in occluded regions (EPE-occ). Furthermore, the ablation of CFA and CLDA validates that our proposed augmentation methods for multi-frame processing can effectively guide the network in learning to integrate information from multiple frames. Specifically, CFA significantly reduces EPE-noc, while CLDA notably decreases EPE-ooc.

This is reasonable, the occluded regions caused by the masks generated by the CFA are generally not extensive. However, using masked flow allows for self-supervision in the non-occluded regions, resulting in a significant reduction in EPE-noc. Conversely, CLDA produces substantial occlusions due to the ego-motion of the camera and the large displacement of objects within the scene, encouraging the network to learn how to handle occluded regions.

**Ablation Analysis for Frames** Our method employs one more frame than the previous multi-frame models. For the purpose of making comparisons under the same input, we removed MIP module while retaining the motion information modeling. Subsequently, we directly warped motion information between different frames to implement a 4-frame model under the constant velocity assumption. As shown in Tab. 4, the results indicate that our 4-frame model with MIP module outperforms the 4-frame model based on the constant velocity motion assumption.

Additionally, we extend M2Flow to 5-frame. As shown in Tab. 4, while the occlusion region further improves with 5-frame, the

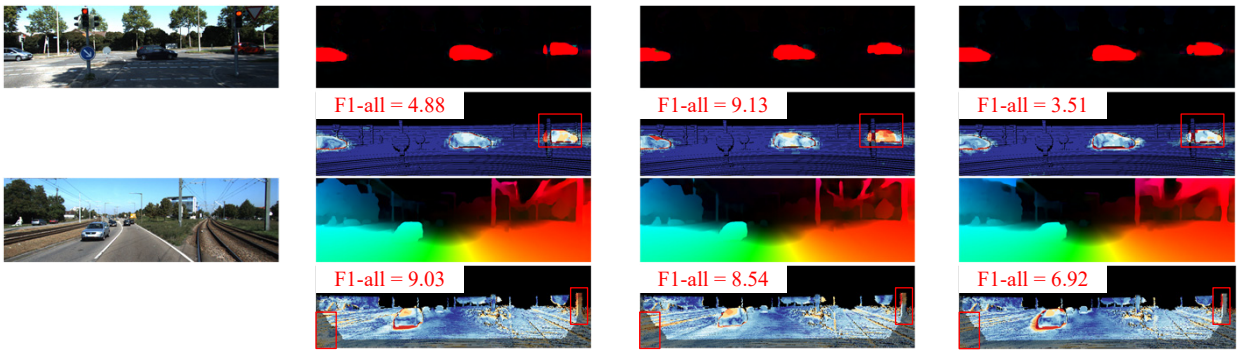



Figure 4: Qualitative results on KITTI test set (sample frame #9, 17) compared with SemARFlow (Yuan et al. 2023) and UnSAMFlow (Yuan et al. 2024). The first and third rows show the predicted optical flow, while the second and fourth rows display the error maps. In the error maps, correct estimations are indicated in blue, incorrect estimations in red, and occluded pixels are represented by dark regions. Additional samples can be viewed on the website of benchmark.

| MIP | CLDA | CFA | F1-all | EPE-all | EPE-noc | EPE-occ |
|---|---|---|---|---|---|---|
| | | | 9.42 | 2.74 | 1.92 | 6.71 |
| ✓ | | | 8.19 | 2.28 | 1.62 | 5.16 |
| ✓ | ✓ | | 8.01 | 2.23 | 1.74 | 4.50 |
| ✓ | | ✓ | 7.57 | 2.07 | **1.44** | 4.85 |
| ✓ | ✓ | ✓ | **7.12** | **1.95** | 1.45 | **4.31** |

Table 3: Ablation study on KITTI-2015 train set (EPE/px and F1/%). Metrics evaluated at "all" (all pixels), "noc" (non-occlusions), "occ" (occlusions). "MIP": motion information propagation module; "CFA": consecutive frames augmentation; "CLDA": chain-based large displacement augmentation.

| Frames | F1-all | EPE-all | EPE-noc | EPE-occ |
|---|---|---|---|---|
| 4 (w/o MIP) | 7.77 | 2.07 | 1.56 | 4.36 |
| 4 (w/ MIP) | **7.12** | **1.95** | **1.45** | 4.31 |
| 5 (w/ MIP) | 7.18 | 2.07 | 1.60 | **4.02** |

Table 4: Ablation analysis for frames on KITTI-2015 train (EPE/px and Fl/%). The experiments used CFA and CLDA.

overall performance is inferior to the 4-frame setting, indicating that a multi-frame approach requires not only increasing the number of input frames but also appropriate feature fusion.

**Comparison of Flow at Various Positions** Our model is capable of concurrently estimating multiple bidirectional optical flows across multiple frames. To evaluate the accuracy of the flows at different positions, we altered the input order of the frames and compared the corresponding flows with the ground truth. As shown in Tab. 5, the flows at various positions achieve comparable accuracy. This further indicates that our model effectively facilitates the mutual propagation of motion information across different frames. Due to the symmetry of the model, $\mathbf{F}_{0\leftarrow1}$ and $\mathbf{F}_{1\leftarrow2}$ have been omitted from Tab. 5.

| Flow | F1-all | EPE-all | EPE-noc | EPE-occ |
|---|---|---|---|---|
| $\mathbf{F}_{1\rightarrow2}$ | 7.14 | 1.96 | 1.46 | **4.30** |
| $\mathbf{F}_{2\rightarrow3}$ | **7.12** | **1.95** | **1.45** | 4.31 |

Table 5: Comparison of flow at various positions on KITTI-2015 train (EPE/px and Fl/%).

| Method | F1-all | EPE-all | EPE-noc | EPE-occ |
|---|---|---|---|---|
| ARFlow | 13.74 | 4.19 | 2.86 | 10.29 |
| M2Flow | **9.33** | **2.47** | **1.72** | **5.82** |

Table 6: Generalization results (train on Cityscapes, and test on KITTI-2015 train).

## Generalization Ability

To evaluate the generalization ability of our model, we trained it on Cityscapes dataset, which consists of urban street scenes. Subsequently, we tested the model on KITTI-2015 train set without fine-tuning. Following (Yuan et al. 2023), we cropped 25% from the bottom of the images to remove the car logo and resized them to 256×704. All other settings remained consistent with KITTI. As shown in Tab. 6, our model demonstrates significantly better performance compared to our backbone network ARFlow (Liu et al. 2020), exhibiting robust generalization capability.

## Conclusion

We propose M2Flow, an unsupervised optical flow network that integrates multi-frame motion information. M2Flow utilizes the MIP module to achieve inter-frame motion information propagation and fusion, enabling the concurrent output of bidirectional optical flows for multiple frames. To address the challenges in training multi-frame networks, we propose consecutive frames augmentation (CFA) and chain-based large displacement augmentation (CLDA) for self-supervision. Experimental results demonstrate that our network excels at predicting flow in occluded regions and achieves significant advantages on benchmark tests.

## Acknowledgments

This research was supported by the Fund of National Key Laboratory of Multispectral Information Intelligent Processing Technology (No. 202410487201) and the National Natural Science Foundation of China under Grant 92470202.

## References

Butler, D. J.; Wulff, J.; Stanley, G. B.; and Black, M. J. 2012. A Naturalistic Open Source Movie for Optical Flow Evaluation. In Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; and Schmid, C., eds., *Computer Vision – ECCV 2012*, 611–625. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-33783-3.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3213–3223.

Dong, Q.; and Fu, Y. 2024. MemFlow: Optical Flow Estimation and Prediction with Memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19068–19078.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T. 2015. FlowNet: Learning Optical Flow with Convolutional Networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2758–2766.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.

Guan, S.; Li, H.; and Zheng, W.-S. 2019. Unsupervised Learning for Optical Flow Estimation Using Pyramid Convolution Lstm. In *2019 IEEE international conference on multimedia and expo (ICME)*, 181–186.

Huang, H.-P.; Herrmann, C.; Hur, J.; Lu, E.; Sargent, K.; Stone, A.; Yang, M.-H.; and Sun, D. 2023. Self-Supervised Autoflow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11412–11421.

Hui, T.-W.; Tang, X.; and Loy, C. C. 2018. LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8981–8989.

Janai, J.; Guney, F.; Ranjan, A.; Black, M.; and Geiger, A. 2018. Unsupervised Learning of Multi-frame Optical Flow with Occlusions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 690–706.

Jonschkowski, R.; Stone, A.; Barron, J. T.; Gordon, A.; Konolige, K.; and Angelova, A. 2020. What Matters in Unsupervised Optical Flow. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 557–572.

Kingma, D.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4015–4026.

Liu, L.; Zhang, J.; He, R.; Liu, Y.; Wang, Y.; Tai, Y.; Luo, D.; Wang, C.; Li, J.; and Huang, F. 2020. 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 (CVPR)*, 6488–6497.

Liu, P.; King, I.; Lyu, M. R.; and Xu, J. 2019a. DDFlow: Learning Optical Flow with Unlabeled Data Distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 8770–8777.

Liu, P.; Lyu, M.; King, I.; and Xu, J. 2019b. SelFlow: Self-Supervised Learning of Optical Flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4571–4580.

Liu, S.; Luo, K.; Ye, N.; Wang, C.; Wang, J.; and Zeng, B. 2021. OIFlow: Occlusion-Inpainting Optical Flow Estimation by Unsupervised Learning. *IEEE Transactions on Image Processing (TIP)*, 30: 6420–6433.

Lu, Y.; Wang, Q.; Ma, S.; Geng, T.; Chen, Y. V.; Chen, H.; and Liu, D. 2023. Transflow: Transformer as Flow Learner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18063–18073.

Luo, K.; Wang, C.; Liu, S.; Fan, H.; Wang, J.; and Sun, J. 2021. UPFlow: Upsampling Pyramid for Unsupervised Optical Flow Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1045–1054.

Marsal, R.; Chabot, F.; Loesch, A.; and Sahbi, H. 2023. BrightFlow: Brightness-Change-Aware Unsupervised Learning of Optical Flow. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2061–2070.

Meister, S.; Hur, J.; and Roth, S. 2018. UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Menze, M.; and Geiger, A. 2015. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3061–3070.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems (NeurIPS)*, 32.

Ranjan, A.; Jampani, V.; Balles, L.; Kim, K.; Sun, D.; Wulff, J.; and Black, M. J. 2019. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12240–12249.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. 2024. Sam 2: Segment Anything in Images and Videos. *arXiv preprint arXiv:2408.00714*.

Ren, Z.; Gallo, O.; Sun, D.; Yang, M.-H.; Sudderth, E. B.; and Kautz, J. 2019. A Fusion Approach for Multi-Frame Optical Flow Estimation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2077–2086.

Ren, Z.; Yan, J.; Ni, B.; Liu, B.; Yang, X.; and Zha, H. 2017. Unsupervised Deep Learning for Optical Flow Estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Shen, S.; Kerofsky, L.; and Yogamani, S. 2023. Optical Flow for Autonomous Driving: Applications, Challenges and Improvements. *arXiv preprint arXiv:2301.04422*.

Shi, X.; Huang, Z.; Bian, W.; Li, D.; Zhang, M.; Cheung, K. C.; See, S.; Qin, H.; Dai, J.; and Li, H. 2023a. VideoFlow: Exploiting Temporal Cues for Multi-frame Optical Flow Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 12435–12446.

Shi, X.; Huang, Z.; Li, D.; Zhang, M.; Cheung, K. C.; See, S.; Qin, H.; Dai, J.; and Li, H. 2023b. FlowFormer++: Masked Cost Volume Autoencoding for Pretraining Optical Flow Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1599–1610.

Smith, L. N.; and Topin, N. 2019. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, 369–386.

Stone, A.; Maurer, D.; Ayvaci, A.; Angelova, A.; and Jonschkowski, R. 2021. SMURF: Self-Teaching Multi-Frame Unsupervised RAFT with Full-Image Warping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3887–3896.

Sun, D.; Vlasic, D.; Herrmann, C.; Jampani, V.; Krainin, M.; Chang, H.; Zabih, R.; Freeman, W. T.; and Liu, C. 2021. AutoFlow: Learning a Better Training Set for Optical Flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10093–10102.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8934–8943.

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2019. Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(6): 1408–1423.

Sun, S.; Chen, Y.; Zhu, Y.; Guo, G.; and Li, G. 2022. SKFlow: Learning Optical Flow with Super Kernels. *Advances in Neural Information Processing Systems (NeurIPS)*, 35: 11313–11326.

Teed, Z.; and Deng, J. 2021. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 4839–4843.

Vihlman, M.; and Visala, A. 2020. Optical Flow in Deep Visual Tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 12112–12119.

Wang, Y.; Yang, Y.; Yang, Z.; Zhao, L.; Wang, P.; and Xu, W. 2018. Occlusion Aware Unsupervised Learning of Optical Flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4884–4893.

Yuan, S.; Luo, L.; Hui, Z.; Pu, C.; Xiang, X.; Ranjan, R.; and Demandolx, D. 2024. UnSAMFlow: Unsupervised Optical Flow Guided by Segment Anything Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 19027–19037.

Yuan, S.; Sun, X.; Kim, H.; Yu, S.; and Tomasi, C. 2022. Optical Flow Training under Limited Label Budget via Active Learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 410–427.

Yuan, S.; Yu, S.; Kim, H.; and Tomasi, C. 2023. SemARFlow: Injecting Semantics into Unsupervised Optical Flow Estimation for Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 9566–9577.

Zhang, K.; Peng, J.; Fu, J.; and Liu, D. 2024. Exploiting Optical Flow Guidance for Transformer-Based Video Inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Zhu, Y.; Sapra, K.; Reda, F. A.; Shih, K. J.; Newsam, S.; Tao, A.; and Catanzaro, B. 2019. Improving Semantic Segmentation via Video Propagation and Label Relaxation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8856–8865.

Zou, Y.; Luo, Z.; and Huang, J.-B. 2018. DF-Net: Unsupervised Joint Learning of Depth and Flow Using Cross-Task Consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 36–53.